Research on Stock Price Prediction Model Based on LSTM

Junrong Nie*

Department of Computer Science and Technology, Dongguan University of Technology, Dongguan, 523129, China

*Corresponding author: junrongn@dgut.edu.cn

Abstract. Stock prices are a non-linear, long-term series of data. The research topic of this paper is to use the Long Short-Term Memory Network (LSTM) model to analyze the historical stock price data of BYD (BYD) to predict the future stock price trend, and to provide intelligent reference for investors by evaluating the model performance, aiming to improve the accuracy of stock price prediction. The accuracy and consistency of stock price prediction can be increased by using this method, which successfully takes advantage of the time characteristics in historical data. After explaining the idea and architecture of LSTM, this article creates a stock price prediction model based on it, using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as evaluation metrics. At the same time, network structures with different layers were designed to verify the effectiveness of the model, and the experimental results were shown. The advantages and limitations of the model are highlighted, and the feasibility of deep learning in the stock financial market and the credibility and advantages of strategic decision-making are demonstrated.

Keywords: Stock price prediction; LSTM; Neural network; Sliding windows.

1. Introduction

An enterprise's stock price is a crucial metric for assessing its profitability, as well as the macroeconomic environmental conditions, market supply and demand, and several other aspects. With the increasing complexity of financial markets and the massive growth of financial data, stock price prediction has become a challenging problem. The study of financial market patterns has received a lot of attention lately in the domains of mathematics, engineering, and finance. [1]. Models for predicting stock prices are also constantly iteratively updated, mainly divided into traditional models and modern models [2]. Conventional models like the autoregressive integrated moving average model (ARIMA) and the autoregressive model (AR) [3]. Autoregressive conditional heteroskedasticity model (ARCH), Markov chain model, gray theory, etc. [4]. Among them, a unique type of recurrent neural network (RNN) that works well for processing data with temporal dependency is the long short-term memory network (LSTM) [5]. Due to the time series nature of stock prices, LSTM models are excellent at predicting stock prices. With its headquarters located in Shenzhen, BYD is a Chinese car and battery manufacturing firm. It is among the top producers of new energy cars worldwide. Electric passenger cars, electric buses, electric logistics trucks, electric taxis, smart highways, and power grid energy storage are among the company's primary business activities. The macroeconomic climate, market trends, business performance, investor mood, and other variables may have an impact on the share price.

BYD's stock price prediction helps investors seize opportunities and steer clear of dangers. Reorienting enterprises for corporate decision-makers and developing smart financial plans are made easier with accurate stock price forecasts.

This study analyzes historical stock price data for BYD using the LSTM model to forecast future price patterns. This helps investors make smarter investment decisions and gives corporate decision-makers useful reference data. The purpose of this article is to anticipate the price of BYD stock using the LSTM model and to assess the model's performance using mean absolute error (MAE) and root mean square error (RMSE).

2. Introduction to the Structure and Principles of LSTM

Deep learning has been developing rapidly in recent years, and different neural network models are used in various fields. The topic of time series data prediction employs a number of neural network model types, such as the RNN model, LSTM model, and Fuzzy Neural Network (FNN) model. The LSTM model is a recurrent neural network improved by the RNN model. The LSTM can effectively transmit and express information in a longer time series. This successfully resolves the RNN's gradient disappearance issue brought on by the addition of network layers and the passage of time, making it more appropriate. The unique neuron cell state of the LSTM neural network topology, which records and transmits data using memory cells, sets it apart from other deep learning techniques. It adopts the same chain structure as RNN, consisting of three large threshold units: forgetting gate, input gate, and output gate, and uses the sigmoid function and tanh function to update the unit state.

In Figure 1, the LSTM structural diagram is displayed. A vector from a node's output to another node's input is represented by each line. Vector addition and other point-wise operations are represented by the pink circles. The layer of the trained neural network is represented by the orange-yellow rectangle. The intersection of lines signifies the joining of vectors, while the separation of lines denotes the repetition of information at various points.

The first gating unit of LSTM is the forgetting gate. f_t represents the gate value. The higher layer's cell state will be impacted by the forgetting gate value. It determines what information the neuron should forget according to the input, and the activation function Sigmoid will compress the value between 0 and 1. As shown in Equation 1.

