

Research Article

An Improved Elman Network for Stock Price Prediction Service

Bo Liu , Qilin Wu, and Qian Cao 

School of Information Engineering, Chaohu University, Chaohu 238000, China

Correspondence should be addressed to Qian Cao; 19875069@qq.com

Received 16 June 2020; Revised 8 August 2020; Accepted 19 August 2020; Published 3 September 2020

Academic Editor: Xiaolong Xu

Copyright © 2020 Bo Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The rapid development of edge computing drives the rapid development of stock market prediction service in terminal equipment. However, the traditional prediction service algorithm is not applicable in terms of stability and efficiency. In view of this challenge, an improved Elman neural network is proposed in this paper. Elman neural network is a typical dynamic recurrent neural network that can be used to provide the stock price prediction service. First, the prediction model parameters and build process are analysed in detail. Then, the historical data of the closing price of Shanghai composite index and the opening price of Shenzhen composite index are collected for training and testing, so as to predict the prices of the next trading day. Finally, the experiment results validate that it is effective to predict the short-term future stock price by using the improved Elman neural network model.

1. Introduction

The stock market can be regarded as a complex nonlinear system, and there are many factors that affect the stock price, especially the recent historical stock price, which has a great influence on the future short-term stock price. So, it is difficult, but valuable, to provide stock price prediction service. Fortunately, with the development of edge computing and neural network technologies, commercial service providers can benefit from the low-latency edge resources and nonlinear expression ability of neural network to provide their users with more efficient service acquisition for stock price prediction. Based on this well-known cognition, we can design a neural network to predict the stock price of the next period based on the historical stock price [1–4]. In this paper, we will use the historical data of the closing price of the Shanghai composite index to predict the closing price of the Shanghai composite index in the next trading day, and the historical data of the opening price of the Shenzhen composite index to predict the opening price of the Shenzhen composite index in the next trading day. In addition, our research could make stock prediction algorithms deployed on edge terminals more efficient.

Over the years, although many scholars have established a large number of mathematical models to predict the stock price, they have not achieved good results and have little

impact. However, the rise of big data technology and artificial intelligence technology will provide another effective solution for stock price prediction. This is the motivation of our research. Specifically, we hope to establish a reasonable artificial intelligence model and make a more accurate prediction of the future short-term stock price by inputting the latest stock price history. We expect this model can offer a reference for people of stock investment.

In this paper, we propose an improved Elman neural network model to predict stock price, and our main contributions include the following:

In order to apply traditional stock prediction algorithms to terminal devices such as edge computing and mobile phones, we build a stock price prediction model based on an improved Elman network with the aim to predict the stock price simpler and more stable. We give the specific model parameters and build process.

In order to reflect the latest stock market situation, we trained and tested the proposed model with the latest dataset, namely, the Shanghai composite index and Shenzhen composite index in 2018, 2019, and 2020; the latest datasets are used to better reflect the current stock market.

To analyse the new algorithm model more clearly, we quantitatively analysed the performance of the model with a variety of mathematical tools and error analysis methods. In addition, a large number of diagrams and tables are provided to further clarify the model.

The rest of this paper is organized as follows. Section 2 reviews and summarizes the related work, on this basis, to clarify the significance of this study. Section 3 is preliminaries in which the principle of Elman neural network is clarified. In Section 4, we proposed our model, and the specific model construction procedures are introduced in detail. Section 5 is experiments in which the model is built, trained, and tested. In addition, we devoted a great deal of space to the analysis of the results in this section. Finally, Section 6 concludes this paper.

2. Preliminaries

Elman neural network is a typical feedback neural network model widely used, which is generally divided into four layers: input layer, hidden layer, bearing layer, and output layer [5, 6].

Figure 1 shows the basic structure of an Elman neural network; the connection of the input layer, hidden layer, and output layer are similar to the feedforward network. The input layer unit only plays the role of signal transmission, and the output layer unit plays the role of weighting. There are linear and nonlinear excitation functions in the hidden layer element, and the excitation function usually takes the nonlinear function of Sigmoid. The bearing layer is used to remember the output value of the hidden layer unit at the previous time, which can be considered as a delay operator with one-step delay. The output of the hidden layer is self-linked to the input of the hidden layer by accepting the delay and storage of the layer, which makes it sensitive to historical data. In other words, Elman neural network adds a bearing layer to the hidden layer as a one-step delay operator to achieve the purpose of memory so that the system has the ability to adapt to time-varying characteristics and enhance the global stability of the network [7–9]. The mathematical expression of its network is

$$\begin{aligned} y(k) &= g(w_3 x(k)), \\ x(k) &= f(w_1 x_c(k) + w_2(u(k-1))), \\ x_c(k) &= x(k-1), \end{aligned} \quad (1)$$

where $y(t)$ is the output node vector, $x(t)$ is the nodal element vector of the hidden layer, $u(t-1)$ is the input vector, $x_c(t)$ is the feedback state vector, w_3 is the connection weight from the hidden layer to the output layer, w_2 is the connection weight from the input layer to the hidden layer, w_1 is the connection weight of the connecting layer to the hidden layer, g is the transfer function of the output neuron and the linear combination of the output of the hidden layer, and f is the transfer function of the hidden layer neuron, usually using the S function.

