Stock Market Prediction Using LSTM

A Project Report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Harshit Garg (181500254)

Hardik Gupta (181500244)

Subhi Varshney (181500730)

Kashish Chaudhary (181500316)

Under the Guidance of

Dr. Ashish Sharma

Department of Computer Engineering & Applications

Institute of Engineering & Technology

GLA University

Mathura- 281406, INDIA

May, 2022

Department of Computer Engineering and Applications
GLA University, 17 km Stone, NH#2, Mathura-Delhi Road,
P.O. Chaumuhan, Mathura-281406 (U.P.)

Declaration

I hereby declare that the work which is being presented in the B.Tech. Project **"Stock Market Prediction using LSTM"**, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my own work carried under the supervision of **Dr. Ashish Sharma, Associate Professor, Dept. of CEA, GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign ______________________

Harshit Garg

181500254

Sign ______________________

Hardik Gupta

181500244

Sign ______________________

Subhi Varshney

181500730

Sign ______________________

Kashish Chaudhary

181500316

Certificate

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

Supervisor
(**Dr. Ashish Sharma**)
Associate Professor
Dept. of Computer Engg, & App.

Project Co-ordinator
(Dr. Mayank Srivastava)
Associate Professor
Dept. of Computer Engg, & App.

Program Co-ordinator
(**Dr Rakesh Kumar Galav**)
Assistant Professor
Dept. of Computer Engg, & App.

Date:

Acknowledgment

We would like to express our deep gratitude to our project supervisor Dr. Ashish Sharma, Associate Professor, Dept. of Computer Engineering, & Applications, GLA University, for his guidance with unsurpassed knowledge and immense encouragement.

We are grateful to Dr. Rohit Agrawal, Head of the Computer Engineering and Applications Department, for providing us with the necessary resources to complete the project.

We want to thank Dr. Mayank Srivastava, the project coordinator, for her unwavering support and encouragement. We would like to express our gratitude to all teaching faculty in the Department of CEA, whose suggestions during the review process aided us in completing our project. We'd like to express our gratitude to our parents, friends, and classmates for their support throughout the project. Last but not least, we want to express our gratitude to everyone who has helped us complete this project successfully, whether directly or indirectly.

Harshit Garg	181500254
Hardik Gupta	181500244
Subhi Varshney	181500730
Kashish Chaudhary	181500316

Abstract

We attempt to use a machine learning approach to predict stock prices in this project. In order to forecast stock prices, machine learning is used effectively. The goal is to forecast stock prices so that investors can make more informed and precise investment decisions. To improve stock prediction accuracy and issue profitable trades, we propose a stock price prediction system that integrates mathematical functions, machine learning, and other external factors.

Stocks are divided into two categories. The term "day trading" is commonly used to describe intraday trading. Interday traders keep positions in securities for at least one day and often for several days, weeks, or months. Because they can store past information, LSTMs are extremely useful in sequence prediction problems. This is significant in our situation because a stock's previous price is critical in predicting its future price. While predicting a stock's actual price is difficult, we can create a model that predicts whether the price will rise or fall.

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER -1

Introduction

1.1 OVERVIEW AND MOTIVATION

People can buy and sell currencies, stocks, equities, and derivatives over virtual platforms supported by brokers in the financial market, which is a dynamic and composite system. The stock market allows investors to purchase shares of publicly traded companies through exchange or over-the-counter trading. This market has provided investors with the opportunity to make money and live a prosperous life by investing small amounts of money at a low risk compared to the risk of starting a new business or the need for a high-paying job. Many factors influence stock markets, resulting in market uncertainty and high volatility.

Although humans can take orders and submit them to the market, automated trading systems (ATS) that are run by computer programmes can perform better and with more momentum than humans in submitting orders. To evaluate and control the performance of ATSs, however, risk strategies and safety measures based on human judgments must be implemented. When creating an ATS, many factors are taken into account, such as the trading strategy to be used, complex mathematical functions that reflect the state of a specific stock, machine learning algorithms that enable the prediction of future stock value, and specific news about the stock being studied.

Time-series prediction is a widely used technique in a variety of real-world applications, including weather forecasting and financial market forecasting. It predicts the result in the next time unit using continuous data over a period of time. In training, many time-series prediction algorithms have proven to be effective. Recurrent Neural Networks (RNN) and their special types, Long-short Term Memory (LSTM) and Gated Recurrent Units, are now the most widely used algorithms (GRU). The stock market is a common area where time-series data is presented, and many researchers have studied it and proposed various models. The LSTM model is used to predict the stock price in this project.

1.2 Objective

In data analysis, time series forecasting and modelling are crucial. Time series analysis is a subset of statistics that is widely used in fields like econometrics and operations research. In analytics and data science, time series is widely used. Stock prices are highly volatile, and their value is determined by a variety of factors. The main goal of this project is to use long short term memory to predict stock prices (LSTM).

1.3 Issue and Challenges

Many researchers have attempted stock market prediction using LSTM, but we have discovered that as the size of the dataset changes, the results change and the accuracy decreases.

1.4 Contribution

We train our model with various dataset sizes to determine how large of a dataset we require for it to provide highly accurate results.

