"RNN's LSTM Based Deep Learning Model for Stock Price Prediction"

A project paper submitted in partial fulfillment of the requirements for the Degree of M.Sc.(Engg.) in Information and Communication Technology.

Submitted By
Minhajul Islam Mozumder

Roll No: 22209039

Registration No: 22209039

Session: 2021-2022

Department of Information and Communication Technology

Comilla University, Bangladesh

Declaration

I, student of department of ICT, Roll No: 22209039, Reg No:Sc- ICT- 22209039, Department of Information and Communication Technology, Comilla University would like to solemnly declare here that the presented report of my project namely, “RNN's LSTM Based Deep Learning Model for Stock Price Prediction” is uniquely prepared by me under the supervision of Supervisor, Department of ICT, Comilla University. While preparing this report I did not breach my copyright internationally.

ABSTRACT

One of the most significant activities in the world of finance is stock trading. The goal of stock market prediction is to forecast the future value of stocks as well as other financial instruments traded on stock exchanges. Most stock brokers base their stock predictions on technical, fundamental, or time series analysis. Python is the computer language used for machine learning techniques for stock market forecasts. In this research, we propose a Deep Learning approach that can be taught using publicly available stock data to learn from experience and then apply that information to make accurate predictions. A crucial area of research is stock price forecasting because of how profitable it can be for people, businesses, and governments. The method and novel applications are examined in this paper. This study examines novel applications and techniques for forecasting a particular corporation's consistent closing price. Customers can purchase or sell assets of the companies that are regularly scheduled there. A difficult task is predicting changes in stock market prices. In this study, stock price predictions are made utilizing deep learning models like the Long Short-Term Memory (LSTM), a development of the recurrent neural network. For the purpose of prediction, the two-year datasets from 2021/02/23 to 2022/07/20 are used. This study examines novel applications and techniques for forecasting a particular corporation's consistent closing price. Customers can purchase or sell assets of the companies that are regularly scheduled there. A difficult task is predicting changes in stock market prices. In this study, stock price predictions are made utilizing deep learning models like the Long Short-Term Memory (LSTM), a development of the recurrent neural network. For prediction, the two-year datasets from 2021/02/23 to 2022/07/20 are used.

Keywords: Stock Price Prediction, Deep learning, LSTM, Recurrent Neural Network, time series analysis.

List of Contents

Contents **Page**

List of Figure

Chapter 1

Introduction

1.1 Introduction

A stock exchange, stock market, or exchange-traded is a group of producers and consumers who represent claims to ownership in a company; these securities may be registered on public stock exchanges or traded solely privately. The stock price of a corporation has traditionally been predicted in two ways. The stock's past price, close and open stock prices, trading volume, adjacent close value, etc. are all used in the technical analysis method to predict the stock's future price. The second sort of analysis is qualitative, and it includes information on the company, the market, textual data on political and economic variables, financial news items, social platforms, and even blogs by economic specialists. The finance model and figure statistics are widely used in modern society by investment firms and the majority of people. The stock market moves in a random manner, so the best indicator of the value of tomorrow is the value of today. It is undeniably tough to estimate stock indices due to market turbulence, which necessitates accurate forecasting methods. In the preceding information sorting, there is a wide variety of knowledge available. Utilizing previous stock price information and business intelligence performance data study and examination of investment models. Future stock prices, price volatility, and market trends are all possible targets for prospective stock market forecasting. In share market forecasting systems, there are two sorts of predictions: dummy predictions and real-time predictions. By defining some guidelines and listing the average price, participants in dummy forecasting make predictions about the price of shares in the future. Real-time forecasters obsessively browse the web and check the share prices of companies. Investors use information analysis to make educated judgments. To forecast future stock market movements, investors typically read news articles and research corporate histories, market trends, and millions of other data points. Many businessmen have been inspired to use the machine learning approach in this area in current times as a result of machine learning's growing significance across a variety of industries, and almost all of them have shown pretty encouraging outcomes. Large and non-linear data sets are typical for analyses of the stock market. Therefore, in order to deal with this, we need various data-efficient models that can find hidden patterns and intricate connections in all massive data sets used for prediction. Consequently, the main goal of this study is to develop a deep learning model which can be used to forecast and evaluate stock market patterns relating to the company's shares, profit, and loss. In order to circumvent the limitations of conventional recurrent neural networks to learn long-term dependence, hochreiter and schmidhuber developed the long-term memory (LSTM) architecture. A unique class of recurrent neural networks called long short-term memory (LSTM) networks are effective enough to learn the importance of order in situations involving sequence prediction. In fields relating to complicated problem domains, such as language processing, speech recognition, and other areas, these kinds of networks were frequently employed. Five crucial components make up an LSTM cell, sometimes referred to as the memory portion of an LSTM unit, which enables it to represent simultaneously long-term and short data. Cell conditions, hidden states, input gates, forget gates, and output gates are some of the components. The controllers that direct the data stream within the LSTM unit are called the LSTM gates.

1.2 Motivations

When compared to earlier strategies, machine learning algorithms have been shown to maximize productivity in this sector by 60–86%. The majority of earlier research in this area relied on traditional algorithms like linear regression, random walk theory, (RWT), and Moving Average Convergence/Divergence (MACD). Some linear models, such as Autoregressive Moving Average (ARMA) as well as Autoregressive Integrated Moving Average (ARIMA), are also used for stock forecasting. Currently, it has been established that machine learning can enhance stock market predictions. Through a process of self-learning, ANN can uncover hidden traits. It is a great estimator and has the ability to identify correlations between input and output in very large complex datasets. As a result, ANN turned out to be the greatest choice for an organization's stock price predictions.

1.2 Objective

Our study's goals are to identify the factors that affect the financial markets and to extract patterns from a massive amount of data. Utilizing LSTM (long short-term memory) techniques to offer users analysis, share price estimation, and stock market forecasting. In this project, a model for the stock market prediction that provides high accuracy is proposed. We made an effort to forecast stock prices using historical datasets from a model with the best possible accuracy.