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

The input gate, which is connected to the value that has to be updated, is the second gating unit. The degree of information update is determined according to the Sigmoid function. As shown in Equation 2. The value passing through the tanh layer will generate a new value $\tilde{C}_t$ to determine the updated content. As shown in Equation 3.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(w_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

Before passing through the third gate control unit, the old neuron state C_{t-1} needs to be updated to C_t. The input gate value, the current time, and the preceding time C_{t-1} are multiplied by the acquired forgetting gate value. The new neuron state C_t is obtained by doubling the obtained un-updated C_t. As demonstrated by Equation 4.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

The third gating unit is the output gate, which determines the output based on the neuron state. Equation 5 illustrates how the output gate value is first computed to identify which portion of the neuron emits the output portion that the Sigmoid layer judges. The sequence information output at time t may then be obtained by processing the neuron using the tanh function and multiplying it by the preceding gate output, as indicated by Equation 6.

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

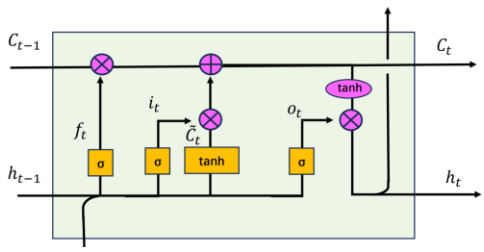

Fig. 1 LSTM structure diagram [6].

3. Model Building

The laptop running Microsoft Windows was used to test the LSTM-based stock prediction model. The GPU version of the PyTorch framework was built. The framework can seamlessly move the AI model from research to production without having to deal with migration. Advantages include equipment deployment, quantitative models to accelerate inference, and front-end improvements [7]. This experiment constructed a two-layer LSTM neural network model to assess and forecast the same stock for stocks where the data scale is not very huge. RMSE and MAE are used by the model's prediction performance assessment indicators to compare experimental findings. The following are the RMSE and MAE calculation formulas:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(X_{prediction,t} - X_{real,t}\right)^2} \quad (7)$$

$$MAE = \frac{1}{N}\sum_{I=1}^{N}\left|\left(X_{prediction,i} - X_{real,i}\right)\right| \quad (8)$$

4. Experiment and Result Analysis

4.1. experiment Process

The experiment's training procedure is carried out in many stages, as seen in Figure 2: obtaining data, data preprocessing, hyperparameter setting, training the model, adjusting parameters, and stock price forecast. [8].

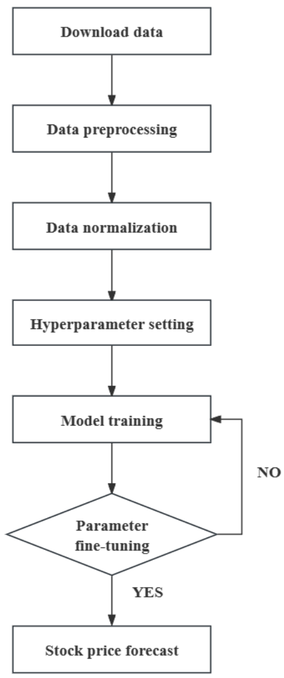

Fig. 2 Sequence of steps (Photo/Picture credit: Original).

4.2. Data Selection

The prediction object in this article is BYD's closing price on the Chinese stock market. The sample data are all from the Kaggle website, and the period is from January 5, 2015, to November 22, 2023. The prediction model's initial data set comprises 2162 trading days, primarily comprising the trading day (TradingDay), opening price (OpenPrice), closing price (ClosePrice), highest price (HighPrice), lowest price (LowPrice), increase or decrease (Return), and so through. This paper selects its closing price as shown in the figure 3. The closing price includes stages of rise, fall, and adjustment, reflecting changes in the company's operating level. To emphasize its predictive power, this experiment uses 90% of the data as the training set and 10% as the test set.

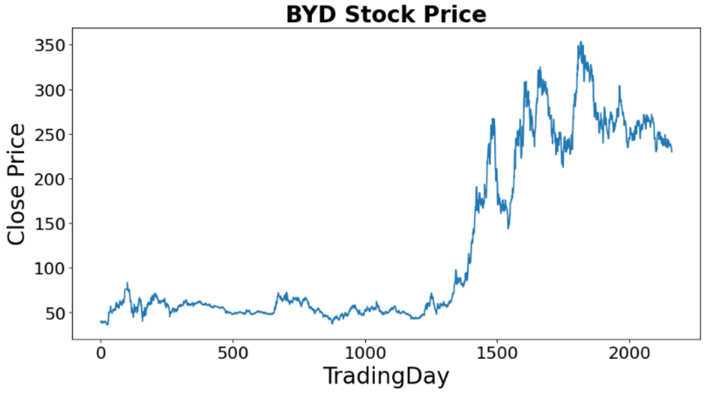

Fig. 3 Open Price of BYD Stock (Photo/Picture credit: Original).

