Int. j. adv. multidisc. res. stud. 2023; 3(3):409-413

Received: 15-04-2023
Accepted: 25-05-2023

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

Stock Market Forecasting Based on Long-Short Term Memory Model

[1] **Nghiem Van Tinh,** [2] **Bui Thi Thi**
[1, 2] Faculty of Electronic, Thai Nguyen University of Technology, Thai Nguyen, Vietnam

Corresponding Author: **Nghiem Van Tinh**

Abstract
Many authors have been utilizing various data models, machine learning, and data mining to anticipate the future movement of stock prices since the stock market's beginnings. This study uses stacked LSTM deep learning to establish a prediction model with view to forecasting Netflix stock values on day-closing. The "Stock ticker" characteristic is used as an input in the forecasting model, which forecasts stock market closing price as a chart using a web application written in Python. Date, Open, Close, High, and Low are the attributes that are included in the model. Data was gathered between the years 2019 and 2022, and I separated it into two parts: a training set and a testing set. Only the testing portion is to be used for the final forecast. The closing time is then displayed against time on a graph. The results suggest that NETFLIX functions effectively.

Keywords: Forecasting, Stock Market, LSTM Model, Forecasting Accuracy

1. Introduction

The financial system of a country depends on the stock market. It is one of the best investment prospects for businesses and investors alike. Stocks are a popular investment choice because they have a higher potential for profit than other types of companies, but there is also a risk that they could lose a lot of money quickly. Therefore, forecasting future stock price patterns is crucial for stock traders' decision-making [1]. One technique for learning to forecast stock price movements using historical data patterns on the stock market is the use of technical elements [2]. To effectively reduce risk, forecasting models based on technical aspects must be rigorous, exhaustive, and precise [3]. There are numerous stock trading prediction models that have been put out. These models often use technical stock trading data aspects, such as high, low, open, close, volume, and changing prices, as their data features. The achievement of the highest and lowest prices in a day are the high and low prices, respectively. The opening and closing prices of the day are, respectively, the open and close prices. Volume measures how often exchanges trade, while change measures how much the price has changed over time [4, 5]. These days, the usage of the graphics processing unit (GPU) that facilitates data learning is one of the computer technologies that are being developed to enable deep learning that is expanding very quickly. Using a GPU instead of a conventional processor will make the data training process much faster [6]. The recurrent neural network (RNN) is one of the deep learning prediction models for time-series data, such as changes in stock prices. When processing input, which is typically sequential data, the RNN algorithm is called repeatedly. This is a sort of neural network design. Because of this, it is excellent at forecasting changes in stock price [7]. Long Short-Term Memory (LSTM) and Gated Recurrent Unit are the two RNN development designs that are most frequently employed. This paper will use the LSTM model to make predictions. TensorFlow3.0, which is mostly used for the development of deep learning algorithms, is the framework that is utilized to predict stock prices using LSTM. The Kaggle platform's historical data on Netflix stock and the Keras deep learning library for stock price prediction served as the foundation for this project. The LSTM model is employed to train the stock data's "close" feature to forecast the closing price. The proposed forecasting model is evaluated using two accuracy measures obtained from the loss function Mean Absolute Percentage Error (MAPE). These experiments demonstrate that the overall forecasted error can be shown to be minor, and the prediction effect can be seen on the graph. However, the price deviation is present even though the forecasted scenario is essentially consistent with the trend of the actual situation. According to the data set, Netflix's stock price is expected to increase on average during the following thirty days.

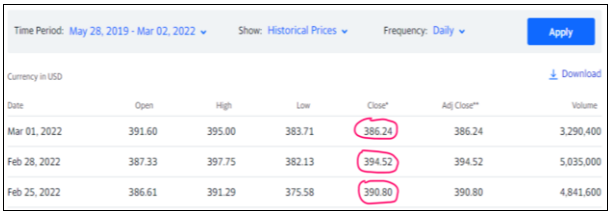
The rest of paper is structured as follows: Section 2 provides the data source and describes the briefly LSTM model. Section 3 proposes the forecasting model based on LSTM model. Implementation of proposed forecasting and experimental results are explained in Section 4. Finally, conclusions are presented in Section 5.