1.3 Related Work

The Support Vector Machine model, researcher - developed by Reddy et al. [1], was based on machine learning and utilized a substantial collection of data that was gathered from several international financial markets. A review of techniques for predicting the overall accuracy of share price variations was conducted by Shreenidhi Sriram, Sanjana Rangarajan, and colleagues[2]. They concentrate on predicting stock prices using machine learning algorithms based on logistic regression, which has been shown to be the best for predicting the market value of a share based on historical data. Researcher Mankar et al.[3] attempted to create a system that uses sentiment analysis theories to forecast changes in the stock prices of various companies. The sentiment analysis model was based on tweets collected via the Twitter API, as well as the discretionary consideration of the closing prices of a number of equities. The most likely method for estimating the movement of stock prices in support of tweet sentiment was determined to be the SVM model. Researchers Vijayarani et al. [4] established a framework for predicting the object variable, or price for a specific day, utilizing the random forest algorithm and the stock market. The random forest algorithm, which depends on multiple data points primarily from historical data, is the most logical one according to their measurements of the accuracy of various algorithms. For the provided dataset, LSTM was deemed to be the best prediction algorithm. The generated version of LSTM outperformed both the basic and upgraded versions, according to researchers Mandi & Skanda [8]. They used a benchmark linear regression model to compare the two. Stock market examination by scholars Vijh et al.[9] compared the performance of ANN and RF models with the goal of reducing forecast inaccuracy. For more accurate stock prediction, Mokalled et al.[10] investigated the use of AI approaches utilizing market and broadcast data and proposed an electronic trading system that incorporates mathematical operations, ML, and other external elements. In Selvin et al.'s[11] analysis, latent dynamics are found in the data rather than the data being fitted to a specific model. There has been used of RNN, LSTM, and CNN. The purpose of using three different models was to identify any long-term reliance in the provided data that may be finally discovered from the effectiveness of these models. On a short-term basis, future values were predicted using the template matching method with data overlap. CNN demonstrates that it only provides outcomes that are more accurate than those produced by the other two approaches and that it uses the existing window for prediction. The Covid-19 pandemic caused a large increase in stock price volatility, necessitating careful price forecasting.

Chapter 2

Methodology

2.1 Deep Learning

Deep learning is a type of machine learning artificial intelligence that comprises techniques that let software train multi-layer neural nets exposed to a lot of data to accomplish tasks like speech and picture recognition. Deep learning is used to retrieve high-level attributes from raw input. Deep learning networks are made up of many layers of interconnected nodes, each of which improves on the one before it to improve categorization or prediction. The network's forward computation is sometimes referred to as forwarding propagation. The visible layers, or input and output layers, of a neural network. The input layer is where the deep learning model receives the data for processing and the output layer is where the final classification or prediction is made. Here, "deep" mentions how many hidden layers a neural network has. Deep networks can have around 150 hidden layers while a neural network can only have 2 or 3 hidden layers. A large number of labeled data sets are needed to train a deep learning model and a neural network that can learn and extract features right away from the data.