This research takes MATLAB as the experimental platform. And the two datasets used in this study are the

closing prices of 490 trading days of the Shanghai composite index from September 26, 2017, to September 30, 2019, and the opening prices of 420 trading days from August 15, 2018, to May 12, 2020. We will use the same model for training and testing based on these two datasets.

3. Related Work

In fact, many researchers have been studying stock price forecasts for years, some of these studies have improved the existing models and some have further processed the data. However, these studies are not perfect, and some of the models are too complex and some of the processing procedures are tedious. These shortcomings will increase the instability of the models and limit the application and extension of the research results.

Shi et al. considered that traditional stock forecasting methods could not fit and analyse highly nonlinear and multifactors of stock market well, so there are problems such as low prediction accuracy and slow training speed. Therefore, they proposed a prediction method of the Elman neural network model based on the principal component analysis method. In order to compare the results better, BP network and Elman network with the same structure are established to predict the stock data [10]. Yu et al. used an improved Elman neural network as the forecasting model and the market price of Zhongji company (No. 000039) in Shenzhen stock market is forecasted; their experiment results get higher precision, steadier forecasting effect, and more rapid convergence speed [11]. Zheng et al. studied the forecast of opening stock price based on Elman neural network in 2015, and they selected the opening prices of Shanghai stock index of 337 trading days from December, 2012 to April, 2014 as the raw data for stimulated forecast, and the result proves the validity of their forecast model [12]. Zhang et al. successfully applied Elman regression neural network to the prediction of stock opening price. Specifically, the authors described the Particle Swarm Optimization (PSO) algorithm for learning optimization of the Elman Recurrent Neural Network, and the results showed that the model based on LSTM was more accurate than other machine learning models [13]. Jun used Adaptive Whale Optimization Algorithm and Elman neural network to predict the stock price and achieved better results based on their experiments [14]. Javad Zahedi et al. used the artificial neural network model and principal component analysis to evaluate the predictability of stock price in Teheran stock exchange with 20 accounting variables. Finally, the goodness of fit of principal component analysis was determined by actual values, and the effective factors of Teheran stock exchange price were accurately predicted and modelled by a new model composed of all variables [15]. Han et al. designed a three-ply BP network and the corresponding mathematical model. Therefore, using 140 days actual price of the stock 600688 as a sample, the network was trained through MATLAB; thereby, the 10 days predictions of the stock price and the dispersion $Q=0.0146$ to the practical data were made [16].

Although scholars have made outstanding contributions in using artificial intelligence to predict stock prices, neither the stability of the models nor the accuracy of the predictions

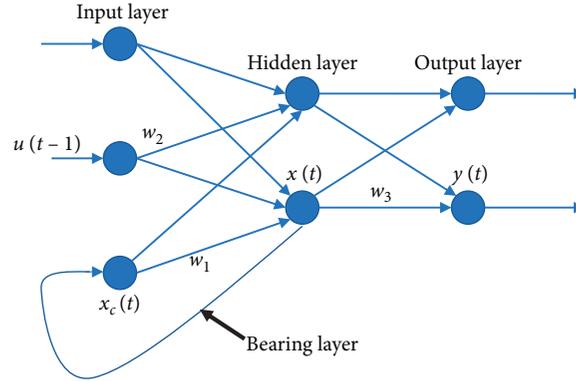


FIGURE 1: The basic structure of an Elman neural network.

is satisfactory. Based on this fact, this study seeks to exploit the neural network model for the prediction of stock price based on Elman network with balancing the simplicity, stability, and accuracy.

4. Supposed Model

The general steps to build the supposed model of this study include data collection, data load, sample set construction, division of sample set and training set, construction of Elman neural network, and training of the neural network model. The specific flow chart is shown in Figure 2.

4.1. Construction of Sample Set. The stock price prediction problem in this study is actually a time series problem, which can be expressed by the following formula:

$$x_n = f(x_{n-1}, x_{n-2}, \dots, x_{n-N}). \quad (2)$$

This formula means that the closing price of the previous N trading day can be used to predict the closing price of the next trading day. The data of 490 closing prices were divided into training samples and test samples; for the training samples, $x_1 \sim x_N$ are selected to form the first sample, where $(x_1, x_2, \dots, x_{N-1})$ are the independent variable and x_N is the dependent variable, and $x_2 \sim x_{N+1}$ are selected to form the second sample, where (x_2, x_3, \dots, x_N) are the independent variable and x_{N+1} is the dependent variable; finally, a training matrix is formed as follows:

$$\begin{pmatrix} x_1 & x_2 & x_i \\ x_2 & x_3 & x_{i+1} \\ \dots & \dots & \dots \\ x_{N-1} & x_N & \\ x_N & x_{N+1} & \end{pmatrix}. \quad (3)$$

In this matrix, each column is a sample, and the last row is the expected output. These samples are fed into the Elman neural network for training, and then the network model can be obtained [17–19].

