2019 IEEE 4th International Conference on Cloud Computing and Big Data Analytics

Stock Prediction Based on LSTM under Different Stability

Fei Qian
School of Information Science and Technology
University of Science and Technology of China
Hefei, China
e-mail: fei123@mail.ustc.edu.cn

Xianfu Chen
School of Information Science and Technology
University of Science and Technology of China
Hefei, China

Abstract—**The boom of Big Data has made the development of prediction algorithms more intelligent, so the studies have gradually shifted from the traditional linear prediction algorithm (a typical representative of time-series prediction algorithm) to the popular deep learning prediction algorithm. The nonlinear deep learning algorithm can better reflect the changeable internal laws and external relations of data, especially for complex stock price data. Long Short Term Memory network (LSTM) is a special algorithm for processing time-series problem. In this work, we conducted a stationary analysis of the stock's time-series data and then used the LSTM neural network algorithm to predict stock data under different stationary conditions, and performed statistical analysis on multiple experimental data. In addition, an ARIMA algorithm was introduced to compare with the LSTM. A large number of experimental results show that the LSTM neural network prediction algorithm has higher prediction accuracy and is not sensitive to the stability response.**

Keywords-prediction; deep learning; LSTM; stationarity; ARIMA

I. INTRODUCTION

Intuitively, the stationary time-series is a sequence whose statistical characteristics will not change with time shifting (such as mean and covariance), or means there is no obvious tendency or periodicity [1]. The stable sequence has a short-term correlation, indicating that only the recent sequence values have a significant effect on the present value for the stationary sequence, and the farther the interval is, the smaller the influence of the past value on the current value. In data prediction and analysis, it is generally necessary to detect the stationarity of the data, which is beneficial for us to choose a better prediction algorithm. For example, the Auto Regressive Moving Average (ARMA) model requires the time-series to be stationary before modeling; however, a differential operation is generally adopted for non-stationary time-series, which known as Autoregressive integrated moving Average (ARIMA) model [2, 3].

Stock price forecasting is the most common forecast based on time-series data in financial market. Stock price data are noisy, complex and non-linear which is easy to be affected by many factors such as policy, economy and psychology. Therefore, we need to conduct data preprocessing and analysis before the prediction (e.g. stationarity detection). Traditional prediction methods cannot accurately capture historical information, which often rely on linear regression and parameter estimation. Therefore, most financial prediction tends to build nonlinear models, such as SVM and deep learning neural network models.

Deep learning is a method of machine learning which developed from the original neural network [4]. In recent years, more and more scholars have been trying to solve time-series problem based on deep learning methods. Recurrent Neural Network (RNN) [5] adds the concept of time to its network structure, making it dedicated to processing time-series data. However, the effect is not very good in actual use. If the sequence is too long, the vanishing gradient problem will occur during optimization [6]. In order to solve this problem, Sepp Hochreiter and Jürgen Schmidhuber proposed a Long Short Term Memory network in 1997 [7]. LSTM is a special recurrent structure that improves the ability of the RNN network to handle long-term dependent tasks and is resistant to vanishing gradient problem. The application of various variants of the RNN network to natural language processing problems (NLP) has been very prominent, such as Graves' successful application of Bidirectional Recurrent Neural Network (BRNN) in handwriting recognition [8] and speech recognition [9]. Similarly, Google deployed two layers of deep LSTM [10] in large-scale speech recognition, and its model achieved advanced results. Recently, some scholars have applied LSTM to stock prediction and achieved good results. For example, a time-weighted LSTM model to redefine stock trend forecasts was proposed in [11]. The bidirectional LSTM and the stacked LSTM with the simple LSTM were compared, and the results showed that the performance of the bidirectional LSTM is the best in stock prediction in [12]. Additionally, in [13], a multi-branch LSTM stock short-term prediction model based on K-means clustering was constructed, and its prediction results showed higher accuracy. In these models, scholars have confirmed the superiority of the LSTM prediction model over traditional prediction models, which can be applied to various nonlinear time-series data, but have not studied the influence of time-series stationary difference on prediction results. This paper mainly compares the influence of stock data under different stability conditions on the prediction results of LSTM model.

The rest organization of this paper is as follows. Section 2 introduced the time-series stability detection methods, the LSTM structure and the stock price prediction model used in this paper. Section 3 collected relevant data for experiments

978-1-7281-1410-1/19/$31.00 ©2019 IEEE

and record experimental results. Finally, conclusions and future work were reached in Section 4.