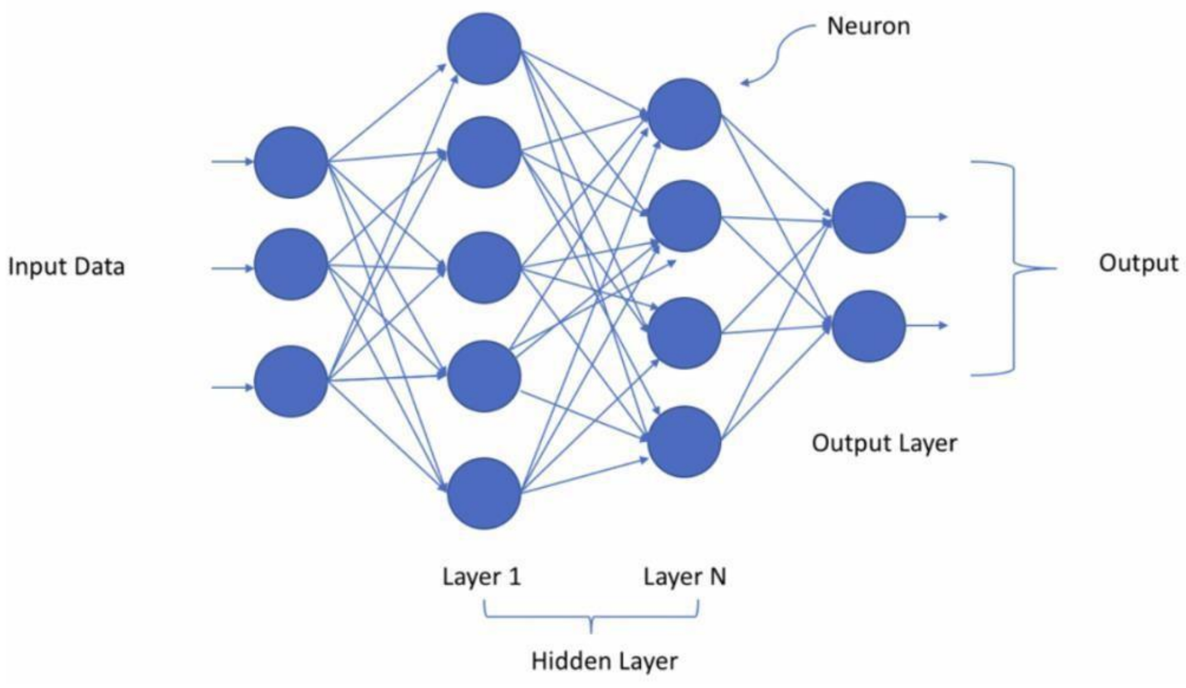


Figure 1: Deep Neural Network

2.2 Recurrent Neural Network

Recurrent networks differ from conventional feed-forward networks in that neurons can transfer data to either the previous or the same layer since they do not just have one-way neural connections. Recurrent neural networks are referred to as short-term memory neural networks since data doesn't really flow unidirectionally in such networks. Feed-forward networks cannot process arbitrary input sequences, whereas RNNs can utilize internal memory. Real-valued activations and various variable weights make up each RNN computing unit. Networks called LSTMs contain loops between them which enable data transmission. The same weights used to build graphs are used to build RNNs by applying them in loops to structures. RNNs are networks with loops that let information travel through them. In RNN, a piece of data is subjected to a number of periodic procedures. In order to forecast prospective events for the future, samples and sequences are discovered, revealed, and then understood. The output unit's RNN is dependent on both its current input and a previously hidden location that stores historical data. The ongoing update of the hidden units in the feedback loop causes the dissolving gradient problem in RNNs. An alternative to Recurrent neural network LSTM (long short-term memory) was introduced to get over this limit of vanishing gradient.

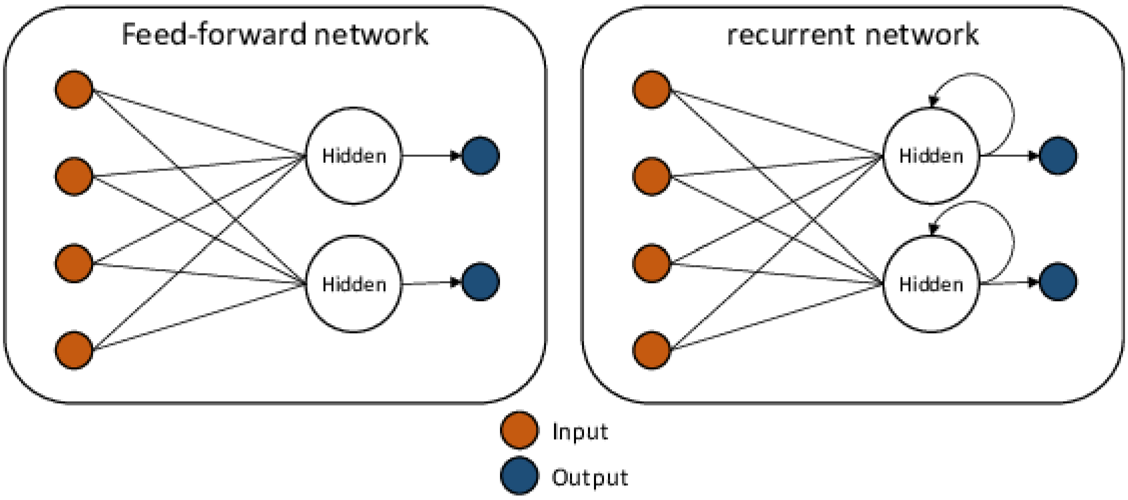


Figure 2: Feed-forward and Recurrent network

2.3 LSTM (Long-Short Term Memory) Model

Long short-term memory is employed in the artificially rnn (recurrent neural network)architecture of deep learning. It is a kind of rnn(recurrent neural network) that has the ability to recognize sequence correlations in sequence prediction issues. In areas where complicated problems must be solved, like speech recognition and machine translation, this behavior is

essential. LSTM is not a straightforward area of deep learning. Unlike traditional feed-forward neural networks, LSTMs have connections for feedback. It has the ability to analyze whole data sequences as well as single data points (like images), but also entire sequences of data (such as speech or video input). The LSTM model has the capacity to hold data over time. They possess a memory that can store n words in order. When working on time-series or sequence data, this capability is extremely helpful. We have complete control over which data we save and which we reject when employing an LSTM model. That is what we do. Utilizing the gate Recent developments in information science have revealed that long short-term memory networks get the ability to selectively remember patterns for extended periods of time for practically all sequence prediction issues. Small adjustments are made by LSTMs as they go through the cell state process. So, LSTM has the ability to selectively recall or forget things. There are three different ways that the information is dependent on a specific cell state.

What determines today's share price is:

1. The tendency the stock has shown over the past days, either an uptrend or a downturn.

2. The stock price from the previous day is important since many traders check it before purchasing a stock.

3. Influencing factors for today's stock price. It can be a newly implemented corporate rule that is drawing heavy criticism, a decline in profits, or a swift shift in top management.

Any issue can be applied to this reliability:

1. Prior state of the cell (i.e. data stored in memory based on the earlier time step.)

2. Previously concealed (i.e. identical to the output of the previous cell)

3. Presently input time step (i.e. at that time, fresh information was provided)

2.4 Architecture of LSTMs

A cell, an input gate, an output gate, and a forget gate make up a standard LSTM unit.