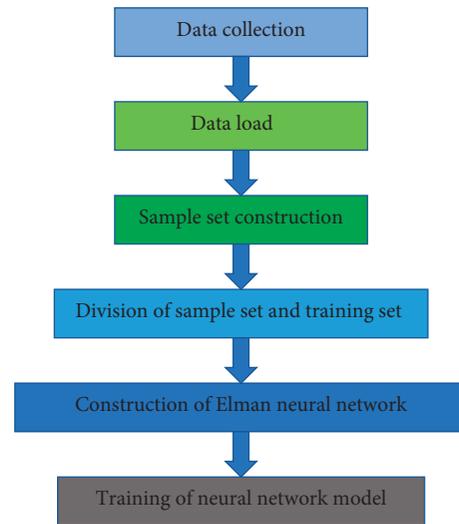


FIGURE 2: Elman neural network model construction steps.

In this study, $x_1 \sim x_N$ are selected to form the first sample, and $x_2 \sim x_{N+1}$ are selected to form the second sample; the rest can be carried out in the same manner. Here, N is randomly set to 8, which means that the closing price of the day is determined by the closing price of the previous seven trading days.

Take the Shanghai composite index dataset as an example, the closing prices of the first eight trading days are 3,343.58, 3345.27, 3339.64, 3348.94, 3374.38, 3382.99, 3388.28, and 3386.10, which means 3,343.58, 3345.27, 3339.64, 3348.94, 3374.38, 3382.99, and 3388.28 will be used to forecast the eighth data 3388.28 which we have already obtained. The closing prices of the first eight trading days are 2999.28, 3006.45, 2977.08, 2985.34, 2955.43, 2929.09, 2932.17, and 2905.19, and the same principle, 2999.28, 3006.45, 2977.08, 2985.34, 2955.43, 2929.09, and 2932.17, will be used to forecast the eighth data 2905.19 which we have already obtained. Therefore, 490 pieces of data will be converted into a 8×483 matrix; 483 columns mean 483 samples, in which the first 7 data in each column are independent variables and the eighth data is the data to be predicted. The 8×483 matrix is shown as follows:

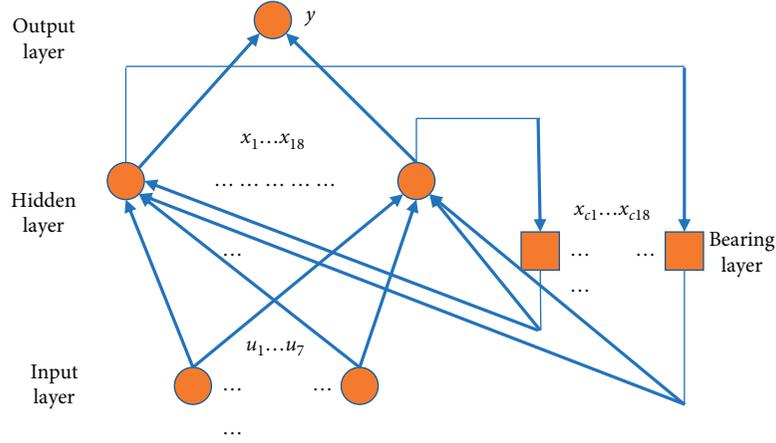


FIGURE 3: Model structure.

$$\begin{pmatrix} 3343.58 & 3390.52 & \dots & 2999.28 \\ 3345.27 & 3378.47 & \dots & 3006.45 \\ 3339.64 & 3372.04 & \dots & 2977.08 \\ 3348.94 & 3381.79 & \dots & 2985.34 \\ 3374.38 & 3370.17 & \dots & 2955.43 \\ 3382.99 & 3378.65 & \dots & 2929.09 \\ 3388.28 & 3380.7 & \dots & 2932.17 \\ 3386.1 & 3388.25 & \dots & 2905.19 \end{pmatrix}. \quad (4)$$

The Shenzhen composite index dataset is 8786.3497, 8470.9094, 8573.5693, 8355.0002, 8419.7868, 8533.4289, 8446.9836, 8480.2244, 8511.3743, 8731.6394, 8716.8172, 8666.9025, 8509.2723, 8440.9528, 8454.1357, 8519.5698, ..., 10477.7614, 10460.9947, 10575.5242, 10618.1651, 10899.9169, 10923.6123, 11053.8157, 10972.0503. In the same way, the Shenzhen composite index dataset is formed as a 8×413 matrix, which is as follows:

$$\begin{pmatrix} 8786.3497 & 8511.3743 & \dots & 10477.7614 \\ 8470.9094 & 8731.6394 & \dots & 10460.9947 \\ 8573.5693 & 8716.8172 & \dots & 10575.5242 \\ 8355.0002 & 8666.9025 & \dots & 10618.1651 \\ 8419.7868 & 8509.2723 & \dots & 10899.9169 \\ 8533.4289 & 8440.9528 & \dots & 10923.6123 \\ 8446.9836 & 8454.1357 & \dots & 11053.8157 \\ 8480.2244 & 8519.5698 & \dots & 10972.0503 \end{pmatrix}. \quad (5)$$

413 columns mean 413 samples, in which the first 7 data in each column are independent variables and the eighth data is the data to be predicted.