Chapter 2

Literature Review

2.1 Literature Work

Hossain et al. [1] developed a hybrid model based on LSTM and Gated Recurrent Unit (GRU) that worked on S&P historical time series data and extracted the data using Yahoo's yfinance API. When compared to individual LSTM or GRU layer forecasts, the hybrid model had an MAE of roughly 0.023 and produced substantially more accurate predictions.

For forecasting the Bombay Stock Exchange (BSE) Sensex data, Shah et al. [2] evaluated LSTM and Deep Neural Network (DNN). Both models were suitable for daily forecasts and had an RMSE of around 1%, while the LSTM model was shown to be more suitable for weekly predictions.

Zhang et al. [3] developed a new stock price trend prediction system that predicts stock prices. For data, it employs the TA- Lib open source library, and for analysis and prediction, it employs Random Forest, Imbalanced learning, and Feature selection. It has a 67.5 percent accuracy and a 3.7 percent standard deviation, making it more ideal for predicting stock values over a longer period of time, such as 30 to 40 days.

Creighton et al. [4] used a hybrid model of exponential smoothing, ARIMA, and backpropagation neural network (BPNN) on the S&amp 500 and S&P 400 daily closing indexes to complement linear and nonlinear predictions. It is based on the KoNstanz Information MinEr (KNIME) analytics platform and works well for predicting weekly data with less noise and more linear growth, but not for daily predictions. The model has a 45.1 percent directional accuracy, an MAE of 16.68, an MSE of 434.121, and an RMSE of around 20.836.

On the NSE data, Selvin et al. [5] employed three algorithms: LSTM, RNN, and Convolutional Neural Network (CNN) - Sliding Window model. The sliding window was 100 minutes long, with 90 minutes for information and 10 minutes for prediction. The results of CNN were found to be superior to those of LSTM and RNN. RNN had an MAE of 5.13 percent, LSTM had an MAE of 5.31 percent, and CNN had an MAE of 4.98 percent, according to the model. ARIMA, a linear model with an MAE of 29.87 percent, was outperformed by deep learning.

For predictions, Sadia et al. [6] used Random Forest and SVM on a historical dataset from Kaggle. The data was trained on OHCLV, trade, and value parameters, yielding an SVM accuracy of 78.7% and a Random forest accuracy of 80.8 percent.

Deepak et al. [7] took a different technique, using SVM with RBF kernel on the BSE Sensex dataset to forecast stock market situations for the coming week, day, and minute. Depending on the share considered, the model had an accuracy of 80 to 85 percent. Tab. 1 outlines the numerous stock market prediction algorithms used on various datasets, as well as their accuracy and mean errors.

Persio et al. [8] employed LSTM, GRU, and Multilayer RNN to detect short term ups and downs in the Google stock price data for financial forecasting. It was also subjected to the Adam optimization technique. For a 5-day data projection, the model had a 72 percent accuracy rate.

2.2 A Comparison of Different Prediction Techniques

Paper Title	**Author**	**Model Used**	**Evaluation**
Hybrid Deep Learning Model for Stock Price Prediction	Hossain et al.	LSTM- GRU hybrid	MSE- 0.00098 MAE- 0.023
A Comparative Study of LSTM and DNN for Stock Market Forecasting	Shah et al.	LSTM, DNN	RMSE- 1%

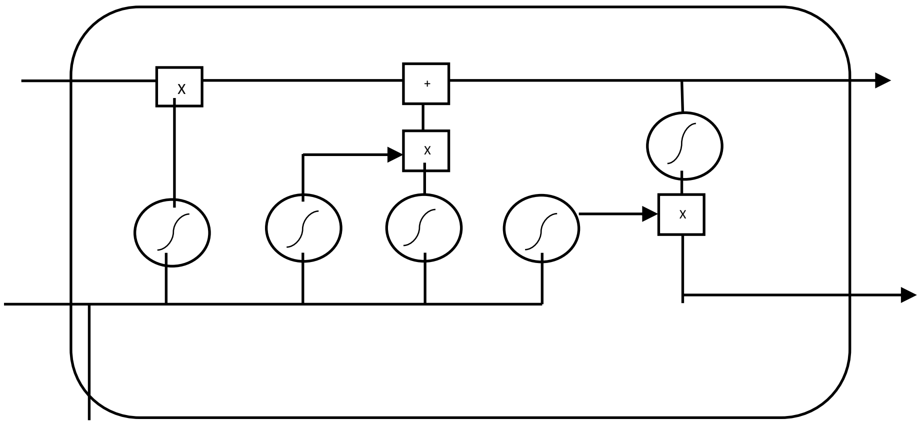
A Novel Data-driven Stock Price Trend Prediction System	Zhang et al.	Random Forest	Accuracy- 67.5% Std deviation- 3.7%
Towards Building a Hybrid Model for Predicting Stock Indexes	Creighton et al.	ARIMA-BPNN hybrid	MAE- 16.68 MSE- 434.121 RMSE- 20.836 Accuracy- 45.1%
Stock price prediction using LSTM, RNN and CNN-sliding window model	Selvin et al.	RNN, LSTM, CNN	MAE - 5.13% (RNN) 5.31% (LSTM) 4.98% (CNN)
Stock Market Prediction Using Machine Learning Algorithms	Sadia et al.	SVM, Random Forest	Accuracy-78.7% (SVM) 80.8% (Random forest)
Machine Learning Approach in Stock Market Prediction	Raut Sushrut Deepak	RNN, LSTM	Accuracy- 80 to 85%
Recurrent Neural Networks Approach to the Financial Forecast of Google Assets	Persio et al.	GRU	Accuracy- 72%

Chapter 3

Proposed system

We propose to use the LSTM (Long Short Term Memory) technique to give effective stock price prediction and to evaluate how well our model has been trained by considering various aspects such as epochs and data size.

