Stock Price Prediction

Minor Project Report

Submitted for the partial fulfillment of the degree of

Bachelor of Technology

In

Artificial intelligence and Data Science

Submitted By

Manish Sharma 0901AD211028

Gaurav Chaudhary 0901AD211015

UNDER THE SUPERVISION AND GUIDANCE OF

Prof. Mir Shahnawaz

Assistance Professor

Centre for Artificial Intelligence

माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर (म.प्र.), भारत

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA

Deemed to be University

(Declared under Distinct Category by Ministry of Education, Government of India)

NAAC ACCREDITED WITH A++ GRADE

January - June 2024

DECLARATION BY THE CANDIDATE

I hereby declare that the work entitled **"Stock Price Prediction"** is my work, conducted under the supervision of **Prof. Mir Shahnawaz, Assistance Professor** during the session Jan-May 2024. The report submitted by me is a record of bonafide work carried out by me.

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Manish Sharma

0901AD211028

Date: 23-04-2024
Place: Gwalior

This is to certify that the above statement made by the candidates is correct to the best of my knowledge and belief.

Guided By:

Prof. Mir Shahnawaz
Assistance Professor
Centre for Artificial Intelligence
MITS, Gwalior

Departmental Project Coordinator

Dr. Tej Singh
Assistance Professor
Centre for Artificial Intelligence
MITS, Gwalior

Approved by Coordinator

Dr. R.R. Singh
Coordinator
Centre for Artificial Intelligence
MITS, Gwalior

PLAGIARISM CHECK CERTIFICATE

This is to certify that I/we, a student of B.Tech. in **Centre for Artifical intelligence** have checked my complete report entitled **"Stock Price Prediction"** for similarity/plagiarism using the "Turnitin" software available in the institute.

This is to certify that the similarity in my report is found to be …… which is within the specified limit (30%).

The full plagiarism report along with the summary is enclosed.

Manish Sharma

0901AD211028

Checked & Approved By:

Dr. Pawan Dubey
Assistance Professor
Centre for Artificial Intelligence
MITS, Gwalior

ABSTRACT

Our project's objective is to identify several methods for projecting a certain stock's future price based on historical data and numerical news indicators. The goal is to build a portfolio of several stocks to lower risk. In order to do this, we used supervised learning techniques to forecast stock prices by analyzing market data.

Predicting stock prices is still a crucial responsibility in the financial markets, but it is exceedingly difficult. In order to accurately estimate stock values, we used time series analytic techniques in this study. To commence this project, we first gather historical stock price data of Apple (AAPL) from Tiingo. We next preprocess the data and apply many time series models, such as LSTM neural networks and ARIMA, To improve model performance, selection and feature engineering are used.

In the end, our research will help stakeholders make more informed decisions in the financial markets by offering helpful insights into the effectiveness of time series analysis for stock price prediction. Investors may now traverse uncertainty with confidence by knowing predictive models and their workings, opening up new possibilities for wealth creation and risk avoidance.

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Artificial Intelligence,** for allowing me to explore this project. I humbly thank **Dr. R. R. Singh**, Coordinator, Centre for Artificial Intelligence, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Prof. Mir Shahnawaz**, Centre for Artificial Intelligence, for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Manish Sharma

0901AD211028

CONTENT

Table of Contents

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In the financial markets, predicting stock prices utilizing time series analysis—especially the Long Short-Term Memory (LSTM) method—is crucial. Time series analysis involves analyzing sequential data points to uncover patterns and trends over time. LSTM, a type of recurrent neural network (RNN), excels in capturing long-term dependencies in sequential data, making it suitable for modeling stock price movements.

There are multiple steps in the LSTM approach. First, historical data on stock prices is gathered and preprocessed to eliminate noise and irregularities. The information is then organized into input-output sequences, where the target variable is future stock prices and the input features are historical stock prices. The LSTM model is then trained on this data, learning complex patterns and relationships between past and future prices.

The performance of the LSTM model can be improved by feature engineering by adding further data, such as technical indicators, market sentiment, or economic factors. Moreover, model assessment and hyperparameter adjustment are essential for maximizing the LSTM's predictive accuracy.

Once trained, the LSTM model can generate forecasts of future stock prices based on historical data. With the help of these projections, investors and financial analysts can decide on trading tactics, risk management, and portfolio management in a dynamic market. Overall, LSTM-based time series analysis provides a powerful tool for predicting stock prices and navigating the complexities of financial markets.