4.2. Construction of Elman Neural Network. Figure 3 shows the proposed model structure, where u_1, \dots, u_7 are input data, x_1, \dots, x_{18} are hidden-layer data, and x_{c1}, \dots, x_{c18} are bearing-layer data. With the help of MATLAB neural network toolbox, Elman neural network can be easily

built. To be specific, the MATLAB neural network toolbox provides an Elmanet function, and Elman network construction can be completed by setting three parameters in the Elmanet function, which are the delay time, the number of hidden layer neurons, and the training function, respectively. In this case, the number of hidden-layer neurons is set to be 18, and TRAINGDX is chosen to be the training function [20–22]. TRAINGDX, which is named gradient descent with momentum and adaptive learning rate backpropagation, is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate. It will return a trained net and the training record. In addition, the maximum number of iterations in the training is set to 3000, the maximum number of validation failures is set to 6, and the error tolerance is set to 0.00001, which means that the training can be stopped if the error value is reached [23, 24]. Figure 4 shows the model structure graphic automatically generated by MATLAB.

To construct the Elman neural network, the MATLAB code can be like this. Firstly, three parameters in the Elmanet function are set, and codes are as follows:

$$\text{net} = \text{Elmanet}(1 : 2, 18, \text{'TRAINGDX'}). \quad (6)$$

Secondly, the maximum number of iterations in the training is set to 3000, and codes are as follows:

$$\text{net.trainParam.epochs} = 3000. \quad (7)$$

Thirdly, the maximum number of validation failures is set to 6, and the error tolerance is set to 0.00001, and codes are as follows:

$$\begin{aligned} \text{net.trainParam.max_fail} &= 6, \\ \text{net.trainParam.goal} &= 0.00001. \end{aligned} \quad (8)$$

Finally, initialize the network, and codes are as follows:

$$\text{net} = \text{init}(\text{net}). \quad (9)$$

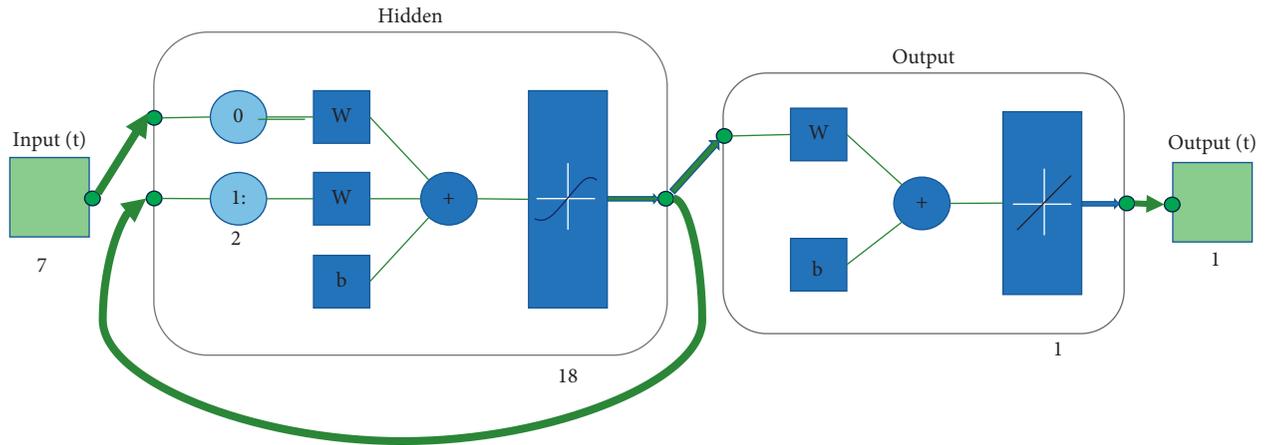


FIGURE 4: The model structure graphic generated by MATLAB.