3.1 LSTM – Overview



LSTMs are a type of RNN that may capture context-specific temporal dependencies over lengthy time periods. Each LSTM neuron is a memory cell that may store other data and keeps track of its own cell state. An LSTM neuron additionally takes in its old cell state and outputs its new cell state, whereas neurons in standard RNNs only take in their prior hidden state and the current input to output a new hidden state.

The following three components or gates, make up an LSTM memory cell, as shown in above figure:

1. Forget gate: This gate determines when particular elements of the cell state should be replaced with more recent data. It produces values that are near to 1 for sections of the cell state that should be kept and zero for values that should be ignored.

2. Input gate: This component of the network learns the conditions under which any information should be kept (or updated) in the cell state based on the input (i.e., previous output o(t-1), input x(t), and prior cell state c(t-1).

3. Output gate: This section determines what information is carried forward (i.e., output o(t) and cell state c(t)) to the next node in the network based on the input and cell state.

As a result, LSTM networks are suitable for examining how price changes in one stock affect the prices of numerous other stocks over time. They can also decide (dynamically) how long information about certain previous trends in stock price movement should be preserved in order to better predict future trends in stock price variation.

3.2 Advantage of LSTM

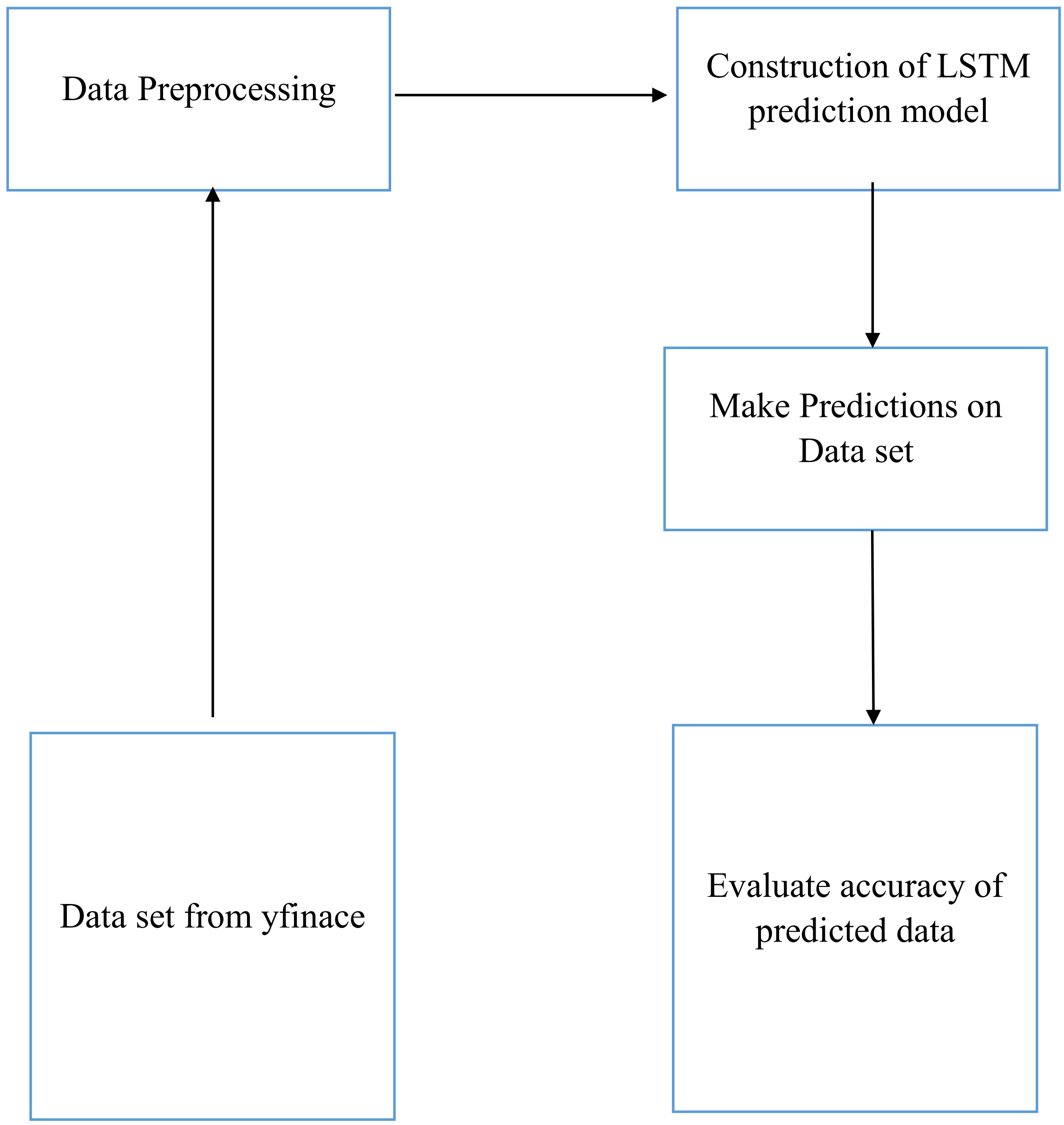
The ability of LSTM to read intermediate context is its key advantage. Without explicitly applying the activation function inside the recurring components, each unit remembers information for a long or short length of time. The release of the forget gate, which ranges between 0 and 1, is the only way for any cell state to be repeated. To put it another way, the LSTM cell's forgetting gateway is in charge of both the hardware and the function of cell state activation. As a result, instead of explicitly increasing or decreasing in each step or layer, the data from the preceding cell can flow through the unmodified cell, and the instruments can convert to their suitable values over a limited time. Because the amount stored in the memory cell is not transformed in a recurrent fashion, the gradient does not cease when trained to distribute rearward, allowing LSTM to solve a perishable gradient problem.