CHAPTER 2: LITERATURE SURVEY

The literature surrounding stock price prediction using time series analysis and LSTM methods encompasses a diverse array of studies and findings. Because long-term dependencies in sequential data may be captured by LSTM networks, its applicability in financial forecasting has been thoroughly investigated by researchers. LSTM models frequently outperform conventional time series methods like ARIMA, according to numerous studies that have compared the two. This is especially true when it comes to capturing complex patterns and nonlinear correlations in stock price data.

Additionally, scholars have explored a range of LSTM model building and augmentation topics, such as ensemble approaches, feature engineering, and hyperparameter optimization. In order to increase forecast accuracy, these research have emphasized the significance of adding pertinent features such technical indicators, sentiment analysis, and economic considerations.

Moreover, research on the usefulness of LSTM-based stock price prediction has been done, showing that it can help portfolio managers, financial analysts, and investors make well-informed judgments. Research has also examined the difficulties and constraints posed by LSTM models, such as the lack of data, overfitting, and interpretability of the model.

In general, the research emphasizes how important LSTM-based time series analysis is for predicting stock prices and how it advances financial market predictive modeling methodologies.

CHAPTER 3: DATA COLLECTION/ VISUALISATION

3.1 <u>Data Collection:</u> Data is an important parameter to train the model. For training and testing of the data, data is mandatory. In this project, we have taken dataset from tiingo, that holds the historical data of the stock of the AAPL (Apple). The data is read by pandas_datareader and the parameter passed in tiingo function are dataset name and API key used in tiingo website. We have taken only 1258 rows of data and have 14 lables in it.

CODE :

```
key = "3d4434cb3da000c73a676e1bebed51e1e2017b64"

df = pdr.get_data_tiingo('AAPL', api_key= key)

df.to_csv("AAPLl.csv")

df = pd.read_csv("AAPL.csv")
```

<u>3.2 Data Representation</u> : Before, implementation of model, first we have to import necessary libraries for data preprocessing , data visualisation ,data reading etc. Then data is first loaded in jupyter Notebook, then is presented.

Our dataset is read with the help of key. The data is represented in the form of dataframe.with the help of datareader we have represented the data with the help of key the dataset is of AAPL(Apple)

CODE:

```
import pandas_datareader as pdr

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import tensorflow as tf

import math
```

In [6]:
```
1 df.head()
```

Out[6]:

	symbol	date	close	high	low	open	volume	adjClose	adjHigh	
0	AAPL	2019-04-15 00:00:00+00:00	199.23	199.85	198.01	198.58	17536646	47.948128	48.097341	4
1	AAPL	2019-04-16 00:00:00+00:00	199.25	201.37	198.56	199.46	25696385	47.952941	48.463155	4
2	AAPL	2019-04-17 00:00:00+00:00	203.13	203.38	198.61	199.54	28906780	48.886730	48.946897	4
3	AAPL	2019-04-18 00:00:00+00:00	203.86	204.15	202.52	203.12	24195766	49.062417	49.132210	4
4	AAPL	2019-04-22 00:00:00+00:00	204.53	204.94	202.34	202.83	19439545	49.223664	49.322337	4

In [7]:
```
1 df.tail()
```

Out[7]:

	symbol	date	close	high	low	open	volume	adjClose	adjHigh
1253	AAPL	2024-04-08 00:00:00+00:00	168.45	169.20	168.24	169.03	37216858	168.45	169.20
1254	AAPL	2024-04-09 00:00:00+00:00	169.67	170.08	168.35	168.70	42231444	169.67	170.08
1255	AAPL	2024-04-10 00:00:00+00:00	167.78	169.09	167.11	168.80	49709336	167.78	169.09
1256	AAPL	2024-04-11 00:00:00+00:00	175.04	175.46	168.16	168.34	91070275	175.04	175.46
1257	AAPL	2024-04-12 00:00:00+00:00	176.55	178.36	174.21	174.26	101670886	176.55	178.36

Fig. 1 Data representation

3.3 **EDA :** Exploratory Data Analysis (EDA) is a critical initial step in the data analysis process, where the main-focus is to understand the data, discover patterns, detect anomalies, and test assumptions. EDA is done mainly for making sense of data that is to be used, rather than using sophisticated or statistical models.