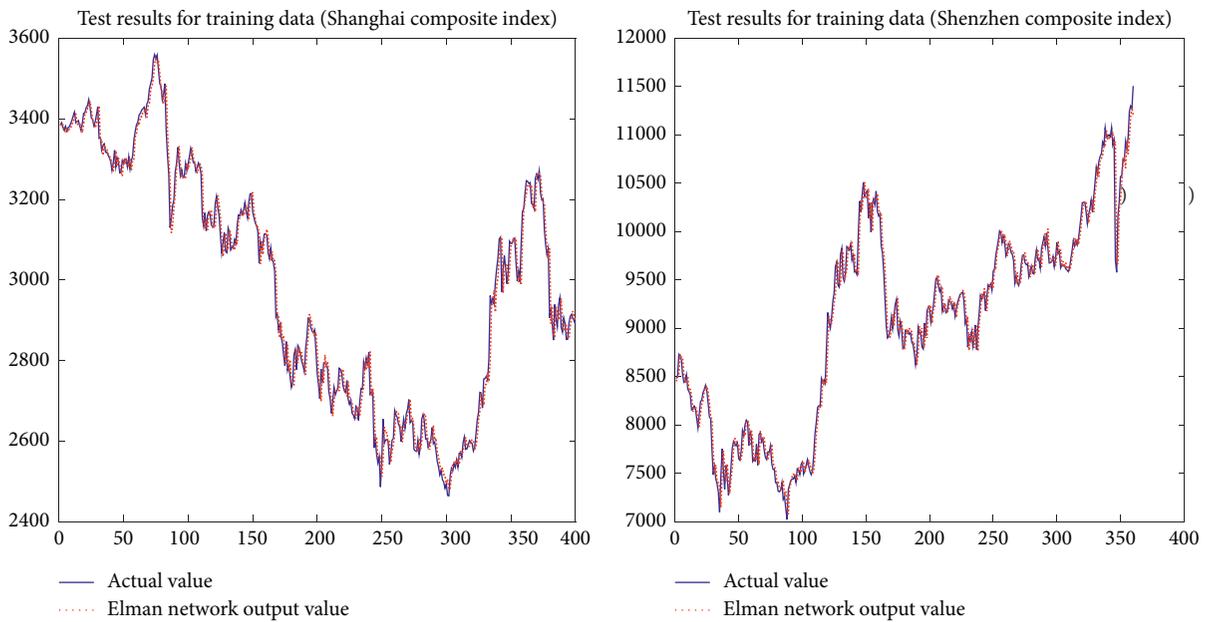


FIGURE 5: Test results for training data.

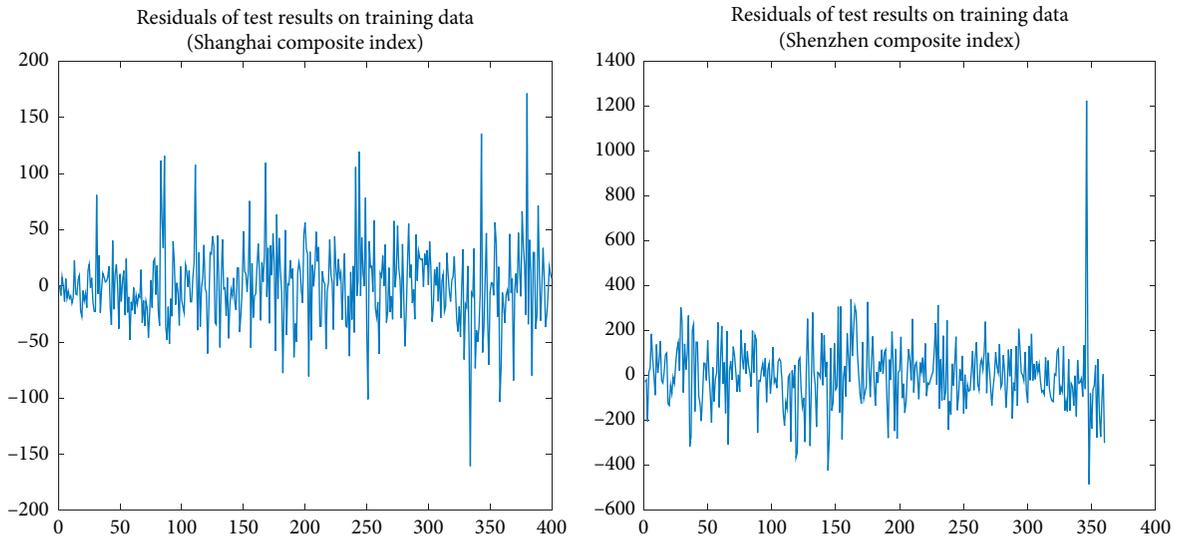


FIGURE 6: Residuals of test results on training data.

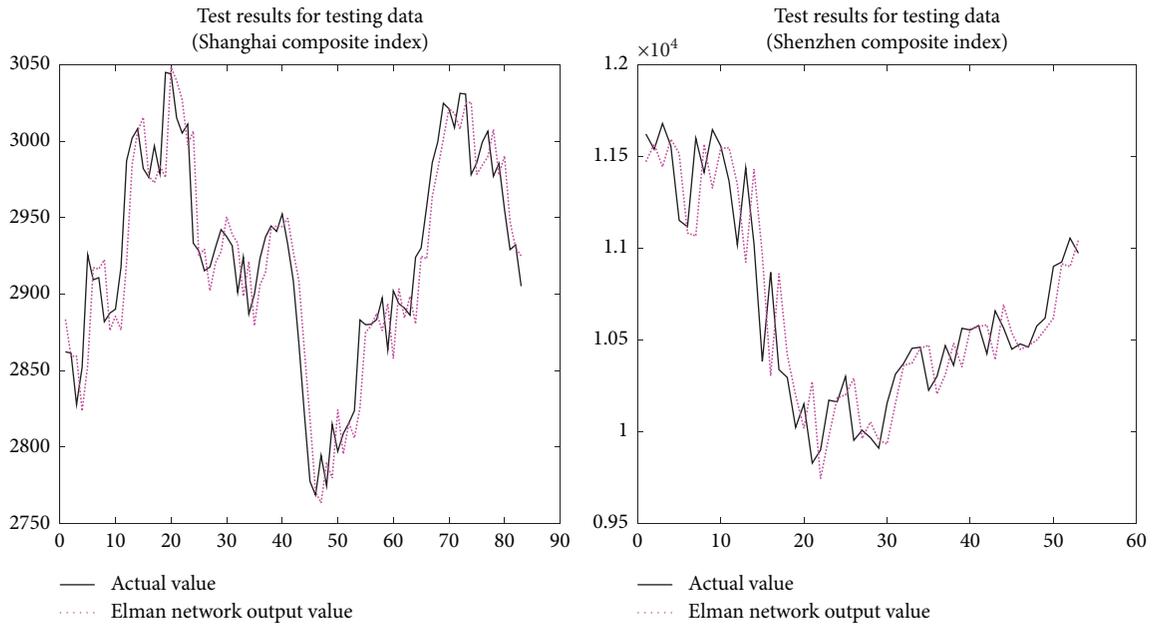


FIGURE 7: Test results for testing data.