II. THEORETICAL FOUNDATION AND MODEL DESIGN

In this part, we first introduced the stationarity analysis methods of time-series data, then presented the structure of LSTM, and finally designed the stock price prediction model and error analysis methods for the experiment in this paper.

A. Stationarity Detection of Time-Series

Time-series prediction is to predict future values by observing historical data, and stationarity is a prerequisite for most time-series prediction. Common stationary detection methods include the observation method and the unit root test method. The observation method is mainly to check whether the original diagram of the sequence has obvious tendency or periodicity, or to compare it with the properties of autocorrelation function (ACF) [14] and partial correlation function (PACF) [15, 16]. The Augmented Dickey-Fuller test (ADF) is one of the most commonly used unit root test methods proposed by Dickey-Fuller in 1979 [17]. In addition to these two categories, some scholars have studied intelligent classification methods, such as clustering time-series into stationary and non-stationary automatically [18]. However, this paper does not do an in-depth study of these, but use it as a tool to detect stationarity in experiments.

B. The Long Short Term Memory Network

Long Short Term Memory network (LSTM) is an upgraded version of RNN. Different from the traditional neural network, there are connections between the hidden layers of RNN. Therefore, the input of the hidden layer not only includes the input of the input layer but also the output of the hidden layer at the previous moment. The expanded structure of RNN is shown in Figure 1. LSTM and RNN have a similar expanded structure, but the memory cell structure of their hidden layer is different. Forget gate, input gate and output gate has been added into the hidden layer's memory cell of LSTM based on the structure of RNN, and the design of these three special gate structures effectively solves the vanishing gradient problem, making it very suitable for dealing with long-term dependency problems. The LSTM memory cell structure of the hidden layer is shown in Figure 2.

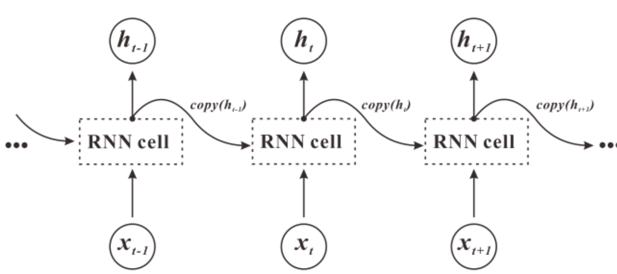

Figure 1. The expanded structure of RNN

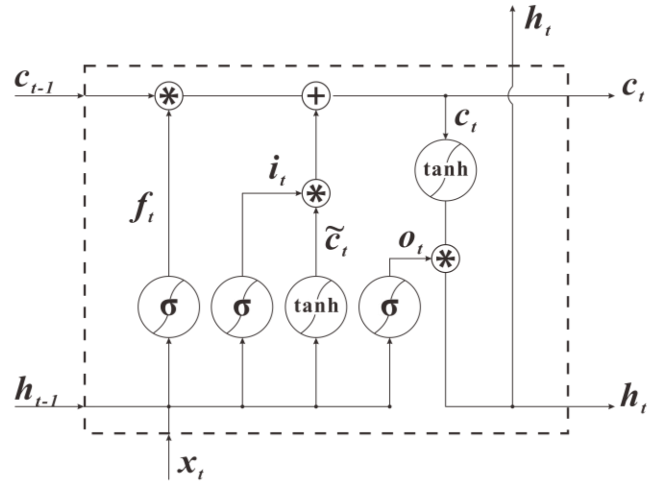

Figure 2. LSTM memory cell structure of the hidden layer

We can describe the information processing process of LSTM by the following 8 equations:

$$i_t = \sigma\left(W_i x_t + H_i h_{t-1} + b_i\right) \quad (1)$$

$$f_t = \sigma\left(W_f x_f + H_f h_{t-1} + b_f\right) \quad (2)$$

$$o_t = \sigma\left(W_o x_t + H_o h_{t-1} + b_o\right) \quad (3)$$

$$\tilde{c}_t = \tanh\left(W_c x_t + H_c h_{t-1} + b_c\right) \quad (4)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (5)$$

$$h_t = o_t * \tanh\left(c_t\right) \quad (6)$$

$$\sigma\left(x\right) = \frac{1}{1+e^{-x}} \quad (7)$$

$$\tanh\left(x\right) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (8)$$

On the left side of the equation:

- i_t is the input gate and determines which information needs to be updated in the cell
- f_t is the forget gate and determines which information should be dropped from the cell.
- o_t is the output gate which determines how much information is output.
- $\tilde{c}_t$ is the candidate value for the states of the memory cell at time t.
- c_t is the state of the current memory cell at time t, which calculated by the combination of i_t and $\tilde{c}_t$, f_t and c_{t-1} through the element-wise multiplication (*).
- h_t is the output value filtered by the output gate.
- σ denotes sigmoid function with the range 0 to 1, tanh function is used to put the value between -1 and 1.