- **INFO** : When you run 'data.info()', you'll get an output that gives you information about the DataFrame, which includes the index, column data types, non-null values, and memory usage. This is extremely helpful during the exploratory data analysis (EDA) phase

```
In [9]:  1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1258 entries, 0 to 1257
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   symbol       1258 non-null   object 
 1   date         1258 non-null   object 
 2   close        1258 non-null   float64
 3   high         1258 non-null   float64
 4   low          1258 non-null   float64
 5   open         1258 non-null   float64
 6   volume       1258 non-null   int64  
 7   adjClose     1258 non-null   float64
 8   adjHigh      1258 non-null   float64
 9   adjLow       1258 non-null   float64
 10  adjOpen      1258 non-null   float64
 11  adjVolume    1258 non-null   int64  
 12  divCash      1258 non-null   float64
 13  splitFactor  1258 non-null   float64
dtypes: float64(10), int64(2), object(2)
memory usage: 137.7+ KB
```

Fig. 2 Data information

- **Shape:** The code 'data.shape', will get you an output indicating the number of rows and columns in the DataFrame.

```
In [8]:  1 df.shape

Out[8]: (1258, 14)
```

Fig. 3 Data Shape

- **Describe:** The '.describe()' function is used in Python, particularly with the Pandas library, to generate descriptive statistics of a DataFrame. When you run 'df.describe()', you'll get a summary of descriptive statistics for numerical columns in the DataFrame.

```
In [10]:  1 df.describe()
```

Out[10]:

	close	high	low	open	volume	adjClose	:
count	1258.000000	1258.000000	1258.000000	1258.000000	1.258000e+03	1258.000000	1258
mean	187.792607	189.729718	185.575734	187.527973	6.950130e+07	129.230154	130
std	69.162266	69.809570	68.166079	68.879808	3.785449e+07	44.316573	44
min	106.840000	110.190000	103.100000	104.540000	1.136204e+07	41.870498	42
25%	144.810000	146.475000	142.654150	144.041250	4.341483e+07	93.016412	93
50%	170.105000	171.680000	168.610000	170.030000	6.453489e+07	140.737533	142
75%	198.730000	200.090000	197.245000	198.570000	8.838305e+07	165.725069	167
max	506.090000	515.140000	500.330000	514.790000	3.326072e+08	197.858551	199

Fig. 4 Data Describe

3.4 VISULATION :

Data Visualization is the representation of data using common graphics such as charts, graphs, infographics, etc. these visual representations of information communicate complex data relationships and views of data in an easy-to-understand manner. dataset attributes are represented to get a better picture of the data.

- **Line Graph** : A line chart is a type of chart that displays data as a series of data points connected by lines. It is commonly used to show trends or patterns over time, where the horizontal axis represents time or categories and the vertical axis represents measured numerical values. Line charts are effective for visualizing continuous data and are often used in various fields such as economics, science, engineering, and statistics.

  Code :

```
plt.figure(figsize=(15,8))

plt.title('Close Price History')

plt.plot(df['close'])

plt.xlabel('index',fontsize=18)

plt.ylabel('Close Price USD ($)',fontsize=20)

plt.show()
```