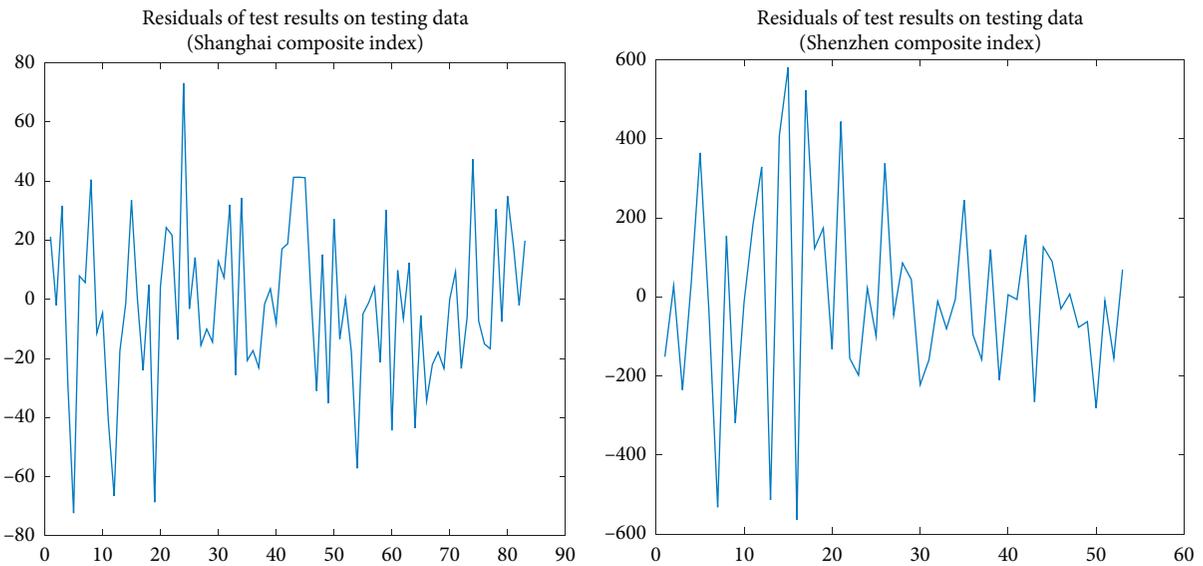


FIGURE 8: Residuals of test results on testing data.

TABLE 1: Relative error of test (Shanghai composite index).

Number	1	2	3	4	5	6	7
Relative error	0.006674	-0.001090	0.010719	-0.010057	-0.025153	0.001690	0.000677
Number	8	9	10	11	12	13	14
Relative error	0.012402	-0.003898	-0.002381	-0.015004	-0.023218	-0.007697	-0.002238
Number	15	16	17	18	19	20	21
Relative error	0.009203	-0.000263	-0.009339	0.000532	-0.022964	-0.000520	0.007789
Number	22	23	24	25	26	27	28
Relative error	0.006250	-0.004937	0.023693	-0.000488	0.004250	-0.005476	-0.003717
Number	29	30	31	32	33	34	35
Relative error	-0.005492	0.003612	0.001676	0.010246	-0.009194	0.010889	-0.007296
Number	36	37	38	39	40	41	42
Relative error	-0.006752	-0.008459	-0.001553	-0.000139	-0.003553	0.004742	0.005815
Number	43	44	45	46	47	48	49

TABLE 1: Continued.

Relative error	0.013786	0.014388	0.014745	0.000828	-0.010776	0.005480	-0.012749
Number	50	51	52	53	54	55	56
Relative error	0.009279	-0.004690	0.000227	-0.006596	-0.019610	-0.002958	-0.001138
Number	57	58	59	60	61	62	63
Relative error	0.000246	-0.007637	0.009974	-0.016051	0.002405	-0.002644	0.003361
Number	64	65	66	67	68	69	70
Relative error	-0.015557	-0.002587	-0.012756	-0.009015	-0.007153	-0.009234	-0.001688
Number	71	72	73	74	75	76	77
Relative error	0.002176	-0.008728	-0.003110	0.015200	-0.002282	-0.005979	-0.006169
Number	78	79	80	81	82	83	
Relative error	0.009414	-0.002398	0.010743	0.006008	-0.001105	0.006332	

TABLE 2: Relative error of test (Shenzhen composite index).