On the right side of the equation:

- x_t is the input to the memory cell at time t.
- W_i , W_f , W_o , W_c , H_i , H_f , H_o and H_c are weight matrices.
- b_i , b_f , b_o and b_c are bias vectors.

C. Model Design

In order to verify the influence of stability difference on LSTM prediction algorithm, we selected several sets of stock price data with different stability first by using the detection method of ADF [19] in python. The stability is closely related to the size of P value in the detection result, and with increasing P value, the stability decreases significantly. Then, LSTM prediction model was designed for stock price data, and implemented by the framework of TensorFlow. Finally, we compared LSTM model with ARIMA model. In this experiment, we use Mean Absolute Error (MAE) in (9) and Root Mean Square Error (RMSE) in (10) to assess the performance and describe errors between the predicted value and actual value. In following equation y_i and y_i' denote actual value and predictive value respectively.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - y_i'\right| \tag{9}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\left|y_i - y_i'\right|\right)^2} \tag{10}$$

III. Experiments and Results

In this experiment, we selected three stocks with similar trend and different stability (399300, 399007, 399009), and extracted the last 1000 days' dataset of each stock. We divided each stock data into training set and test set (7:3) according to the method of hold-out. The training set was used to train the prediction model, and the test set was used to test the quality of the prediction model. Since the dataset range is different for different samples, we firstly standardized the dataset by the method of Min-Max, which can reduce the range of each data to 0-1, and then LSTM model was employed to test stock price data. Finally, the records of 10-repeated experiments for each dataset were counted, and the results were shown in Table I. Although the difference of P value reflects the stability of data, the influence of P value on the predicted results of three data can be ignored almost. In other words, LSTM has strong adaptability to data with different stability.

TABLE I. Average Error of 10 Experiments

Datasets	P value	Train MAE	Train RMSE	Test MAE	Test RMSE
399300	0.01504	0.02136	0.04140	0.02227	0.02759
399007	0.08020	0.02413	0.04137	0.01648	0.02004
399009	0.20615	0.03759	0.06119	0.01897	0.02122

Keeping the five decimal places

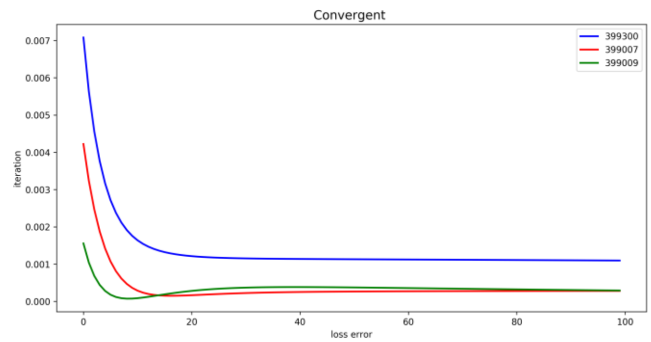

Figure 3. Comparison of convergence.

Three convergent data that was closest to the average error in all experimental results were selected, and their convergence comparison diagram is shown in Figure 3. From this figure, we can see that the biggest P value's convergence rate is slightly faster, but it is unobvious.

Here we had set the learning rate to 0.001, timestep to 5, and the number of hidden layer's node to 10.

In the end, we predicted the same dataset with LSTM model and ARIMA model. The result was shown in figure 4. The RMSE of LSTM model in this experiment is 0.0374, and the RMSE of ARIMA model in this experiment is 0.10158, which is larger. We can see that the LSTM prediction shows more details about the actual data, so the LSTM model performed better.

IV. Conclusion and Future Work

In this paper, we have studied the influence of stability's difference on LSTM stock price prediction. The stability of experimental results has little effect on the prediction results, but it has a slight influence on the convergence speed of the algorithm. The greater the P value is, the faster the convergence speed will be. At last, we conducted a comparative experiment with the LSTM model and the ARIMA model, and the results showed that the error rate of the LSTM algorithm was 66.78% lower than that of ARIMA. Therefore, we can draw a conclusion that LSTM algorithm performs better in prediction and has smaller errors. However, the disadvantage of LSTM algorithm is that it takes a lot of time to train the model and requires large sample of data.