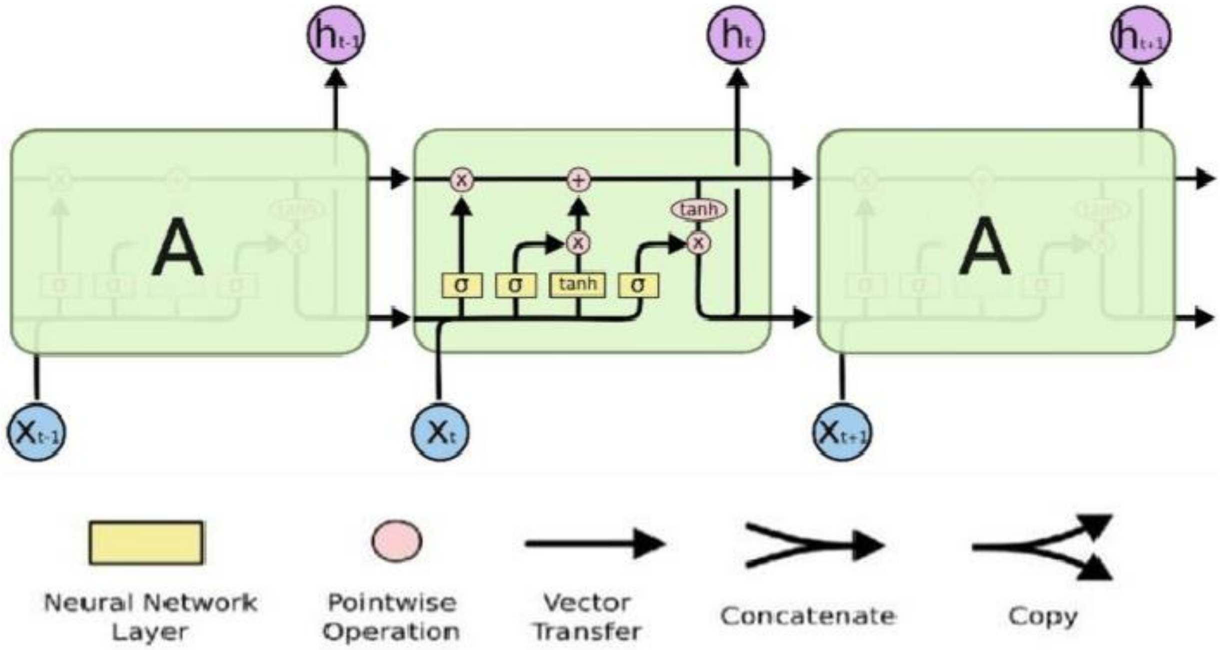


Figure 3: An LSTM's repeating module has four interconnected layers.

Cell state: The crucial component of LSTM is indeed the cell state, shown by the horizontal line rising to the top of the image. Similar to a conveyor belt, it. With just a few modest linear interactions, it completes the full chain. It is quite easy for data to move unaltered. Its cell state, which itself is strictly regulated by structures known as gates, can be added to or removed by LSTMs.

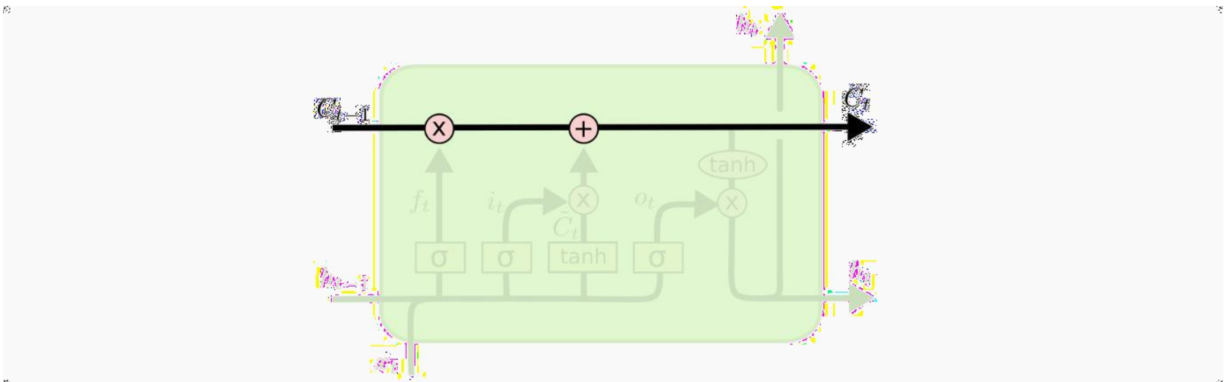
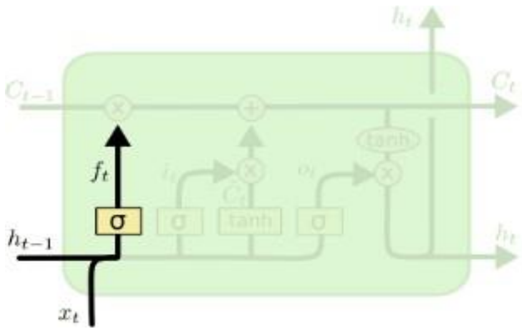


Figure 4: Cell state of LSTM model

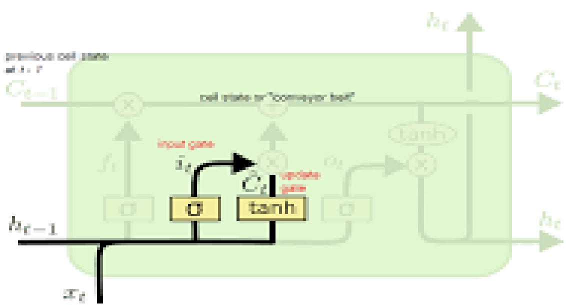
Forget Gate: The LSTM unit first decides whether the structure should be relocated from its previous cell state in this phase (t-1). This "missing gate layer" gate in the sigmoid layer makes this choice. A score among 0 and 1 is output both for values on cell state Ct-1 after looking at ht1 and Xt. A 0 means "totally get rid of it," whereas a 1 means "absolutely keep it."



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

Figure 5: Forget gate of LSTM Model

Input Gate: The following step is to decide which new data we will store inside the cell state. There are two parts to it. First, the values which will be updated next are chosen by a sigmoid layer called the "input gate layer." A recent candidate value Ct vector created by a tanh h layer can be added to this layer. We will connect these two in the next stage to enhance the situation. It is now time to change Ct-1, its old cell state, to Ct, the new cell state. The previous steps have already decided what to do; all that is left is for us to carry it out. We decide to match and forget the previous decision we made to forget the old stage multiplication. Then, we include I(t)*Ct. These are the updated figures, scaled in accordance with how much we decided to raise each state's value.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 6: Input gate of LSTM model

Output Gate: A filtered form of the output will rely on the status of our cells. The first step is to perform a sigmoid layer to choose which portion of the hidden state will be output. After that, we maintain the cell state using tanh to obtain values within -1 and 1, multiply that value by the sigmoid gate's output, and output only the portions that we have chosen.