3.3 Methodology

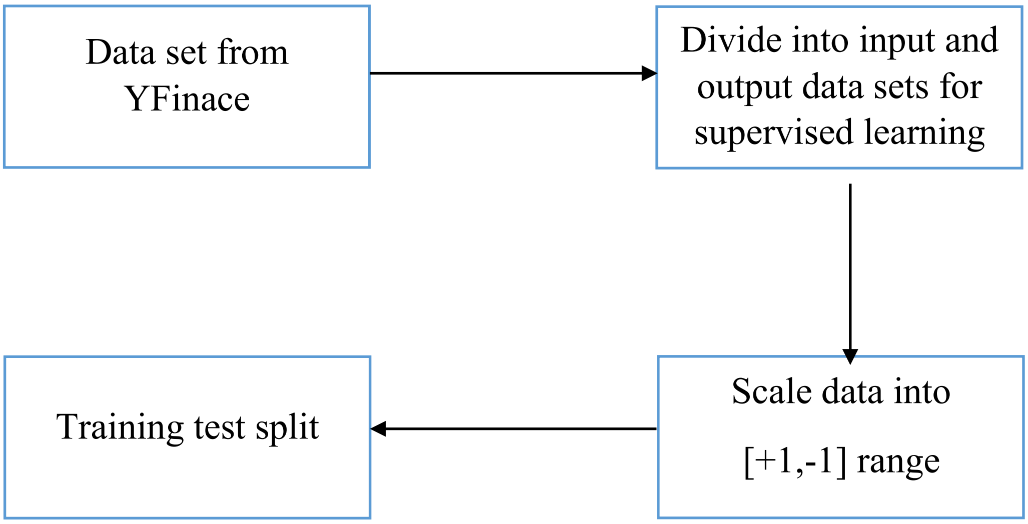
```
# LSTM

Inputs: dataset

Outputs: Predicted data


# Split dataset into 75% training and 25% testing data

size = length(dataset) * 0.75

train = dataset [0 to size]

test = dataset [size to length(dataset)]


# Procedure to fit the LSTM model

Procedure LSTMAlgorithm (train, test, train_size, epochs)

X = train

y = test

model = Sequential ()

model.add(LSTM(50), stateful=True)

model.compile(optimizer='adam', loss='mse')

model.fit(X, y, epochs=epochs, validation_split=0.2)

return model


# Procedure to make predictions

Procedure getPredictonsFromModel (model, X)

predictions = model.predict(X)
```

```
return predictions

#Fit the lstm model

model = LSTMAlgorithm(train,epoch)


#Make Predictions

pred = model.predict(train)


#Validate the model

n = len(dataset)
```

3.4 System architecture



3.5 Data Pre-processing



Tiingo API, Yahoo, and Google Finance are all good places to look for stock market data. These websites provide APIs through which stock datasets from multiple firms can be downloaded by simply providing parameters.

The following processes are used to convert the data into a format that can be used with the prediction model:

1. For supervised learning, transform time-series data into input-output components.

2. Changing the data's scale to [-1, +1].

Chapter 4

Implementation and Result Analysis

Loading the data set

Taking apple stock data from 2007-03-01 to 2022-01-28 We take closing price for our prediction.

We should reset the index

```
df1=df.reset_index()['close']
```

Train and Test Split

Whenever training Timeseries data we should divide the data differently we should train the data with the respective date.

```
data_training=pd.DataFrame(df['Close'][0:int(len(df)*0.70)])
data_testing=pd.DataFrame(df['Close'][int(len(df)*0.70):int(len(df))])
```

Here we will use min-max scalar to transform the values from 0 to 1.We should reshape so that we can use fit transform.

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
data_training_array=scaler.fit_transform(data_training)
```

And Train data and Test data is ready.

Data Preprocessing

Consider the time steps, as well as how previous data should be considered if we want to predict the price of a stock in a day.

The timestep value will now be set to 100. Let's divide the data into two groups: X and Y. The first 100 elements are used as your first record in the 0th iteration, while the 101 elements are placed in the X. The Y will be used to display the 100 elements.

