Proceedings of the 3rd International Conference on Financial Technology and Business Analysis
DOI: 10.54254/2754-1169/152/2024.19430

Analysis of Stock Prediction M+odel Based on LSTM

Yuyang Zhou[1,a,*]

[1]Sichuan Agricultural University, College of the Environment Science, Chengdu, Sichuan Province, 611134, China
a. 18010680061@163.com
**corresponding author*

Abstract: As the complexity and volatility of the stock market continue to increase, investors are placing greater emphasis on the need for accurate stock price predictions. Traditional statistical models often face limitations in time series forecasting, particularly when it comes to capturing the intricate and dynamic changes in stock prices. This paper explores the application of the LSTM model for stock price prediction by developing a forecasting model based on LSTM. First, relevant stock data is collected through a web crawler, and then an LSTM model is trained using this data to predict the price of a specific stock. To assess the performance of the model, evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), and the coefficient of determination (R^2) are employed. The results demonstrate that the LSTM neural network model effectively predicts nonlinear stock trends.

Keywords: long short-term memory neural network, stock price prediction, quantitative investment, data analysis.

1. Introduction

The stock market plays a pivotal role in our nation's market economy, serving as an indicator of the current stage of socio-economic development. The forecasting of stock and index prices has long been a focal point within the financial sector. Before the advent of deep learning techniques for quantitative investments, traditional models such as autoregressive (AR), moving average (MA), and autoregressive integrated moving average (ARIMA) were predominantly utilized. Wu Yuxia and Wen Xin successfully employed the ARIMA model to predict short-term stock prices [1]. While these conventional statistical models have demonstrated some efficacy in time series data prediction, they also exhibit notable limitations when confronted with complex nonlinear factors. With advancements in artificial intelligence, machine learning algorithms have gained widespread application across various fields due to their robust feature extraction and learning capabilities [2]. Deng et al. utilized a backpropagation (BP) neural network for stock price prediction [3]. Despite the significant advantages that machine learning algorithms offer in managing nonlinear data, general machine learning approaches still encounter challenges related to feature extraction owing to the time-series correlations inherent within stock data.

With the advent of big data analysis, deep learning technologies have demonstrated considerable potential to surpass traditional machine learning methods in data prediction, owing to their remarkable learning and generalization capabilities. The recurrent neural network (RNN) is particularly effective for processing time series data as it can capture and retain prior information within a sequence. However, RNNs face challenges related to vanishing gradients when handling long sequential inputs. To mitigate this issue, the long short-term memory (LSTM) network—a variant of RNN—efficiently manages and updates information flow through the introduction of a gating mechanism, thereby enhancing its ability to address long-term dependencies. In recent years, LSTM models, along with their hybrid and modified versions, have been extensively applied across various domains to tackle classification and prediction tasks. This study integrates quantitative investment strategies with deep learning techniques and text analysis to forecast Apple's stock price by leveraging the LSTM model's capacity for retaining both short- and long-term memory. The objective of this research is to facilitate quantitative analyses of future trends in stock prices through nonlinear target forecasting

2. Basic Theory

2.1. LSTM model

Long Short-Term Memory (LSTM) networks, a specialized variant of recurrent neural networks (RNNs), were introduced by Hochreiter and Schmidhuber in 1997 [4]. LSTM is a deep learning architecture particularly effective in handling and predicting time series data. The model incorporates gating mechanisms—specifically the input gate, forget gate, and output gate—which allow it to selectively retain relevant information while discarding non-essential details. These gates facilitate the capture of long-term dependencies within the input sequence, thereby enhancing the model's ability to learn and maintain temporal relationships over extended periods [5-8].

The unit model structure and forward propagation formula of LSTM are shown in Figure 1.