2. Methods and Materials

2.1 Data Source

The data source applied in this experimental result is a collection of historical data on company stock prices attained from the Yahoo Finance website https://finance.yahoo.com/quote/NFLX/history?p=NFLX.

Fig 1 shows an example of stock price time series data from May 2019 to March 2022. It should be noted that the stock market does not trade on any holidays.


Time Period: May 28, 2019 - Mar 02, 2022 Show: Historical Prices Frequency: Daily Apply

Currency in USD Download

Date	Open	High	Low	Close*	Adj Close**	Volume
Mar 01, 2022	391.60	395.00	383.71	386.24	386.24	3,290,400
Feb 28, 2022	387.33	397.75	382.13	394.52	394.52	5,035,000
Feb 25, 2022	386.61	391.29	375.58	390.80	390.80	4,841,600

Fig 1: An example of NFLX stock price time series data for three years

2.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a kind of recurrent neural network (RNN) that enable long-term memory. The RNN learns by re-propagating the gradient when looking for the optimal value. However, the gradient may disappear or diverge if t becomes longer. This happens because ordinary RNNs do not adequately train long-term memory that relies on sequential data. LSTM has been proposed as an algorithm to cope with this issue. RNN has only one activation function in the intermediate layer, whereas LSTM has multiple activation functions with complex advanced operations performed on various gates [8–11]. LSTM has variable C_t for long-term information storage in its cells or blocks. The old information is removed or new information is updated to the C_t to activate the corresponding long-term memory. The arithmetic portion in the intermediate layer of the LSTM is called the cell or block [12]. The structure of the LSTM block and its gates is given in Fig 2. Its gates are briefly described and respectively calculated according to the purpose of their operation.

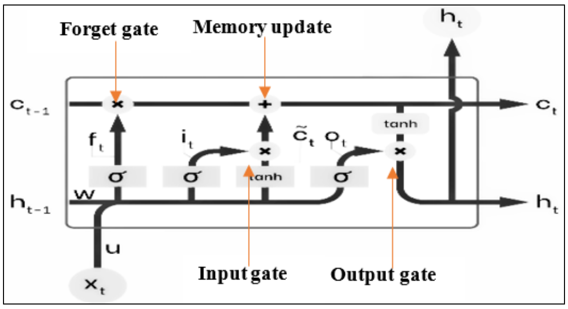


Fig 2: Structure of the LSTM blocks and their gates

Input Layer: This is the first layer in the model, where input values are put into the model.

LSTM Layer: This layer is used to learn and store information about complex relationships between values in time series. The LSTM layer includes gates to control the flow of information into and out of this layer. These gates include: forget gate (forget gate), input gate (input gate) and output gate (output gate).

Output Layer: This is the last layer in the model where the output values are generated.

Fully Connected Layer: This layer connects all neurons from the LSTM layer to the output layer. This layer helps to increase the learning and predictive capabilities of the model.

1. Input Gate. The candidate long-term memory in the current cell state C_t and the storage rate i_t are calculated using Eq. 1 and 2, respectively.

$$\tilde{C}_t = \tanh(u_c * x_t + w_c * h_{t-1} + b_c) \tag{1}$$

$$i_t = \sigma(u_i * x_t + w_i * h_{t-1} + b_i) \tag{2}$$

2. Forgetting Gate. This gate controls to forget information from long-term memory.The storage rate ft is calculated using Eq. 3.

$$f_t = \sigma(u_f * x_t + w_f * h_{t-1} + b_f) \tag{3}$$

3. Output Gate. The output value ot and ht are, respectively, computed using Eqs. 4 and 5

$$O_t = \sigma(u_O * x_t + w_O * h_{t-1} + b_O) \tag{4}$$

$$h_t = O_t \otimes \tanh(C_t) \tag{5}$$

4. Memory Update. The latest long-term memory Ct is updated using Eq. 6

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \tag{6}$$

3. The Forecasting Model Based on LSTM

The objective of this Section is to propose a forecasting model based on the LSTM model to predict stock price trends using Netflix stock prices over the last ten years. The forecasting process is divided into 4 main steps as shown in Fig 3: 1. Pre-process the data. 2. Divide the data into two data sets (training and test). 3. Create a stacked LSTM model 4. Forecast the test data and plot the output.