```
x_train=[]

y_train=[]

for i in range(100,data_training_array.shape[0]):

   x_train.append(data_training_array[i-100:i])

   y_train.append(data_training_array[i,0])

x_train, y_train = np.array(x_train),np.array(y_train)
```

LSTM

We will be using a sequential model and adding the layers of the LSTM as said, in the above sentence. The first layer should be the time step in 1.

```
from keras.layers import Dense,Dropout,LSTM

from keras.models import Sequential

model = Sequential()

model.add(LSTM(units = 50, activation='relu', return_sequences= True,

           input_shape = (x_train.shape[1],1)))

model.add(Dropout(0.2))

model.add(LSTM(units = 60, activation='relu', return_sequences= True))

model.add(Dropout(0.3))

model.add(LSTM(units = 80, activation='relu', return_sequences= True))
```

```
model.add(Dropout(0.4))

model.add(LSTM(units = 120, activation='relu' ))

model.add(Dropout(0.5 ))

model.add(Dense(units=1))
```

Now the final part is to fit the X_train and the y_train.

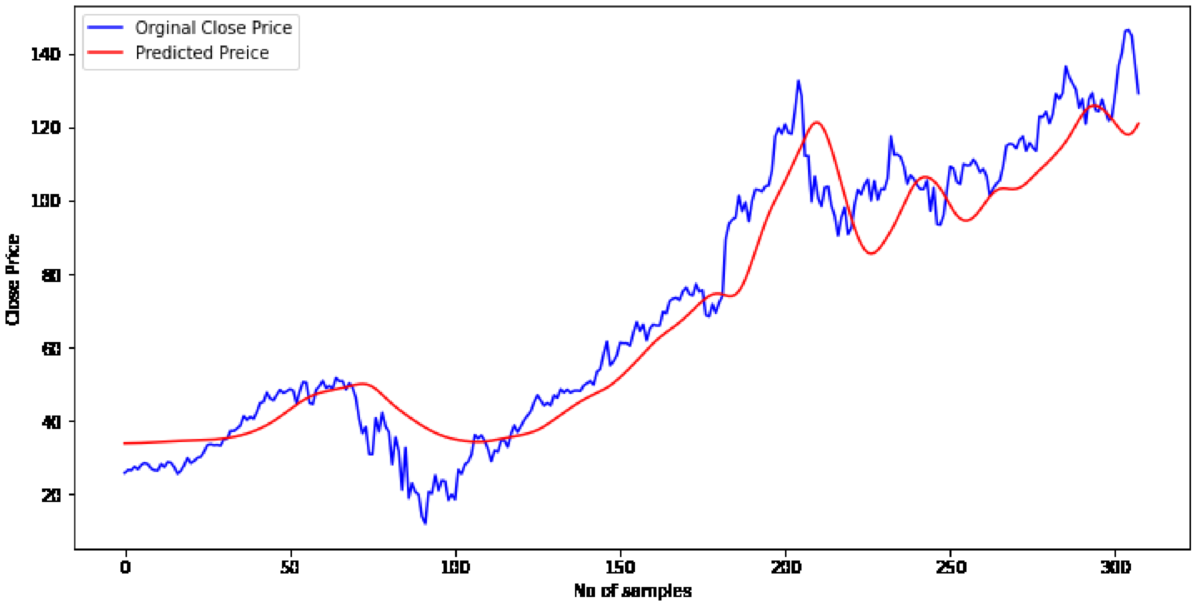
Predictions

Predict both the X_train and the X_test, now let’s scaler inverse transform because we want to see the root mean square performance.

```
train_predict=model.predict(X_train)

test_predict=model.predict(X_test)

train_predict=scaler.inverse_transform(train_predict)