Number	1	2	3	4	5	6	7
Relative error	-0.013747	0.002825	-0.017177	0.004370	0.030084	-0.001853	-0.047231
Number	8	9	10	11	12	13	14
Relative error	0.017591	-0.017927	0.007011	0.010986	0.026179	-0.044038	0.035607
Number	15	16	17	18	19	20	21
Relative error	0.061603	-0.043408	0.047926	0.008911	0.028250	-0.016142	0.039523
Number	22	23	24	25	26	27	28
Relative error	-0.012140	-0.021609	0.001765	-0.010977	0.036256	-0.005063	0.006179
Number	29	30	31	32	33	34	35
Relative error	0.003674	-0.022697	-0.014897	-0.002466	-0.006432	0.001260	0.023451
Number	36	37	38	39	40	41	42
Relative error	-0.009376	-0.016781	0.009971	-0.019686	0.001902	-0.002325	0.014443
Number	43	44	45	46	47	48	49
Relative error	-0.024677	0.009285	0.008341	-0.003239	-0.000341	-0.010959	-0.005216
Number	50	51	52	53	54	55	56
Relative error	-0.026052	-0.000483	-0.012797	0.007071			

After all the above steps, construction of Elman neural network is completed [25–27].

5. Experiments

5.1. Training of the Supposed Model. When the Elman neural network is built, the model can be trained, but all the data has to be normalized first considering of the performance and stability of the model. The normalization operation can use the `mapminmax` function provided by MATLAB toolbox, and the default normalization interval of `mapminmax` function is $[-1, 1]$. The detailed MATLAB code is as follows:

$$\begin{aligned}
 [\text{train } x_1, \text{st}_1] &= \text{map min max}(\text{train } x), \\
 \text{train_ty}_1 &= \text{sim}(\text{net}, \text{train } x_1), \\
 \text{train_ty} &= \text{map min max}(\text{'reverse'}, \text{train_ty}_1, \text{st}_2).
 \end{aligned}
 \tag{10}$$

After the normalized operation of training data, `trainx` and `trainx1` were obtained. The normalized training data (`trainx1`) were input into the network model to obtain the current network output (`train_ty1`) and then reversely normalized into normal data to obtain `train_ty`, which is the corresponding stock price of the training data. What we want to emphasize is that the data used in the test should be normalized first and then the output should be unnormalized.

5.2. The Test Results and the Quantitative Analysis. Figure 5 shows a graph of the actual and predicted values; the blue solid line is the actual value and the red dotted line represents the Elman network output value. Apparently, the model fits the training data well. In addition, we further calculated the residuals of test results on training; Figure 6 shows the residuals of training results on training data, and residual in mathematical statistics refers to the difference between the actual observed value and the estimated value (the fitting value).

Figure 7 shows a graph of the actual and predicted values; the black solid line is the actual value and the red dotted line represents the Elman network output value. In addition, we further calculate the residuals of test results on testing data; Figure 8 shows the residuals of test results on testing data. And, the relative errors of each prediction are also calculated for further study and analysis. All relative error values are shown in Tables 1 and 2. By analysing these graphs and data, it is clear that the prediction effect of the model is pretty good.

6. Conclusions

This study is based on a basic premise that the historical stock price will have a great impact on the future short-term stock price. On this premise, we established an improved Elman model and collected the historical data of the Shanghai composite index and the Shenzhen composite

index as a dataset for the experiment. As for dataset processing, we divided two datasets, one for training and the other for testing. In addition, the data were normalized. Regarding model building, we take MATLAB as the platform, and set the number of hidden-layer neurons to be 18. TRAINGDX is chosen to be the training function. In terms of training, the maximum number of iterations in the training is set to 3000, the maximum number of validation failures is set to 6, and the error tolerance is set to 0.00001. Finally, we use the model to test the training data and the test data. In order to analyse the experimental results, we also calculated relative error and the residuals and drew a picture to show them. Based on Elman network, this study predicted the short-term stock price in the future and achieved a good prediction effect. However, it is unrealistic to predict the long-term stock price in the future, which is difficult to achieve [28–30]. This study provides an effective experimental method for predicting the near future stock price.

Data Availability

All of the data used in this study are already available on the Internet and is easily accessible.

Conflicts of Interest

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

Acknowledgments

This work was supported by the key project of the Natural Science Research of Higher Education Institutions in Anhui Province (Grant no. KJ2018A0461); the Anhui Province Key Research and Development Program Project (Grant no. 201904a05020091); the Provincial Quality Engineering Project from Department of Education Anhui Province (Grant no. 2019mooc283); and the Domestic and Foreign Research and Study Program for Outstanding Young Backbone Talents in Colleges and Universities (Grant no. Gxgnfx2019034).