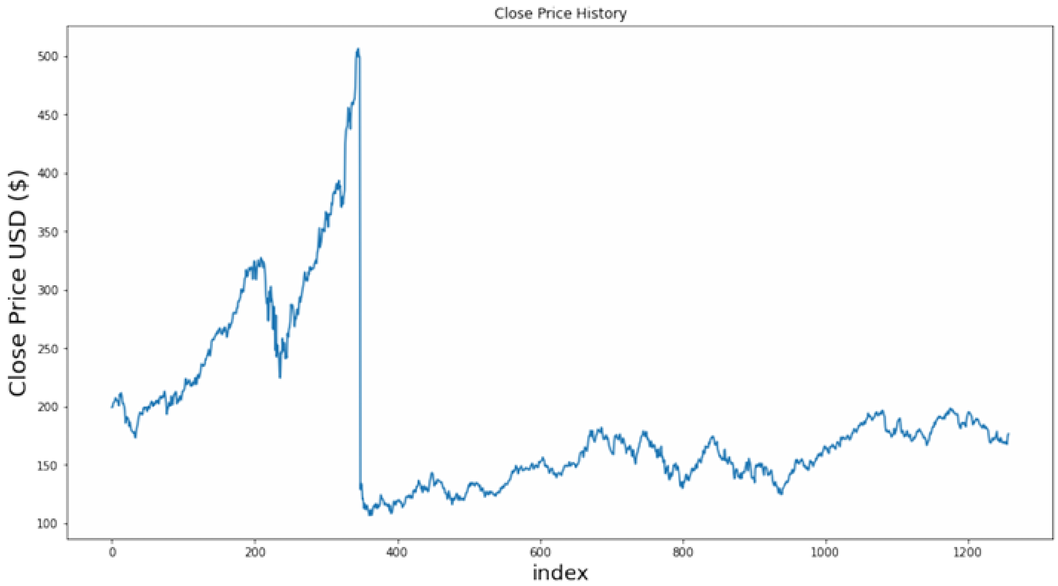

Fig. 5 Close Price History Line Graph

CHAPTER 4: MODEL TRANING

4.1 Train – test split : Here we will split the data into training and test subset. Our model will train on training subset and then will be proceeded to decision making procedure. The ratio of training subset and testing subset is 65% and 35%.

CODE :

```
training_size = int(len(df1)*0.65)

test_size = len(df1)-training_size

train_data,test_data = df1[0:training_size,:],

df1[training_size: len(df1),:1]
```

```
In [21]:  1 training_size, test_size

Out[21]: (817, 441)
```

Fig. 6 Training Size

4.2 Scaling the values :

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(0,1))

df1 = scaler.fit_transform(np.array(df1).reshape(-1,1))

# Create MinMaxScaler with the desired feature range

scaler = MinMaxScaler(feature_range=(0, 1))

# Assuming df1 is your DataFrame or array, reshape it if needed

df1_reshaped = np.array(df1).reshape(-1, 1)


# Fit and transform the data using MinMaxScaler

df1 = scaler.fit_transform(df1_reshaped)
```

```
In [18]:  1 df1

Out[18]: array([[0.23140889],
                [0.23145899],
                [0.24117721],
                ...,
                [0.15263619],
                [0.17082029],
                [0.17460238]])
```

Fig. 7 Reshaped

<u>4.3 Creating LSTM model</u> :

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import LSTM

model = Sequential ()

model.add(LSTM(50,return_sequences = True, input_shape = (100,1)))

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

model.add(LSTM(50))

model.add(Dense(1))

model.compile(loss = "mean_squared_error", optimizer = "adam")

```
In [30]:  1 model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 lstm_1 (LSTM)               (None, 100, 50)           20200

 lstm_2 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 50851 (198.64 KB)
Trainable params: 50851 (198.64 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 8 Model Summary

4.4 Training Till 100 Epoch : An epoch is a single training pass of the whole training dataset in the context of a Long Short-Term Memory (LSTM) model. Training a model in machine learning, especially neural networks, entails iterative optimization, where the model gains performance improvements by learning from the input.

```
In [31]:  1 model.fit(X_train, y_train, validation_data = (X_test, y_test), epochs
val_loss: 1.6023e-04
Epoch 62/100
12/12 [==============================] - 5s 419ms/step - loss: 0.0015 -
val_loss: 1.7118e-04
Epoch 63/100
12/12 [==============================] - 5s 414ms/step - loss: 0.0021 -
val_loss: 1.8693e-04
Epoch 64/100
12/12 [==============================] - 5s 408ms/step - loss: 0.0019 -
val_loss: 3.3789e-04
Epoch 65/100
12/12 [==============================] - 5s 418ms/step - loss: 0.0018 -
val_loss: 3.8990e-04
Epoch 66/100
12/12 [==============================] - 5s 426ms/step - loss: 0.0026 -
val_loss: 1.7920e-04
Epoch 67/100
12/12 [==============================] - 5s 419ms/step - loss: 0.0016 -
val_loss: 2.6580e-04
Epoch 68/100
```

Fig. 9 Model fit Epoch

```
In [34]:  1 train_predict = model.predict(X_train)
          2 test_predict = model.predict(X_test)

23/23 [==============================] - 4s 74ms/step
11/11 [==============================] - 1s 76ms/step
```