test_predict=scaler.inverse_transform(test_predict)
```

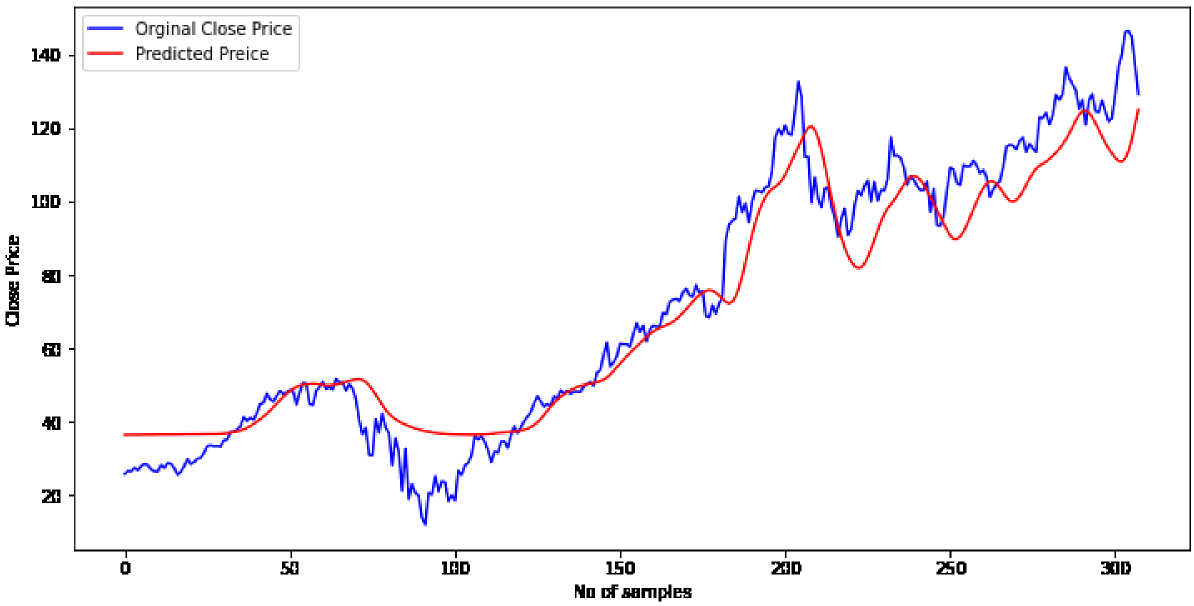
Comparison of Actual and Predicted Price with different Dataset size and epochs

Dataset size – 1025



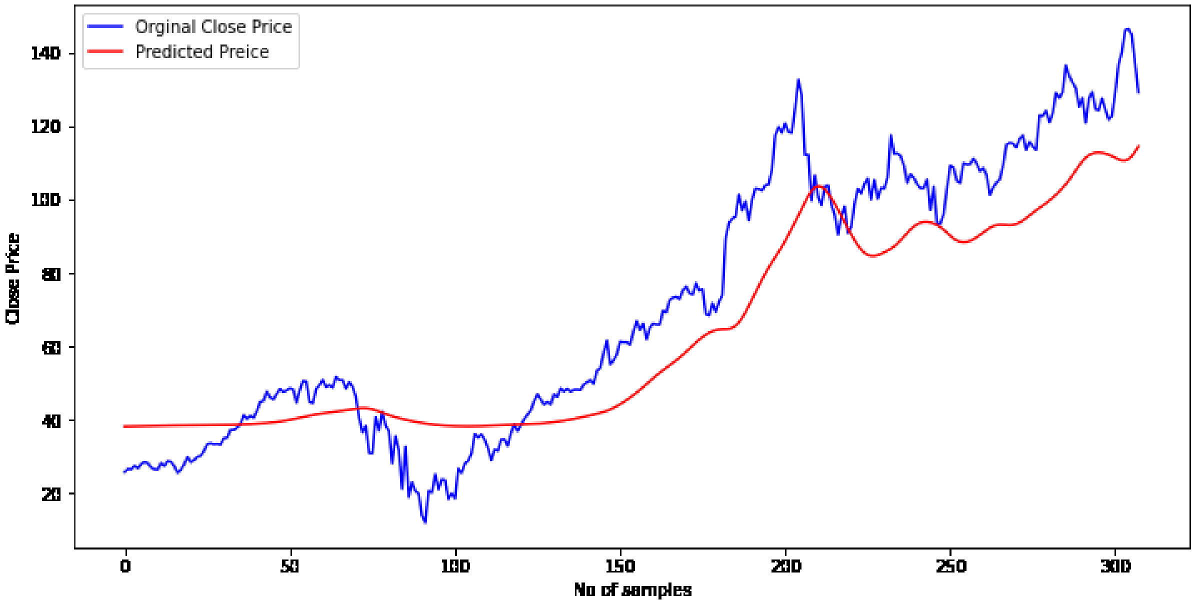
Dataset size - 1025

Epochs - 25



Dataset size - 1025

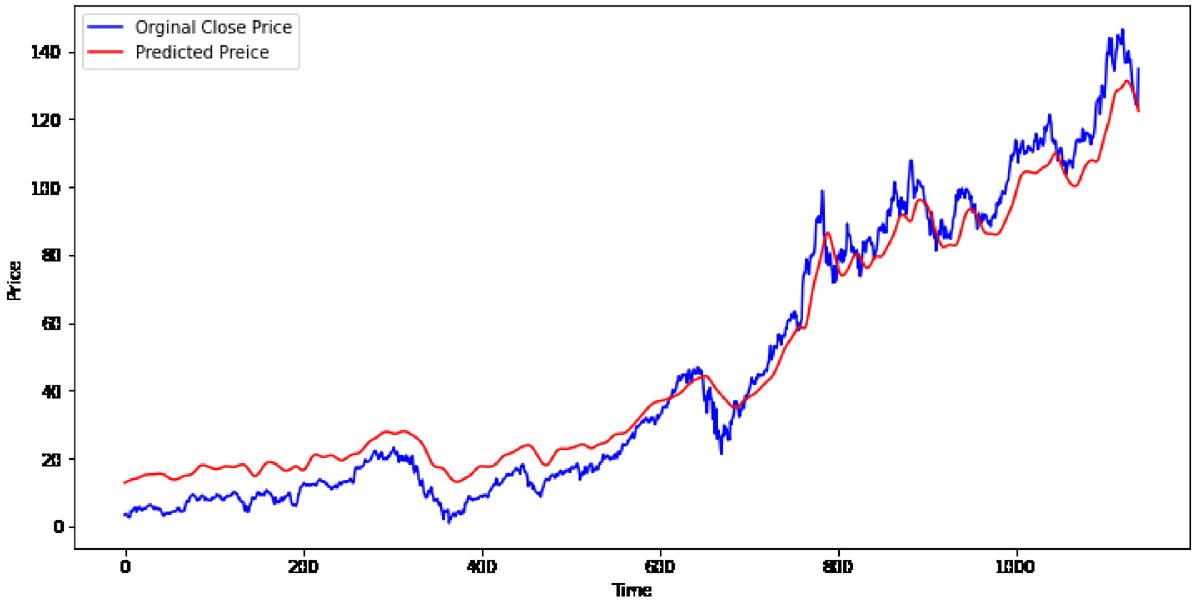
Epochs – 50



Dataset size - 1025

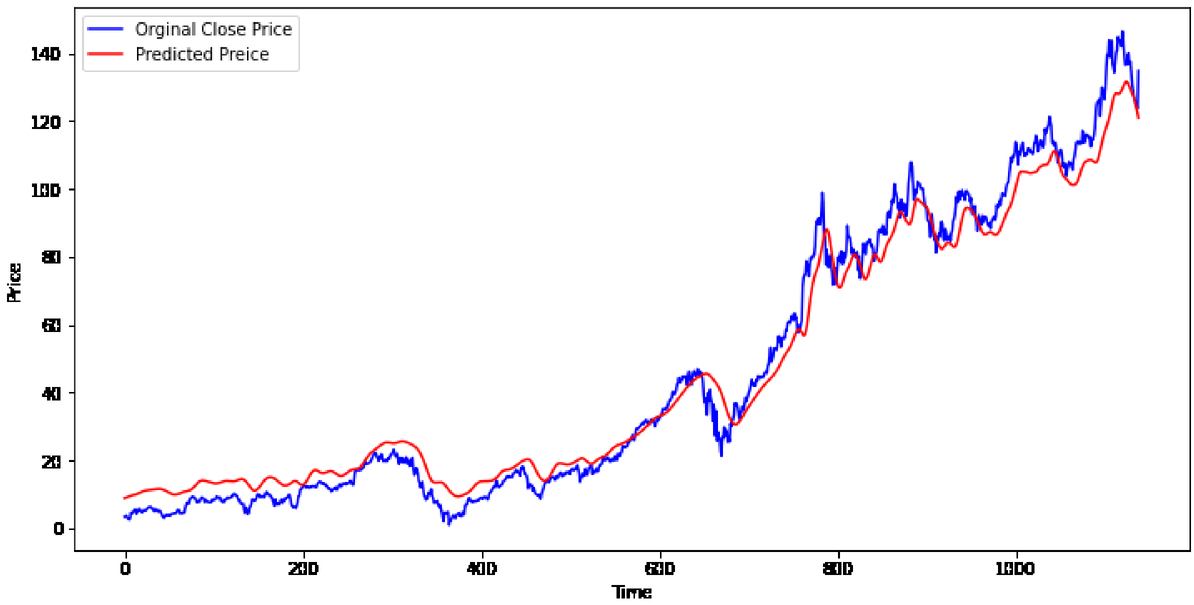
Epochs – 100

Dataset Size - 3795



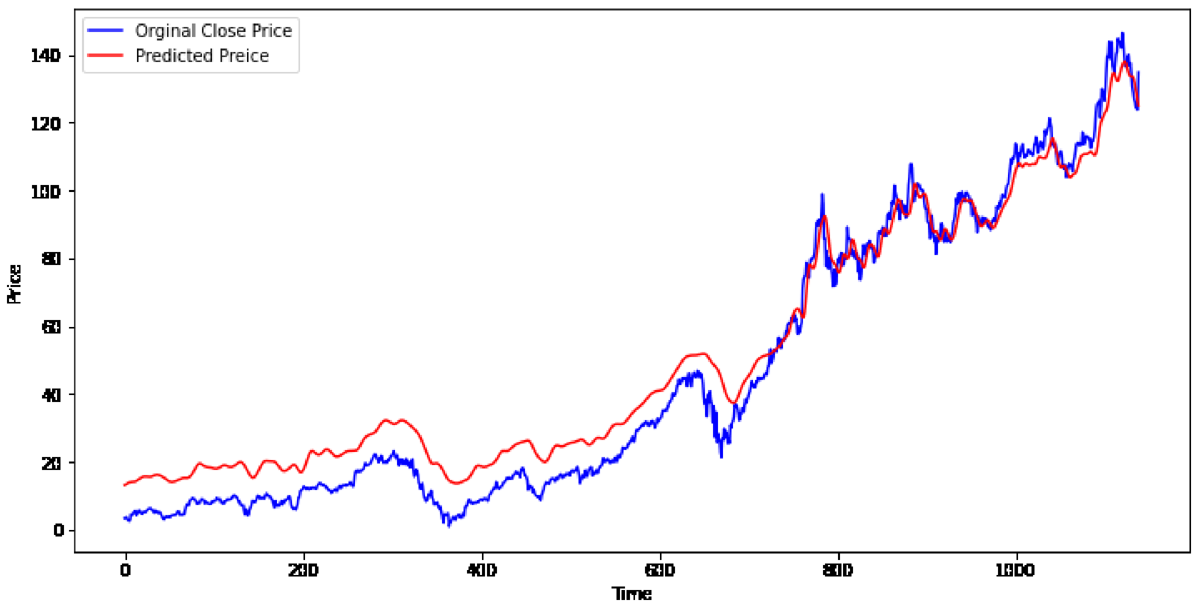
Dataset size - 3795

Epochs – 25



Dataset size - 3795

Epochs – 50



Dataset size - 3795

Epochs – 100

Result Analysis

Dataset size - 1025

	25 Epochs	50 Epochs	100 Epochs
Mean Absolute Error	8.5813	7.655	13.186
Mean Squared Error	112.680	102.796	245.170
Root Mean Square Error	10.615	10.138	15.657
R2 Score	0.916	0.923	0.818

Dataset size - 3795

	25 Epochs	50 Epochs	100 Epochs
Mean Absolute Error	7.261	5.213	7.42
Mean Squared Error	66.310	40.160	70.10
Root Mean Square Error	8.143	6.33	8.37
R2 Score	0.96	0.976	0.95

An analysis of the results also shows that when the dataset is larger, the models are more accurate. With more data, the model can flesh out more patterns and adjust the layer weights more effectively.

The stock market is, at its core, a reflection of human emotions. There are limitations to pure number crunching and analysis. An addition to this stock prediction system could be a news feed analysis from social media platforms like Twitter, where emotions are gauged from the articles. This sentiment analysis can be combined with the LSTM to improve weight training and accuracy.

Chapter 5 Conclusion