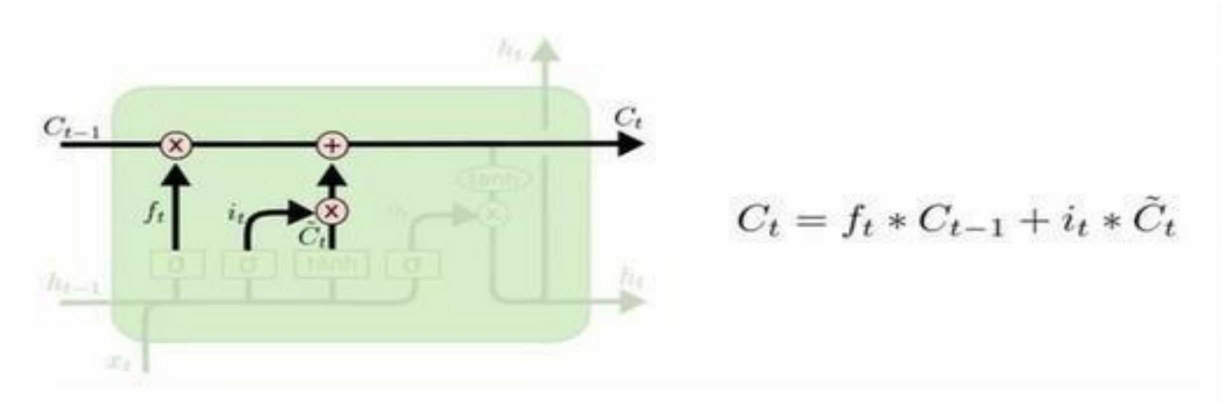


Figure 7: LSTM model output gate

Data gathering, data pre-processing, and lastly models, inputs, and pre-processing make upthe various components that make up system design. Data from the stock market, including the open, high, low, and close values. It is known as OHLC data. The stock's OHLC data is accessible on numerous exchange websites. Downloads of this information are typically free.

2.5 Download & import the financial Data

The main focus of any prediction methodology is data. So the fundamental component in this case is financial data. I used data from the Tesla stock market for 8 years, from 2015-06-29 to 2022-07-20. I have data that I have gathered from the Yahoo Finance website, which includes high, low, open, close, volume, and adjClose.

Table 1. Historical data of TSLA price

Date	High	Low	Open	Close	Volume	Adj Close
2015-06-29	53.189999	52.139999	52.389999	52.403999	17394500.0	52.403999
2015-06-30	54.183998	52.799999	52.959999	53.652000	15434500.0	53.652000
2015-07-01	54.523998	53.570000	54.222000	53.830002	10506000.0	53.830002
2015-07-02	56.490002	54.661999	56.040001	56.004002	35819500.0	56.004002
2015-07-06	56.338001	55.259998	55.776001	55.944000	20609500.0	55.944000

Date	High	Low	Open	Close	Volume	Adj Close
...	...	...	...	...	...	...
2022-07-14	715.960022	688.000000	704.690002	714.940002	26185800.0	714.940002
2022-07-15	730.869995	710.669983	720.000000	720.200012	23165500.0	720.200012
2022-07-18	751.549988	718.809998	734.809998	721.640015	27512500.0	721.640015
2022-07-19	741.419983	710.929993	735.000000	736.590027	26885000.0	736.590027
2022-07-20	751.952820	739.510010	740.349976	745.750000	3595868.0	745.750000

1778 rows × 6 columns

Figure 8: Plotting of TESLA historical prices.

Data on stocks has been gathered here. The comma-separated numbers file should then be initialized in the programming environment. Later, we look for column names. All accessible information includes data, open price, high price, low price, closing price, volume, and adjusted close price. We have shown a plotting graph of the share price in the above figure. where open, high, low, and close prices are the different stock price formats.

Figure 9: Ploting of Open and Close price History

From the above figure, it can be said that the open and close prices are very close values.

2.6 Splits the dataset into train set and validation set

At this stage, the datasets have been separated into train sets and test sets. 80% [29/06/2015 - 22/02/2021] of data have been held for the training set and the remaining 20% [23/02/2021 - 20/07/2022] for the validation set. For training intentions, the train and test sets have been further divided into Xtrian, Ytrain, Xtest and Ytest.

Pseudo code for train data splitting:

```
for i in range(60,
len(train_data)):
x_train.append(train_data[i-60:I, :])
y_train.append(train_data[i, :])
```

Pseudo code for train data splitting:

```
for i in range(60,len(test_data)):
x_test.append(test_data[i-60:i, :])
y_test.append(test_data[i, :])
```

2.7 Data Pre-Processing

Data is not always accurate in the actual world. Consistencies, noise, insufficient data, and missing values all contribute to its corruption. The further data we collect, the better ones we can train, which is a general rule in machine learning. Pre-processing data is a crucial stage in machine learning which helps to enhance data quality and enable the extraction of insightful knowledge from data. Preparing real data (cleaning and arranging it) in order to make it suitable for creating and refining machine learning models is known as data preparation.

Feature scaling:

The most crucial step in machine learning's data preprocessing is feature scaling. The distinguishing parameter of a dataset can be standardized within a specific range using feature scaling. Feature scaling, in other words, specifies the range of factors so how you can compare them on an equal footing.

There are two techniques to achieve feature scaling in machine learning.

Normalization: Values are adjusted and rescaled throughout the normalization process such that they finish up within 0 and 1. Additionally called Min-Max scaling.

The normalizing formula is as follows:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Here, Xmax and Xmin are the feature's maximum and minimum values, respectively.