Fig. 10 Train / Test Predict

4.5 Ploting graph of Train Predict Data and Test Predict Data :

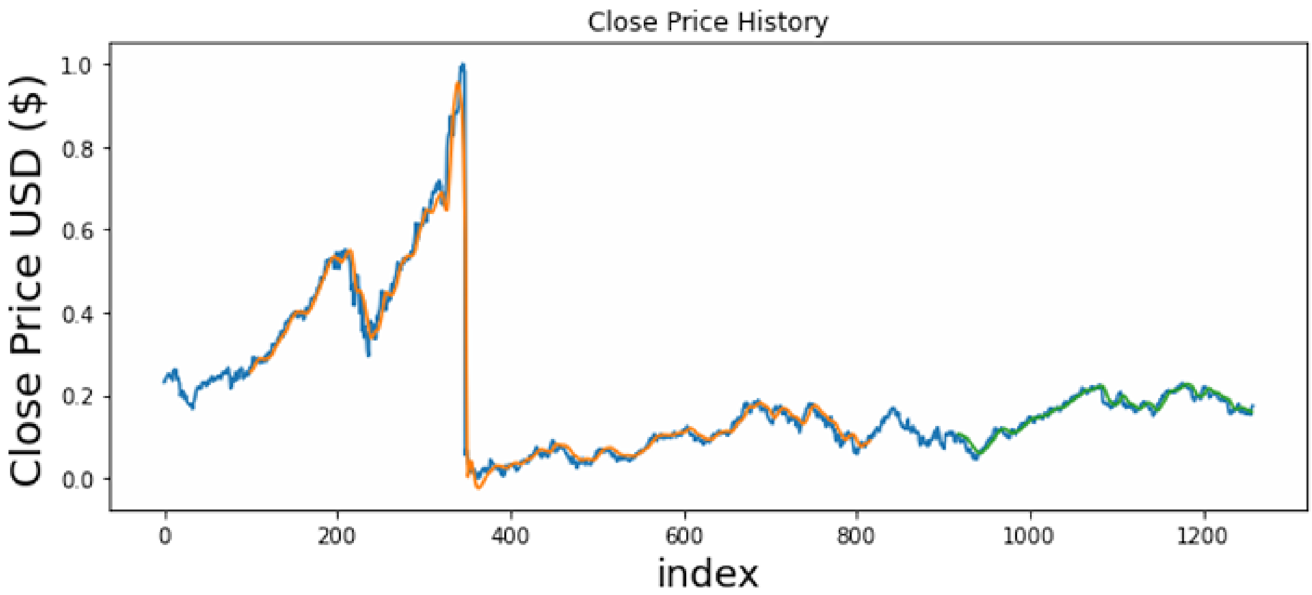

Fig. 11 Train / Test Predict Line Graph

4.5 Accuracy: Accuracy in the context of machine learning refers to the measure of how often a model correctly predicts a target variable. It is one of the most direct metrics used to evaluate the performance of a classification model.

- **R2 Score**: R2 score also known as the coefficient of determination, is a statistical measure used to evaluate the goodness of fit of a regression model. It provides insight into how well the independent variables in the model explain the variation in the dependent variable. The *R*2 score is a value between 0 and 1, where:
  1 indicates a perfect fit, 0 indicates a not perfect fit.

  $R2 = 1-$ (Sum of Squared Residuals / Total sum of squares)

- **mean_squared_error** : The mean squared error (MSE) is a common metric used to evaluate the performance of a regression model. It quantifies the root mean square of the difference between the actual (observed) values and the predicted values produced by the model. A lower MSE means that the model predictions are closer to the true values, which means better accuracy.

- **mean_absolute_error** : The mean absolute error (MAE) is a metric used to evaluate the performance of a regression model. It measures the average absolute difference between the actual (observed) values and the predicted values produced by the model. The MAE is less sensitive to outliers compared to the mean squared error (MSE) because it does not include the squared difference.

- **Square Root** : The sqrt function is used to calculate the square root of a given number. It is a mathematical operation commonly used in various fields, including mathematics, engineering, physics, and programming.

Code:

```
from math import sqrt

from tensorflow.keras.utils import plot_model

# plot_model

from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error

print("Mean Absolute Error:", mean_absolute_error(y_test,test_predict))

print('Mean Squared Error:', mean_squared_error(y_test,test_predict))

print('Root Mean Squared Error:', sqrt(mean_squared_error(y_test,test_predict)))

print("Coefficient of Determination:", r2_score(y_test,test_predict))
```

Output:

```
Mean Absolute Error: 0.00922784698785258
Mean Squared Error: 0.00013649536842484915
Root Mean Squared Error: 0.011683123230748238
Coefficient of Determination: 0.9321359601606991
```