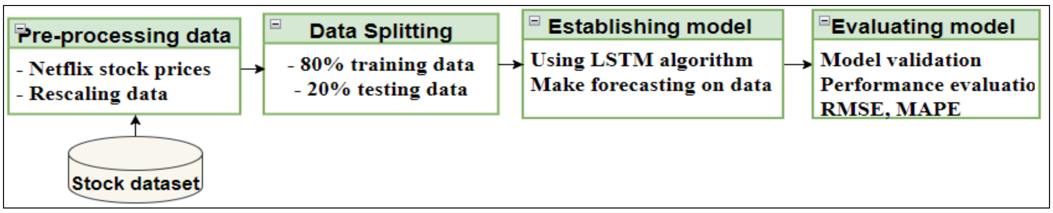


Fig 3: Flowchart of the forecasting model using LSTM

1. Preprocessing or data preparation is a very important stage to make the raw data into quality data that is ready to be processed according to model development needs to model evaluation. It is the initial data processing in advance to be trained in building the model while being validated up to data testing to evaluate the performance of the built model. The following are four sequential steps in the data preprocessing stage. As noticeable, around March 2019, we see a sudden drop in the price, after which it reports steady growth until recently. It will be challenging for a machine learning model to correctly estimate the rapid changes that we can see in March 2019 and February 2022. We will focus on evaluating the model performance in predicting the more recent values after training it on the past data. First, we define the features and the target as discussed above.

```
new_data=pd.DataFrame(data,columns=['Date','Open','High','Low','Close'])
new_data.index=new_data.Date
new_data.drop('Date', axis=1, inplace=True)
```

2. We split our data into training and testing sets. Shuffling is not permitted in time-series datasets. In the beginning, we take two steps worth of past data to predict the current value. Thus, the model will look at yesterday's and today's values to predict today's closing price.

3. Building the LSTM model

We will use the Sequential and LSTM algorithm to build an LSTM model. Given the simplicity of the model and the data, we note that the loss reduction stagnates after only 20 epochs. You can observe this by plotting the training loss against the number of epochs, and LSTM does not learn much after 10-20 epochs. Also, we used different Epochs for training data (12 epochs, 25 epochs, 50 epochs, and 100 epochs) to compare forecasting performance between these epochs.

```
Epoch 1/20: loss improved from inf to 0.07670, saving model to Data\AI_NEW.hdf5
11/11 - 8s - loss: 0.0767 - 8s/epoch - 733ms/step
Epoch 2/20: loss improved from 0.07670 to 0.02324, saving model to Data\AI_NEW.hdf5
11/11 - 2s - loss: 0.0232 - 2s/epoch - 155ms/step
.........................................................................................................................
........................
Epoch 19/20: loss improved from 0.00925 to 0.00903, saving model to Data\AI_NEW.hdf5
11/11 - 2s - loss: 0.0090 - 2s/epoch - 198ms/step
Epoch 20/20: loss improved from 0.00903 to 0.00794, saving model to Data\AI_NEW.hdf5
11/11 - 2s - loss: 0.0079 - 2s/epoch - 175ms/step
dict_keys(['loss'])
```

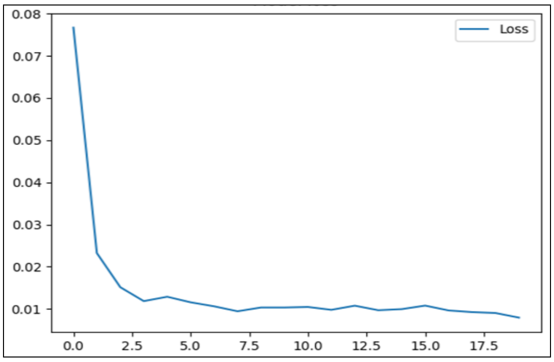


Fig 4: Graphs visualizing the training loss

4. Forecast the test data and plot the output

Performance measurement

In this study, the function used to evaluate the performance of the LSTM model is the loss error function or the difference between the actual and forecasted values. The loss function in this paper uses Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). Eqs.(7) and (8) give the calculation formulas for MAPE and RMSE values, respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(F_i - R_i)^2} \quad (7)$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{F_i - R_i}{R_i}\right| * 100\% \quad (8)$$

Where, R_i denotes actual data at year i, F_i is forecasted value at year i, n is number of the forecasted data, m is order of the fuzzy logical relationships. .