- Whenever the parameter is the lowest number in the column, the denominator will be 0, and X' will also be 0.
- But from the other hand, the denominator will equal the denominator if indeed the data value inside the columns is the max value, making X' equal to 1.
- The relationship between X' is around 0 and 1 if the value of X would be between the lowest and maximum values.

Standardization: A different method of scaling places values with such a single standard deviation in the middle of the mean. This indicates that the attribute's mean approaches zero and that the formation, as a result, does have an units standard deviation.

The method for standardizing is as follows:

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation } (x)}$$

Really understanding the size of the data is LSTM. We always should scale or normalize the dataset before entering it because doing so is a recommended practice. The neural networks are offered. The data value respectively 0 and 1 was transformed using the MinMaxScaler algorithm. I'm going to assume that you'll resize the data within [-1,1]. That value may also be provided. To do this, MinMaxScaler, a fantastic preprocessing package from Sklearn, can be used.

2.8 Develop LSTM model

This is the principal step of this paper. Here, the model we consider is multi-layer and multivariate.

Pseudo code of the model is:

```
model=Sequential()

model.add(LSTM(60,return_sequences=True,input_shap=(x_train.
shape[1],1)))

model.add(LSTM(30,return_sequences=False))

model.add(Dense(25))

model.add(Dense(1)
```

The summary about the built LSTM model:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 60, 60)            14880