Many investors all over the world are interested in stock investing. Making a decision, on the other hand, is a difficult task due to the numerous factors involved. Investors are eager to forecast the stock market's future after making successful investments. Even a small improvement in performance can have a huge impact. By providing supporting information such as future stock price guidance, a good forecasting system can assist investors in making more accurate and profitable investments. Other factors, such as politics, economic growth, financial matters, and the atmosphere on social media, could influence prices in addition to historical prices. Emotional analysis has been shown in numerous studies to have a significant impact on future prices. As a result, combining technical and fundamental analysis can yield very accurate predictions.

This paper proposes the LSTM model for forecasting stock values. We can now say that if we take a large data set and train the LSTM model for 50 epochs, we will get highly accurate predicted data.

Our model's output has yielded some promising results. The results of the tests show that our model can track the evolution of closing prices.

We will try to find a method to forecast future data with the same accuracy in our future work, as well as develop a mobile application that shows all of the forecasting.

References

[1] Mohammad Asiful Hossain, Rezaul Karim, "Ruppa Thulasiram Hybrid Deep Learning Model for Stock Price Prediction", IEEE Symposium Series on Computational Intelligence SSCI, pp.1837-1844 (2018).

[2] Dev Shah, Wesley Campbell, Farhana H. Zulkernine, "A Comparative Study of LSTM and DNN for Stock Market Forecasting", IEEE International Conference on Big Data (2018)

[3] Jing Zhang, Shicheng Cui, Yan Xu, Qianmu Li, and Tao Li, "A Novel Data-driven Stock Price Trend Prediction System", Elsevier Expert Systems with Applications, Vol. 97, pp.60–69 (2018).