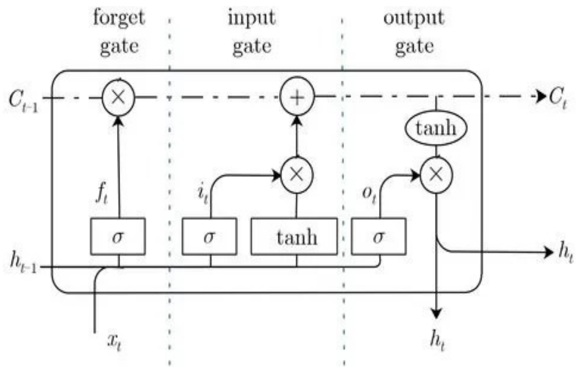

Figure 1: LSTM Model Structure

$$f_t = \sigma(w_{fh} \cdot h_{t-1} + w_{fx} \cdot x_t + b_f)$$
$$i_t = \sigma(w_{ih} \cdot h_{t-1} + w_{ix} \cdot x_t + b_i)$$
$$\tilde{c}_t = \tanh(w_{ch} \cdot h_{t-1} + w_{cx} \cdot x_t + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$o_t = \sigma(w_{oh} \cdot h_{t-1} + w_{ox} \cdot x_t + b_o)$$
$$h_t = o_t \odot \tanh(c_t)$$

The LSTM model is made up of individual neurons, each of which contains a cell state and three gating mechanisms: the input gate, the forget gate, and the output gate. The input gate controls the integration of new information into the cell state, while the forget gate manages both the weight and the activation function related to the cell state. The output gate regulates what information is output from the network, thereby influencing the hidden state value. A key feature of the LSTM is its cell state, which maintains a stable flow of information through the RNN by allowing data to pass through entire cells with minimal deviation or branching.

2.1.1. Forget gate

$$f_t = \sigma(w_{fh} \cdot h_{t-1} + w_{fx} \cdot x_t + b_f) \quad (1)$$

The internal structure of traditional RNNs exhibits a high level of uniformity across time steps. At each time step, the input x(t) is combined with the previous hidden state h(t-1) to form the vector [x(t), h(t-1)]. This vector is then processed through a fully connected layer, followed by an activation function—usually the sigmoid function—to compute f(t). In this context, f(t) serves as a gate value, which, much like a door controlling access, regulates the flow of information within the network. Moreover, the forget gate value plays an essential role in determining the amount of previous information to discard, which in turn affects the cell state in subsequent layers.

Given that the forgetting gate value is computed based on both x(t) and h(t−1), we can interpret this entire formula as delineating how much historical information retained in the cell state of preceding layers is forgotten based on inputs at both present and past time steps.

2.1.2. Input gate

$$\mathrm{i_t} = \sigma(\mathrm{w_{ih}} \cdot \mathrm{h_{t-1}} + \mathrm{w_{ix}} \cdot \mathrm{x_t} + \mathrm{b_i}) \quad (2)$$

$$\tilde{\mathrm{c}}_\mathrm{t} = \tanh(\mathrm{w_{ch}} \cdot \mathrm{h_{t-1}} + \mathrm{w_{cx}} \cdot \mathrm{x_t} + \mathrm{b_c}) \quad (3)$$

There are two approaches for calculating the input gate. The first approach produces the input gate value, which closely resembles the formulation of the forgetting gate, with the key difference being their intended purposes. This formulation determines the extent to which incoming information should be filtered. The second approach to the input gate follows the standard calculation method used in traditional recurrent neural networks (RNNs). However, in the case of Long Short-Term Memory (LSTM) networks, this formulation influences the current cell state, rather than an inferred state as seen in classical RNNs.

2.1.3. Output gate

$$\mathrm{o_t} = \sigma(\mathrm{w_{oh}} \cdot \mathrm{h_{t-1}} + \mathrm{w_{ox}} \cdot \mathrm{x_t} + \mathrm{b_o}) \quad (4)$$

$$\mathrm{h_t} = \mathrm{o_t} \odot \tanh(\mathrm{c_t}) \quad (5)$$

The output gate is governed by two formulas. The first formula computes the gate value for the output gate, which is derived using the same method as for the forget gate and input gate. The second formula uses this gate value to generate the implied state h(t), which then interacts with the updated cell state C(t). This interaction involves applying the tanh activation function, resulting in h(t) being used as part of the input for the next time step. In essence, the entire process of the output gate focuses on generating the implied state h(t).