 lstm_3 (LSTM)               (None, 30)                10920

 dense_2 (Dense)             (None, 25)                775

 dense_3 (Dense)             (None, 1)                 26

=================================================================
Total params: 26,601
Trainable params: 26,601
Non-trainable params: 0
```

Chapter 3

Result&
Discussion

Result

3.1 Check model fitting by training loss potting

When the model is finished training, the loss per epoch that the model save after each epoch

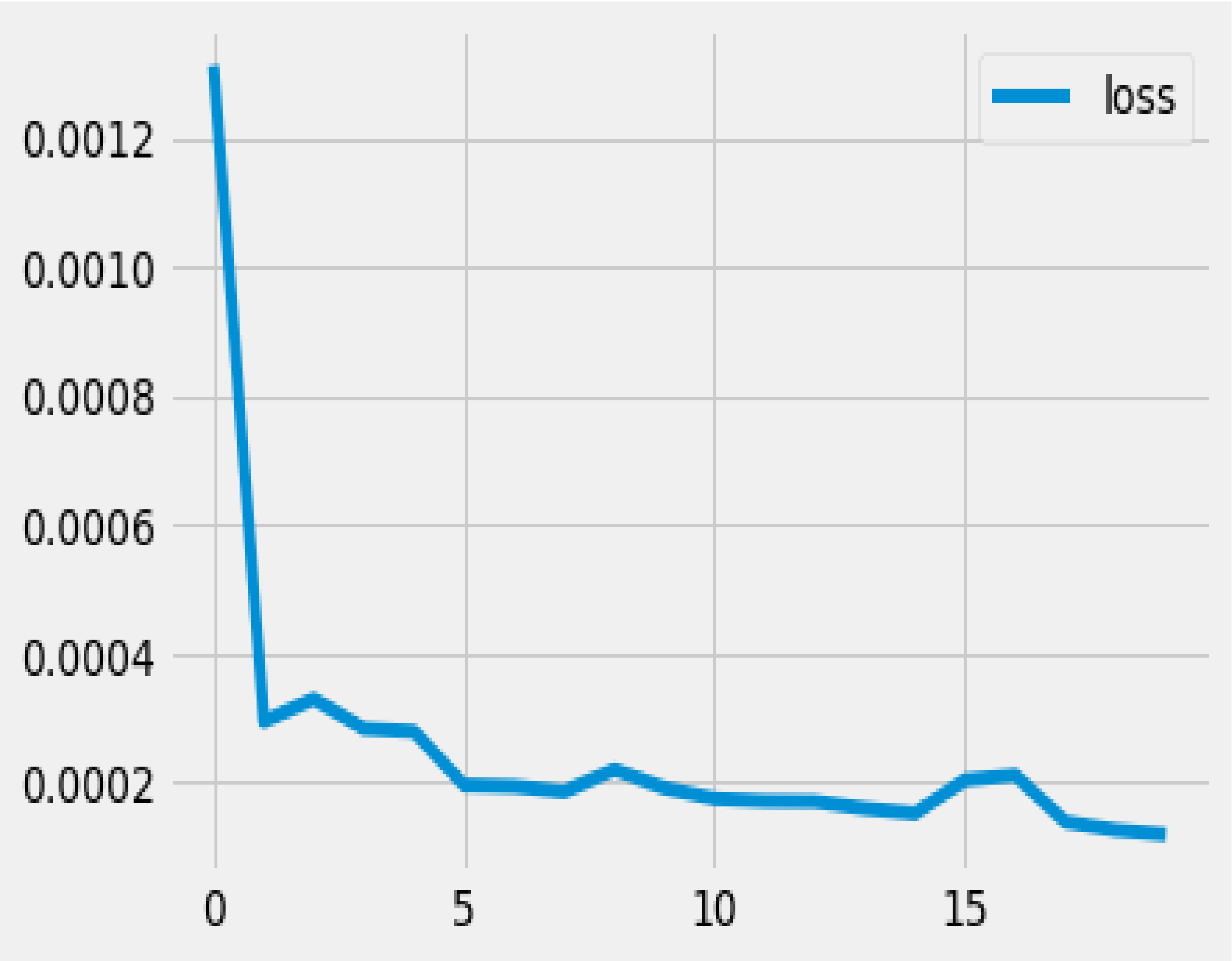


Figure 10: Loss per-Epoch

With each epoch, the loss has decreased, as can be shown. After completing all of the epochs, the loss was reduced to less than 0.0002, starting with such a loss of 0.0012.

3.2 Calculate the various error factors

At this point, different error factors are estimated to assess the model's effectiveness. Included in the consideration are the Mean Absolute Error (MAE), Root Mean Absolute Error (RMAE), Percentage Mean Absolute Error (PMAE), and R2-Score.

Mean-absolute-error: 34.22188651931118

Root-mean-squared-error: 46.307455094323124

Mean-absolute-percentage-error: 0.040663352167813194

R2_score: 0.9214375754338462

However, our main anxiety is to choose a feature selection algorithm in the LSTM model that provides maximum forecasting accuracy. In this case, a relative analysis of valid test data and stock market forecasted data is given:

Figure 11: Comparison of predict data with actual test data for TSLA

While the red color depicts the actual price of the testing data, the blue color displays the close price of the training data. Finally, the predictions based on the testing results that appear to be very correct are shown by the green hue.

3.3 Discussion

To verify model accuracy. The model is again tested with Apple, Microsoft, and Google stock markets from the Yahoo Finance website. For every share, the accuracy model provides the

best algorithm and the highest prediction accuracy of that algorithm. Figure 4.3, 4.4, 4.5 shows their best predicted accuracy value.

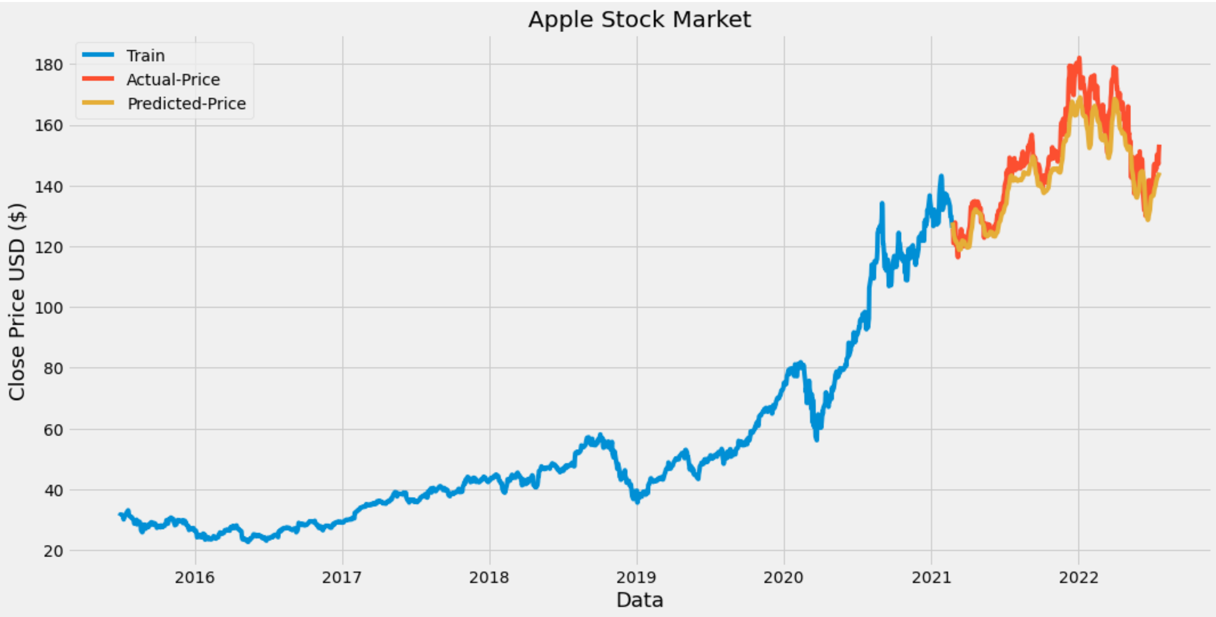


Figure 12: Comparison of predict data with actual test data for Apple

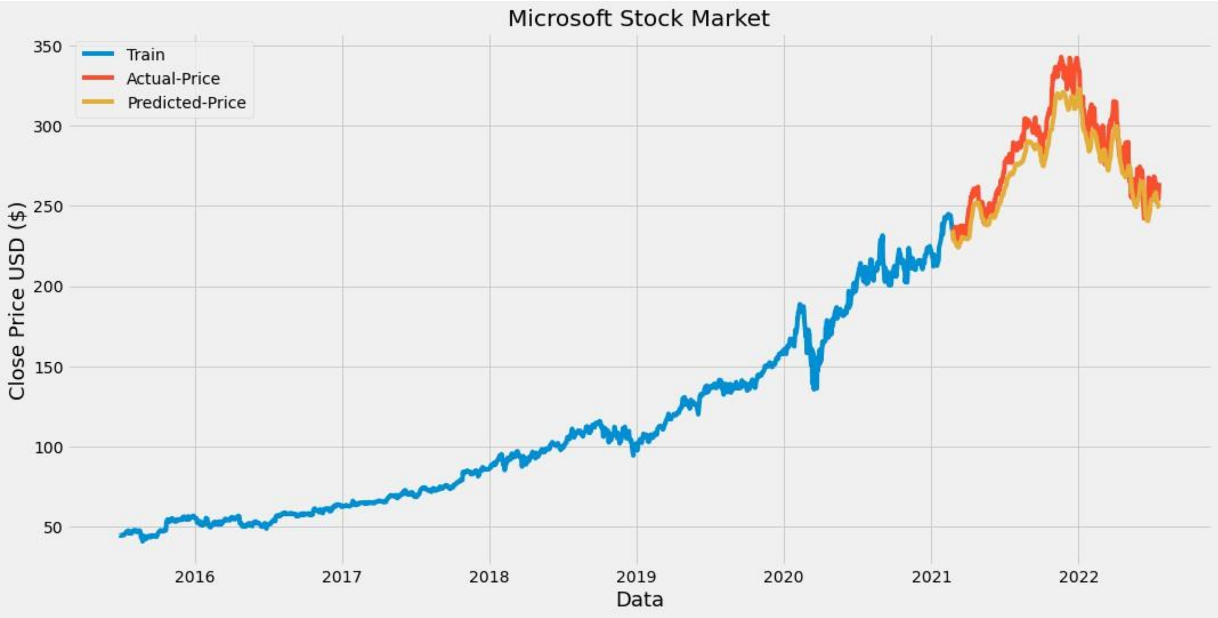


Figure 13: Comparison of predict data with actual test data for Microsoft

Figure 14: Comparison of predict data with actual test data for Google

Overall, after using this model, it is shown that the average to best result is stock price forecast. But this order figure may change if we increase the number of stocks.

Chapter 4

Conclusion

4.1 Conclusion

In this study, we utilize an RNN-like deep learning LSTM model to forecast Tesla's historical stock price. The proposed model's accuracy results are developed by splitting the trade data into training and testing samples at a ratio of 80% and 20%, respectively. We chose MAE, RMSE, MAPE, and R2-score as our model assessment metrics among our suggested models because these metrics are required for calculating accuracy rate and provide the main level. We discovered that these metrics have the following values: 34.2218, 46.307455, 0.0406633, and 0.92143, combined. Figure 4.2 also shows that the anticipated and real price graphs are pretty similar, which generally brings us to the conclusion that LSTMs are an excellent choice for time series prediction. In the future, a dataset significantly larger than the one we utilized in this study will be used to predict stock values, and integrating additional variables and components into our datasets will also assist increase accuracy.

4.2 Future Work

We will continue to research more feature values as we train and refine our model. In order to select a good model to improve prediction accuracy, we will compare the benefits and drawbacks of several neural network models used in stock prediction.

References

[1] Reddy, V. K. S. (2018). Stock market prediction using machine learning. International Research Journal of Engineering and Technology (IRJET), 5(10), 1033- 1035.