[4] Jonathan Creighton, and Farhana H Zulkernine, "Towards Building a Hybrid Model for Predicting Stock Indexes ", IEEE International Conference on Big Data (2017).

[5] Sreelekshmy Selvin, R Vinayakumar, E. A Gopalakrishnan, Vijay Krishna Menon, "Stock price prediction using LSTM, RNN and CNN-sliding window model", IEEE Conference Paper, doi: 10.1109/ICACCI.2017.8126078, pp.1643- 1647 (2017).

[6] K. Hiba Sadia, Aditya Sharma, Adarrsh Paul, SarmisthaPadhi, Saurav Sanyal , "Stock Market Prediction Using Machine Learning Algorithms", International Journal of Engineering and Advanced Technology, Vol. 8, No. 4, pp.25-31 (2019).

[7] Raut Sushrut Deepak, Shinde Isha Uday, and Dr D Malathi," Machine Learning Approach in Stock Market Prediction", International Journal of Pure and Applied Mathematics, Vol. 115, No. 8, pp.71-77 (2017).

[8] Luca Di Persio, and Oleksandr Honchar, "Recurrent Neural Networks Approach to the Financial Forecast of Google Assets", International Journal of Mathematics and Computers in Simulation, Vol. 11, pp.7-13 (2017).

[9] Usmani, Mehak, Syed Hasan Adil, Kamran Raza, and Syed Saad Azhar Ali, "Stock market prediction using machine learning techniques," 3rd international conference on computer and information sciences (ICCOINS), pp. 322-327. IEEE, 2016.

[10] Nayak, Aparna, MM Manohara Pai, and Radhika M. Pai, "Prediction models for Indian stock market" Procedia Computer Science 89 (2016): 441-449.

[11] Pahwa, Kunal, and Neha Agarwal, "Stock market analysis using supervised machine learning", 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). IEEE, 2019.

[12] Roondiwala, Murtaza, Harshal Patel, and Shraddha Varma, "Predicting stock prices using LSTM", International Journal of Science and Research (IJSR) 6.4 (2017): 1754-1756.