DOI: 10.54254/2754-1169/152/2024.19430

2.1.4. Cell state update structure

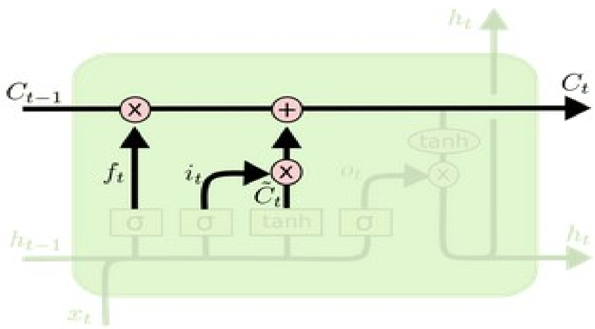

Figure 2: Cell state update structure

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{6}$$

The structure and calculation formula for cell updates, as shown in Figure 2, are straightforward to comprehend. Instead of a fully connected layer, the process involves multiplying the forget gate value, obtained previously, with the cell state C(t-1) from the prior time step. This is then added to the product of the input gate value and the unupdated cell state C(t) from the current time step. The updated cell state, C(t), is subsequently used as part of the input for the next time step. Essentially, the entire process of updating the cell state revolves around the application of the forget gate and the input gate.

By utilizing three gates and a single neuron cell state, the LSTM model effectively leverages data to produce a long-term memory effect on earlier inputs. Subsequently, the parameters are continuously updated through backpropagation, which progressively aligns the model's output with the true values while minimizing function loss [9-10].

3. Empirical study

3.1. Data processing

This paper takes the prediction of Apple's stock price as an example, and selects 14 years' stock price data of Apple from 2010 to 2024 from Yahoo! Financial database. Python was used in the following six steps: (1) Data acquisition from Yahooo Finance (2) data preprocessing: A. Create data set with time step B. Divide data into training set and test set (3) Build LSTM model (4) Train model (5) Make predictions (6) Visualize results Build an LSTM stock price prediction model, and the prediction results are shown in Figure 3.