4.3. Data Processing

The obtained original data set has dates in reverse order or has null values, etc. To collect regular stock time series data and create a more comprehensive and useful data set, the downloaded data set has to be sorted by date and null values deleted.

Data normalization: The price of the stock is input as a feature value, and the values are scaled to the range of -1 to 1. Feature scaling is to improve the performance and stability of the model, because different features may have different scales or ranges, and direct model training may cause some features to dominate the learning process, so the feature sequence needs to be normalized [9].

4.4. Sliding Window Settings

Because the LSTM neural network is a time series model, how different sliding windows are configured affects how accurate the prediction results are [10]. The prediction results have a considerable variation and some data fluctuations when the sliding window is 5, as Table 1 illustrates. This is because the time step that the sliding window considers is shorter and the effect of global variables is not taken into consideration. When set to 30, the time range taken into consideration is excessively broad, making it easy to overlook the influence of eigenvalues over a brief period and producing erroneous prediction outcomes. When the sliding window is set to 20, the error of the prediction result is smaller, reflecting its higher accuracy and suitability for stock prediction.

Table 1. Comparison of three different sliding windows

Sliding window size	RMSE	MAE
5	6.62	5.24
20	5.95	4.43
30	6.06	4.54

4.5. Experimental Results

To forecast stock values, first construct a fully linked layer model and a single-layer LSTM network. Following several experiment tests and debugging, after assessing the model's prediction accuracy and computation volume. In the feedforward network layer, 128 hidden neurons were used, with a learning rate of lr=0.01. The Adam optimizer is also used. In this method, the step size and learning rate are randomly determined by means of random sampling from the probability distribution. Every gradient training parameter's adaptive learning rate is estimated, and samples from the optimization procedure are utilized to better focus the information search [11]. A single layer and a double layer are used in this experiment to evaluate the neural network's performance.

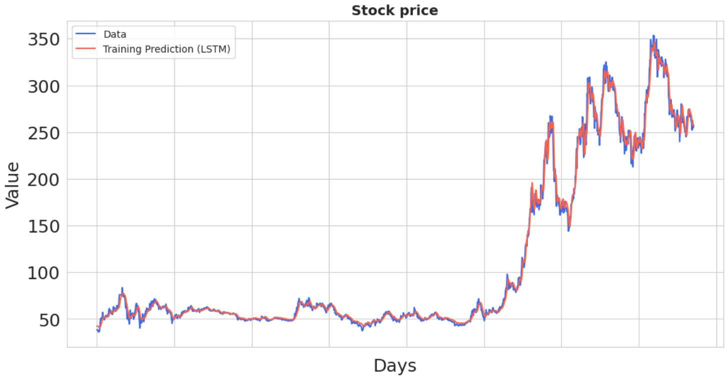

Fig. 4 Single-layer LSTM network model prediction result (Photo/Picture credit: Original).

Table 2. Single-layer LSTM network model prediction result

Performance	RMSE	MAE	Train Time(second)
value	6.13	4.54	41.5

By observing Table 2 and Figure 4, research can find that the single-layer LSTM network has large prediction errors. The large gap with actual prices suggests that the network model may not adequately capture the complex relationships in the data. In an attempt to remedy this, the experiment chose to further modify the neural network's architecture, attempting to employ both a fully connected layer model and a two-layer LSTM network.

Thus, a fully linked layer model and a double-layer LSTM network were subsequently built in response to the performance of the single-layer LSTM network being observed. As shown in Figure 5, to keep the complexity of the model relatively consistent, experiments ensured that the new model had the same number of neurons as a single-layer LSTM network. Then, the input data and experimental data are imported into the two models for training and testing respectively, and their performance is compared to obtain more in-depth results.

Fig. 5 Double-layer LSTM network model prediction result (Photo/Picture credit: Original).