References

- [1] Q. Shayea, *Neural Networks to Predict Stock Market Price*, World Congress on Engineering and Computer Science, San Francisco, CA USA, 2017.
- [2] X. Xu, B. Shen, X. Yin et al., “Edge server quantification and placement for offloading social media services in industrial cognitive IoV,” *IEEE Transactions on Industrial Informatics*, 2020.
- [3] V. Rohit, C. Pkumar, and S. Upendra, “Neural networks through stock market data prediction,” in *Proceedings of the 2017 International Conference of Electronics*, Coimbatore, India, April 2017.
- [4] D. Das, A. S. Sadiq, N. B. Ahmad, and J. Lloret, “Stock market prediction with big data through hybridization of data mining and optimized neural network techniques,” *Journal of Multiple-Valued Logic and Soft Computing*, vol. 29, no. 1-2, pp. 157–181, 2017.
- [5] J. Zahedi and M. Rounaghi, “Application of artificial neural network models and principal component analysis method in predicting stock prices on tehran stock exchange,” *Physica A: Statistical Mechanics and its Application*, vol. 38, pp. 178–187, 2015.
- [6] X. Han, “Stock price prediction with neural network based on MATLAB,” *Systems Engineering*, 2003.
- [7] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, “Be-Come: blockchain-enabled computation offloading for IoT in mobile edge computing,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.
- [8] R. Mahanta, T. N. Pandey, A. K. Jagadev, and S. Dehuri, “Optimized radial basis functional neural network for stock index prediction,” in *Proceedings of the IEEE Conference Publications*, Chennai, India, March 2016.
- [9] <https://www.mathworks.com/help/deeplearning/ref/Elmnet.html>.
- [10] Y. Zhang, G. Cui, S. Deng et al., “Efficient query of quality correlation for service composition,” *IEEE Transactions on Services Computing*, p. 1, 2018.
- [11] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. Alam Bhuiyan, “Adaptive computation offloading with edge for 5G-environmented Internet of connected vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [12] Y. Zhang, C. Yin, Q. Wu et al., “Location-aware deep collaborative filtering for service recommendation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems (TSMC)*, 2019.
- [13] J. Yu and P. Guo, “Stock price forecasting Model Based on Improved Elman Neural Network,” *Computer Technology and Development*, 2008.
- [14] H. Shi and X. Liu, “Application on stock price prediction of Elman neural networks based on principal component analysis method,” in *Proceedings of the 2014 11th International Computer Conference on Wavelet Active Media Technology & Information Processing*, Chengdu, China, December 2014.
- [15] X. Zhang, S. Qu, J. Huang, B. Fang, and P. Yu, “Stock market prediction via multi-source multiple instance learning,” *IEEE Access*, vol. 6, no. 99, pp. 50720–50728, 2018.
- [16] M. Billah, S. Waheed, and A. Hanifa, “Predicting closing stock price using artificial neural network and adaptive neuro fuzzy inference system (ANFIS: the case of the Dhaka stock exchange,” *International Journal of Computer Applications*, vol. 129, no. 11, pp. 1–5, 2015.
- [17] https://www.mathworks.com/help/deeplearning/index.html?s_tid=CRUX_lftnav.
- [18] Z. Zhang, Y. Shen, and G. Zhang, “Short-term prediction for opening price of stock market based on self-adapting variant PSO-elman neural network,” in *Proceedings of the IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, November 2017.
- [19] V. Andrea and L. Karel, “MatConvNet-convolutional neural networks for MATLAB,” in *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 689–692, Brisbane, Australia, October 2015.
- [20] L. Ren, Y. Liu, Z. Rui, H. Li, and R. Feng, “Application of elman neural network and MATLAB to load forecasting,” in *Proceedings of the International Conference on Information Technology and Computer Science*, Kiev, Ukraine, July 2009.
- [21] K. Kim and W. Lee, “Stock market prediction using artificial neural networks with optimal feature transformation,” *Neural Computing & Applications*, vol. 13, no. 3, pp. 255–260, 2004.

- [22] H. Grigoryan, "Stock market prediction using artificial neural networks. Case Study of TALIT, Nasdaq OMX Baltic Stock," *Database Systems Journal*, 2015.
- [23] S. Nayak, B. Misra, and H. Behera, "An adaptive second order neural network with genetic-algorithm-based training (ASONN-GA) to forecast the closing prices of the stock market," *International Journal of Applied Metaheuristic Computing*, vol. 7, no. 2, pp. 39–57, 2016.
- [24] Y. Zhang, K. Wang, Q. He et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, 2019.
- [25] P. Kai, H. Huang, S. Wan, and V. Leung, "End-edge-cloud collaborative computation offloading for multiple mobile users in heterogeneous edge-server environment," *Wireless Network*, vol. 2020, 2020.
- [26] C. Goutami and S. Chattopadhyay, "Monthly sunspot number time series analysis and its model construction through autoregressive artificial neural network," *The European Physical Journal Plus*, vol. 127, no. 4, 2012.
- [27] Z. Guo, J. Wu, H. Lu, and J. Wang, "A case study on a hybrid wind speed forecasting method using BP neural network," *Knowledge-Based Systems*, vol. 24, no. 7, pp. 1048–1056, 2011.
- [28] Y. Zhang and L. Wu, "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network," *Expert Systems with Applications*, vol. 36, no. 5, pp. 8849–8854, 2008.
- [29] X. Han, "Stock Price Prediction with Neural Network Based on MATLAB," *Systems Engineering*, vol. 2003, 2003.
- [30] K. Peng, B. Zhao, S. Xue, and Q. Huang, "Energy- and resource-aware computation offloading for complex tasks in edge environment," *Complexity*, vol. 2020, Article ID 9548262, 14 pages, 2020.