DOI: 10.54254/2754-1169/152/2024.19430

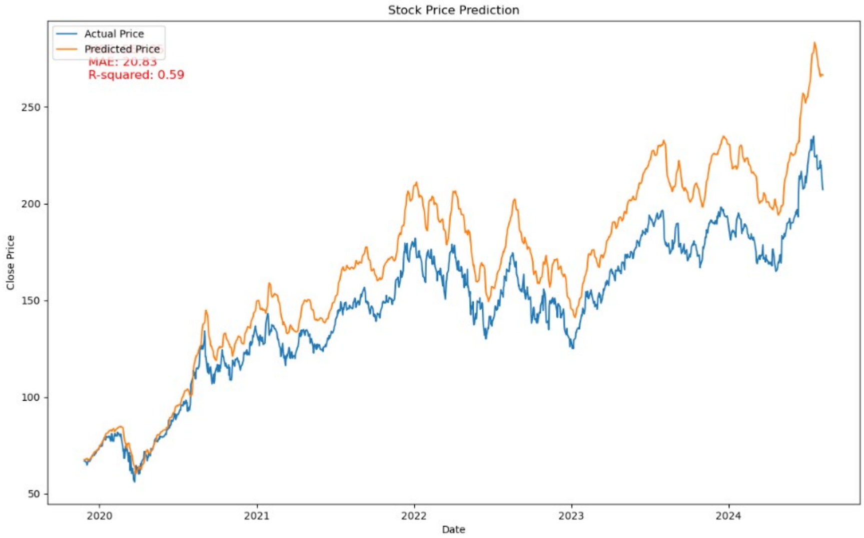

Figure 3. Results of prediction

3.2. Analysis of results

There exists a degree of uncertainty in predicting stock and stock index prices, often resulting in discrepancies between the outcomes generated by stock price prediction models and actual values [11]. To demonstrate the precision of the model's predictions, this paper employs the mean absolute error (MAE), mean square error (MSE) and R-squared to quantify the error of stock price. The closer the r-squared is to 1, the better the correlation. This result indicates a general correlation. And with regard to the stock price, LSTM has a certain memory ability. It has stronger "memory" for the data closer to the current time and weaker "memory" for the data farther away from the current time. Therefore, the model will rely more on the data near the end in predicting the stock price. Specifically, when the stock price experiences a consistent upward trend in the near future, the model tends to predict a continuation of this rise. Conversely, when recent stock price faces successive declines, the model is likely to anticipate further drops. This characteristic may result in significant consequences. The model only adjusts its predictions when there is a noticeable shift in the stock price trend such as a sudden fall following a prolonged increase or a rapid rise after a sustained decrease. Hence, the model will experience substantial "delay." in its responses.

4. Conclusion

This paper presents the development of an LSTM model designed to predict the price of a single stock, with a particular focus on Apple Inc. The findings reveal that both the accuracy and correlation of predictions tend to decline progressively over time. Whether employing LSTM or other machine learning models, these approaches fundamentally encapsulate the operational principles governing stock prices through mathematical logic; however, this is contingent upon the assumption that stock prices exhibit consistent patterns, especially in the long term. Nevertheless, the stock market is subject to influences from policies and news originating from various sources. Any unforeseen developments—whether positive or negative—can disrupt existing trends or patterns associated with a given stock. The model's capacity to accurately reflect the original trend of stock prices is significantly hindered by these news factors, which represent random and unpredictable events acting as noise. Furthermore, many widely-traded stocks experience undue fluctuations due to news or external variables, complicating efforts to capture their inherent regularity. Consequently, future research should integrate additional elements such as news sentiment and policy analysis into a hybrid

DOI: 10.54254/2754-1169/152/2024.19430

model based on LSTM. This approach would enhance both the accuracy and comprehensiveness of stock price predictions, thereby enabling investors to devise more effective investment strategies while mitigating risks.

References

[1] *Wu Yuxia, Wen Xin. Short-term Stock Price Forecasting based on ARIMA Model [J]. Journal of Statistics and Decision Making, 2016,32 (23) : 83-86.*

[2] *He Qing, Li Ning, LUO Wenjuan, et al. A Survey of Machine Learning Algorithms under Big Data [J]. Journal of Pattern Recognition and Artificial Intelligence, 2014,27 (4) : 327-336.*

[3] *Deng Xuan-kun, WAN Liang, HUANG Nana. Research on Stock Prediction Based on DAE-BP Neural Network [J]. Journal of Computer Engineering and Applications, 2019,55 (3) : 126-132.*

[4] *HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780.*

[5] *ADHIKARI R, AGRAWAL R K. A combination of artificial neural network and random walk models for financial time series forecasting [J]. Neural Computing & Applications, 2014, 24(6):1441-1449*

[6] *GAO T, XIU L, CHAI Y, et al. Deep learning with stock indicators and two-dimensional principal component analysis for closing price prediction system [C]//2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS). 2016: 166-169.*

[7] *MOGHAR A, HAMICHE M. Stock market prediction using LSTM recurrent neural network [J]. Procedia Computer Science, 2020, 170: 1168-1173*

[8] *Zhang Y, Li L. RF-MIP-LSTM stock price prediction model [J/OL]. Computer Engineering and Applications,1-14[2024-0813].*

[9] *Yang Qing, Wang Chenwei. Research on Global Stock Index prediction based on Deep learning LSTM neural network [J]. Journal of Statistical Research, 2019, 36(3): 65-67.*

[10] *Peng Yan, LIU Yuhong, ZHANG Rongfen. Modeling and Analysis of Stock Price prediction based on LSTM [J]. Computer Engineering and Applications, 2019, 55(11): 209-212.*

[11] *Zhang Ying, Li Lu. RF-MIP-LSTM Stock Price Prediction Model [J/OL]. Computer Engineering and Applications.*