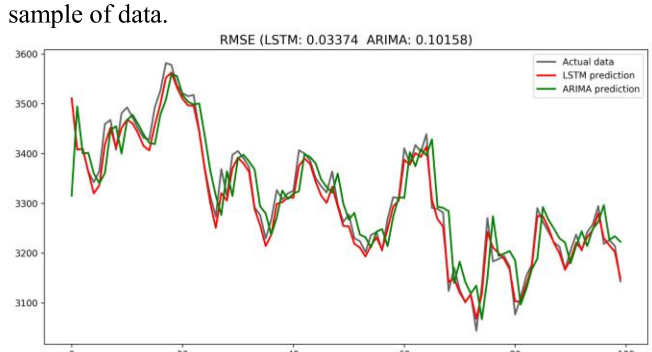

Figure 4. Comparison of the predicted results of LSTM and ARIMA

For future work, we plan to focus on improving the performance of LSTM algorithm, hoping that it cannot only show higher prediction accuracy, but also improve the training speed and apply to other fields.

ACKNOWLEDGMENT

This work is supported by National Key R&D Program of China (No.2018YFC0808600).

REFERENCES

[1] Jian feng ZHAO, "Modeling of Software Aging Based on Non-stationary Time Series," in International Conference on Information System and Artificial Intelligence, 2016, pp. 176-180

[2] Box, G. E. P. And Jenkins, G. M. (1976). Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco

[3] Javier Contier, "ARIMA Models to Predict Next-Day Electricity Price," in IEEE Transactions on Power Systems 2003, 18(3): pp. 1014-1020

[4] Gododfellow I, Bengio Y, Courville A. Deep learning [M]. The MIT Press, 2016

[5] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986c). Learning representations by back-propagating errors. Nature, 323, pp. 533-536.

[6] Gustavsson A, Magnuson A, Blomberg B, et al. On the difficulty of training Recurrent Neural Networks[J]. Computer Science, 2013.

[7] Sepp Hochreiter, Jurgen Schmidhuber. Long short-term memory[J]. Neural Computation. 9(8): pp. 1735-1780, 1997.

[8] Graves, A. and Schmidhuber, J. (2009). "Offline handwriting recognition with multidimensional recurrent neural networks," In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, NIPS'2008, pp. 545-552.

[9] Graves, A., Mohamed, A., and Hinton, G. (2013). "Speech recognition with deep recurrent neural networks," In ICASSP'2013, pp. 6645-6649.

[10] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in Fifteenth Annual Conference of the International Speech Communication Association, 2014

[11] Zhiyong Zhao, Ruonan Rao and Shaoxiong Tu, "Time-weighted LSTM Model with Redefined Labeling for Stock Trend Prediction," in International Conference on Tools with Artificial Intelligence, 2017 IEEE, pp. 1210-1217.

[12] Khaled A. Althelaya, El-Sayed M. El-Alfy, Salahadin Mohammed, "Evaluation of Bidirectional LSTM for Short- and Long-Term Stock Market Prediction," in 9th International Conference on Information and Communication Systems (ICICS), 2018 IEEE, pp. 151-156.

[13] XiuLi Shao, Doudou Ma, and Yiwei Liu. "Short-term Forecast of Stock Price of Multi-branch Lstm Based on K-means," in 4th International Conference on Systems and Informatics (ICSAI 2017), 2017 IEEE, pp. 1546-1551.

[14] Galeano, P., Pe, D., Multivariate analysis in vector time series. Resenhas 4, pp. 383-404, 2000.

[15] Cleveland W.S., The inverse autocorrelations of a time series and their applications. Technometrics 14, pp. 277-293, 1972.

[16] Chatfield C., Inverse autocorrelations. J. Roy. Statist. Soc. Ser. A 142, pp. 363-377, 1979.

[17] Dickey, D.A and Fuller, W.A. Distribution of the estimators for autore gressive time series with aunit root. Journal of the American Statistic al Ass ociation, 1979, 74: pp. 427-431.

[18] Heshan Guan, Shuliang Zou and Mengya Liu. "Clustering Univariate Time Series into Stationary and Non-stationary," in Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2010 IEEE, pp. 2782-2785.

[19] http://statsmodels.sourceforge.net/stable/generated/statsmodels.tsa.stattools.adfuller.html