[2] Usmani, M., Adil, S. H., Raza, K., & Ali, S. S. A. (2016, August). Stock market prediction using machine learning techniques. In 2016 3rd international conference on computer and information sciences (ICCOINS) (pp. 322-327). IEEE.

[3] Parmar, I., Agarwal, N., Saxena, S., Arora, R., Gupta, S., Dhiman, H., & Chouhan, L. (2018, December). Stock market prediction using machine learning. In 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC) (pp. 574-576). IEEE.

[4] Agrawal, S., Thakkar, D., Soni, D., Bhimani, K., & Patel, C. (2019). Stock market prediction using machine learning techniques. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5(2), 1099-1103.

[5] Mankar, T., Hotchandani, T., Madhwani, M., Chidrawar, A., & Lifna, C. S. (2018, January). Stock market prediction based on social sentiments using machine learning. In 2018 International Conference on Smart City and Emerging Technology (ICSCET) (pp. 1-3). IEEE.

[6] Emioma, C. C., & Edeki, S. O. (2021). Stock price prediction using machine learning on least-squares linear regression basis. Journal of Physics: Conference Series, 1734, 012058. https://doi.org/10.1088/1742-6596/1734/1/012058

[7] Vijayarani, S., Suganya, E., & Jeevitha, T. PREDICTING STOCK MARKET USING MACHINE LEARNING ALGORITHMS. IRJMETS, December-2020.

[8] Sriram, S., & Rangarajan, S. Stock Market Prediction using Logistic Regression Analysis-A Pilot Study. [9] Hernández-Álvarez, M., Hernández, E. A. T., & Yoo, S. G. (2019, February). Stock Market Data Prediction Using Machine Learning Techniques. In International Conference on Information Technology & Systems (pp. 539-547). Springer, Cham.

[10] Pothuganti, K. (2021). Long Short-Term Memory (LSTM) Algorithm Based Prediction of Stock Market Exchange. International Journal of Research Publication and Reviews, 2(1), 90-93.

[11] Pahwa, N., Khalfay, N., Soni, V., & Vora, D. (2017). Stock prediction using machine learning a review paper. International Journal of Computer Applications, 163(5), 36-43.

[12] Kompella, S., & Chakravarthy Chilukuri, K. C. C. (2020). Stock Market Prediction Using Machine Learning Methods. International Journal of Computer Engineering and Technology, 10(3), 2019.

[13] Mokalled, Wassim El-Hajj Mariam, and Mohamad Jaber. "Automated stock price prediction using machine learning." In Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019), pp. 16-24. 2019.

[14] Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. Procedia Computer Science, 167, 599-606.

[15] Parray, I. R., Khurana, S. S., Kumar, M., & Altalbe, A. A. (2020). Time series data analysis of stock price movement using machine learning techniques. Soft Computing, 24(21), 16509-16517.

[16] Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNNsliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE. [17] Mottaghi, N., & Farhangdoost, S. (2021). Stock Price Forecasting in Presence of Covid-19 Pandemic and Evaluating Performances of Machine Learning Models for Time-Series Forecasting. arXiv preprint arXiv:2105.02785.

[18] Mehtab, S., & Sen, J. (2020). Stock price prediction using convolutional neural networks on a multivariate timeseries. arXiv preprint arXiv:2001.09769.

[19] Mandi, R., & Skanda, G. (2021). Analysis of Stock Market using Machine Learning. International Journal of Engineering and Applied Physics, 1(2), 162-167.

[20] Karmiani, D., Kazi, R., Nambisan, A., Shah, A., & Kamble, V. (2019, February). Comparison of predictive algorithms: backpropagation, SVM, LSTM and Kalman Filter for stock market. In 2019 Amity International Conference on Artificial Intelligence (AICAI) (pp. 228-234). IEEE.

[21] Sen, J., Mehtab, S., & Dutta, A. (2021). Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models.

[22] Nelson, D. M., Pereira, A. C., & de Oliveira, R. A. (2017, May). Stock market's price movement prediction with LSTM neural networks. In 2017 International joint conference on neural networks (IJCNN) (pp. 1419-1426). IEEE.