Table 3. Double-layer LSTM network model prediction result

Performance	RMSE	MAE	Train Time(second)
value	5.35	4.09	115.7

According to the results of the second experiment, it can be seen that the model using the two-layer LSTM network has achieved significant improvement in prediction performance. As shown in Table 3, compared with the single-layer LSTM network, its values in RMSE and MAE are reduced by 0.78 and 0.45 respectively. Scores are optimized by 13% and 10%. This shows that by introducing appropriate layers of neural networks, the network model can more effectively capture the complex relationships in the data, thus improving its prediction accuracy [12]. It is evident that having the right amount of network layers facilitates feature extraction and raises the model's accuracy. In this experiment, the two-layer neural network was expanded with an additional layer of the same neural network, but the prediction effect was the same as that of the two layers. Furthermore, the training time was 180 seconds. This demonstrates that while adding layers to a neural network won't always result in better prediction performance, it will result in significantly more computation. This demonstrates that while adding layers to a neural network won't always result in better prediction performance, it probably will result in significantly more computation. The study also found that for time series data such as stocks with a suitable network layer structure, continuously increasing the number of network layers does not have a significant effect on improving prediction accuracy. Since there are slight fluctuations in data prediction, the neural network structure needs to be further adjusted to reduce the jitter of the model. When comparing the computational effort with prediction performance, it is found that better performance can be achieved without additional network layers. The established two-layer LSTM network model performs pretty well in stock price prediction, as demonstrated by the findings, which also exhibit low computational cost and good prediction accuracy.

5. Conclusion

Predicting stock prices is a significant and difficult undertaking that requires intricate time series analysis and the effect of several factors. This paper explores the appropriate number of network layers and number of neurons in the LSTM network model to analyze and predict the stock price of BYD Co., Ltd., and obtains good prediction results. However, it is difficult for existing models to cover all influencing factors and fully reflect their impact on stock prices. To improve prediction technology, a detailed examination of the stock market's regulations is needed. This will improve the model's accuracy and usefulness by allowing it to more closely resemble real-world situations. To improve the model's generalization ability and prediction accuracy, bring it closer to reality, increase the model's applicability to technology, and improve forecast accuracy, future research will try to incorporate additional functions, such as trading volume, news index, and market index, among others. Provide a more valuable reference for investors' investment decisions.

References

[1] Liu J, Lin C M M, Chao F. Gradient boost with convolution neural network for stock forecast. Advances in Computational Intelligence Systems: Contributions Presented at the 19th UK Workshop on Computational Intelligence, September 4-6, 2019, Portsmouth, UK 19. Springer International Publishing, 2020: 155-165.

[2] Yuhan W, Yameng Z, Guoliang W. Stock Price prediction based on BP neural network and high order fuzzy cognitive graph. Intelligent Computers and Applications, 2023, 13(08): 100-6+13.

[3] Ariyo A A, Adewumi A O, Ayo C K. Stock price prediction using the ARIMA model. 2014 UKSim-AMSS 16th international conference on computer modelling and simulation. IEEE, 2014: 106-112.

[4] Jinchen H, Shunyu L, Nan S, et al. Application of Deep Learning in Stock forecasting. Information Technology and Informatization, 2023, (09): 190-3.

[5] Rather A M, Agarwal A, Sastry V N. Recurrent neural network and a hybrid model for prediction of stock returns. Expert Systems with Applications, 2015, 42(6): 3234-3241.

[6] Peng Y, Yuhong L, Rongfen Z. Modeling and Analysis of Stock price prediction based on LSTM. Computer Engineering and Applications, 2019, 55(11): 209-12.

[7] Yuping H, Weixuan L, Zuhuan X. Comparative analysis of deep learning frameworks based on TensorFlow and PyTorch. Modern Information Technology, 2020, 4(04): 80-2+7.

[8] Hongrui Z, Lei X. Research on stock prediction based on LSTM-CNN-CBAM model. Computer Engineering and Applications, 2021, 57(03): 203-7.

[9] Haowen W, Juan L, Ya G. Stock prediction model based on feature optimization and loss function improvement. Information Systems Engineering, 2023, (05): 125-8.

[10] Lu M, Xu X. TRNN: An efficient time-series recurrent neural network for stock price prediction. Information Sciences, 2024, 657: 119951.

[11] Mehmood F, Ahmad S, Whangbo T K. An efficient optimization technique for training deep neural networks. Mathematics, 2023, 11(6): 1360.

[12] Adebiyi A A, Adewumi A O, Ayo C K. Comparison of ARIMA and artificial neural networks models for stock price prediction. Journal of Applied Mathematics, 2014.