Looking closely at the formula of RMSE, we can see how we will be able to consider the difference between the actual (R_i) and forecasted (F_i) price values for all n time stamps and get an absolute measure of error. The accuracy obtained from MAPE represents the upper limit of the accuracy model, which can be the highest percentage of opportunities achieved using the forecasting model. MAPE is the yield from the normalized absolute distance, producing the closest distance to the actual value.

4. Experiment Results

In this section, the forecasting model based on LSTM is applied to forecast Netflix prices using training forecasts, training observations, testing forecasts, and training observations (Fig 1). Our model uses 80% of the data from 2019-05-28 to 2021-02-23 for training and the other 20% of data for testing. For training, we use the MSE and MAPE to optimize our model. Also, we used different Epochs for training data (20 epochs, 30 epochs, 50 epochs, and 100 epochs) to compare forecasting performance between these epochs. The forecasting results of the LSTM model in the training set are listed in Table 1 and performance evaluation on the testing set is shown in Table 2. The results on these two tables suggest that the model is quite accurate.

Table 1: The result of training for the Netflix stocks with 20 epochs

Date	Actual data	Forecasted value in training
2019-05-28	355.06	355.19
2019-05-29	349.19	354.84
2019-05-30	351.85	354.37
----	----	----
2021-02-22	533.78	544.27
2021-02-23	546.15	542.86
MSE	14.9	
MAPE	0.036	

Table 2: The result of testing for the Netflix stocks with 20 epochs

Date	Actual data	Forecasted value in testing
2021-02-24	553.41	541.55
2021-02-25	546.70	540.72
2021-02-26	538.85	540.01
----	----	----
2022-03-01	386.24	388.37
2022-03-02	380.03	387.57
MAPE	0.013	

The graph below Fig 5 shows the test prediction data in blue, and the red curve represents the overall trend of the Netflix data. Green means that the training has predicted the data. Clearly, there is a high degree of match.

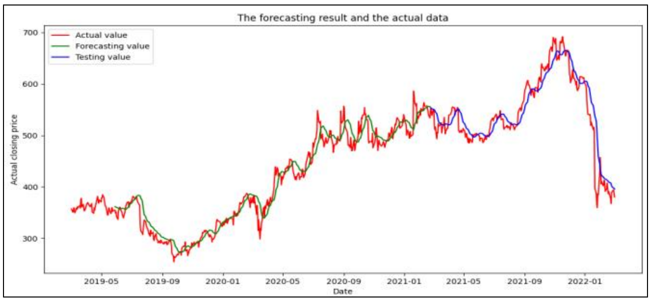
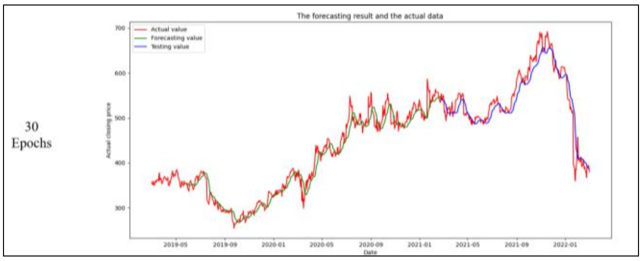
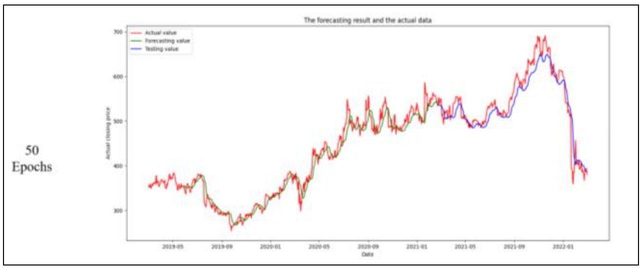


Fig 5: Descriptive curves between actual and predicted values in the training and testing phases





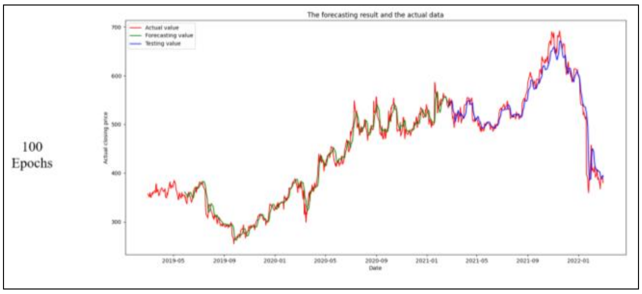


Fig 6: Descriptive curves between actual and predicted values in the training and testing phases with different numbers of epochs

From the forecasting results in Fig 6, we can observe that the same data with the number of more epochs can improve our testing result and at the same time allow us to have better forecasting values. Therefore, the accuracy of the our model is very good.

5. Conclusion

LSTM is better adapted to handle time series-related issues than econometric models like linear regression. In this paper, we have created a prediction model based on the LSTM deep learning model to predict stock prices on Netflix on the day of closing. The prediction algorithm predicts that the stock market will close using the "Stock ticker" feature as input. Using a Python web application, the price is presented as a chart. The attributes in the model are Date, Open, Close, High, Low, Volume, and Adj Close. Between May 2019 and March 2022, data was collected. The results suggest that NETFLIX functions effectively and the closing time is displayed against time on a graph. However, the stock market is affected by various factors such as the economic environment, political policies, and market news. Although the LSTM model is a good fit for the predicted and actual data. But it also cannot predict the stock price accurately. Because the stock market is not very different from the stock price in the recent period. So it is challenging to predict the data for a longer period of time.

6. References

1. Khan W, Ghazanfar MA, Azam MA. *et al*. Stock market prediction using machine learning classifiers and social media, news. J Ambient Intell Human Computing, 2020.
2. Troiano L, Villa EM, Loia V. Replicating a trading strategy by means of LSTM for financial industry applications. IEEE Trans Ind Inf. 2018; 14(7):3226-3234.
3. Suyanto S, Safitri J, Adji AP. Fundamental and technical factors on stock prices in pharmaceutical and cosmetic companies. Finance Account Bus Anal (FABA). 2021; 3(1):67-73.
4. Srivastava PR, Zhang ZJ, Eachempati P. Deep neural network and time series approach for finance systems: Predicting the movement of the Indian stock market. J Organ End User Comput (JOEUC). 2021; 33(5):204-226.

5. Nabipour M, *et al*. Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data: A comparative analysis. IEEE Access. 2020; 8:150199-150212.
6. Budiharto W. Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM). J Big Data. 2021; 8(1):1-9.
7. Zhang Y, Chu G, Shen D. The role of investor attention in predicting stock prices: The long short-term memory networks perspective. Finance Res Lett. 2021; 38:p101484.
8. Le XH, *et al*. Application of long short-term memory (LSTM) neural network for flood forecasting. Water. 2019; 11(7):p1387.
9. Baytas IM, *et al*. Patient subtyping via time-aware LSTM networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017.
10. Goodfellow I, Bengio Y, Courville A. Deep learning. Boston: MIT Press, 2016.
11. Ingle V, Deshmukh S. Ensemble deep learning framework for stock market data prediction (EDLF-DP). In: Paper presented at the Global Transitions Proceedings. 2021; 2(1):47-66. Doi: https://doi.org/10.1016/j.gltp.2021.01.008.
12. Moghar A, Hamiche M. Stock market prediction using LSTM recurrent neural network, In: Paper presented at international workshop on statistical methods and artificial intelligence. 2020; 1701:1168-1173.