Fig. 12 Accuracy

CHAPTER 5: PREDICTING DATA

5.1 Code to predict Future Data:

```
lst_output = [  ]

n_steps = 100

i = 0

while(i < 30):

        if len(temp_input) > 100:

        X_input = np.array(temp_input[1:])

        print("{} day input {}".format(i, X_input))

        X_input = X_input.reshape(1, n_steps, 1) # Corrected assignmen

        # print(X_input)

        yhat = model.predict(X_input, verbose=0)

        print("{} day output {}".format(i, yhat))

        temp_input.extend(yhat[0].tolist())

        temp_input = temp_input[1:]

        # print(temp_input)

        lst_output.extend(yhat.tolist())

        i += 1

else:

        X_input = X_input.reshape(1, n_steps, 1) # Corrected assignment

        yhat = model.predict(X_input, verbose=0)

        print(yhat[0])

        temp_input.extend(yhat[0].tolist())
```

```
        print(len(temp_input))

        lst_output.extend(yhat.tolist())

        i += 1

print(lst_output)
```

Output:

```
[0.16171956]
101
1 day input [0.21192235 0.20989355 0.2115717  0.2082154  0.20776456 0.2
0929242
 0.20671259 0.20816531 0.21139637 0.20686287 0.21685661 0.21410144
 0.2189856  0.22259236 0.21625548 0.22008766 0.22822793 0.22860363
 0.2272511  0.22304321 0.22567314 0.22038823 0.22001252 0.21730745
 0.21592987 0.21618034 0.21725736 0.21462743 0.19737007 0.19388854
 0.18802755 0.18619912 0.19716969 0.19611772 0.19874765 0.19724483
 0.19807138 0.19233563 0.18995617 0.20485911 0.21219787 0.21803381
 0.22126487 0.21956168 0.21873513 0.21435191 0.21262367 0.20338134
 0.19426425 0.2004258  0.19789606 0.20247965 0.20653726 0.20681277
 0.20408265 0.20541014 0.20115216 0.19586725 0.19363807 0.19291171
 0.18902943 0.18715091 0.18905448 0.1941891  0.18955542 0.18614903
 0.18983093 0.18680025 0.1851221  0.18239198 0.17097057 0.15849718
 0.15599249 0.15569192 0.16002505 0.16508453 0.16628679 0.16102693
 0.16571071 0.16475892 0.16751409 0.17342517 0.17991234 0.16162805
 0.16390733 0.16032561 0.15747026 0.16648716 0.16190357 0.15827176
 0.15529117 0.15731997 0.15524108 0.15714465 0.15431434 0.15737007
```

Fig. 13 Predicted Values

5.2 Ploting Predicted Data:

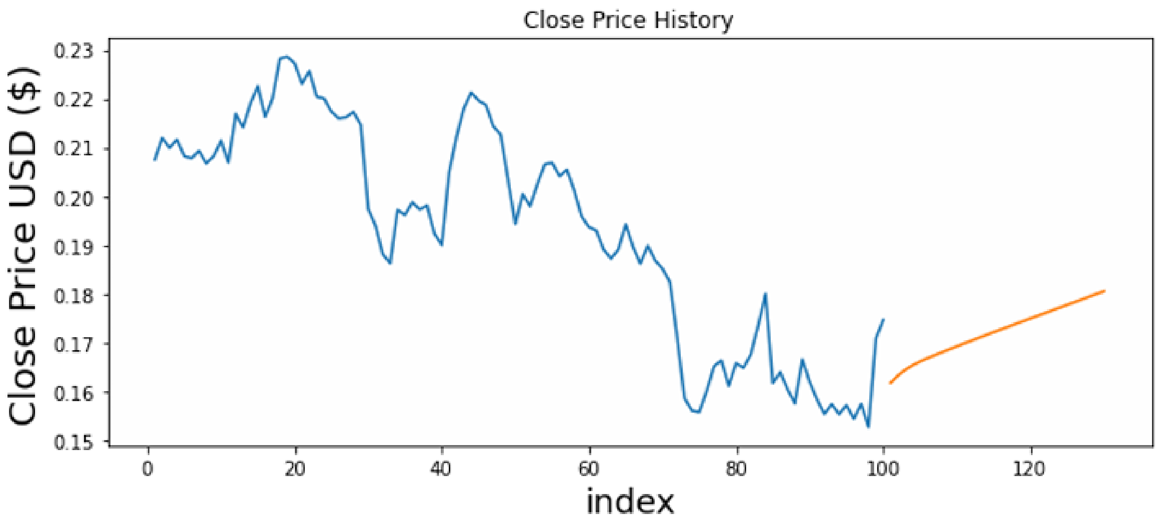

Fig. 14 Predicted Value Line Graph

CHAPTER 6: CONCLUTION

6.1 Application: - The application of stock price prediction using time series data is diverse and covers different areas within the economic market. Here are some of its key applications:

6.1.1 Trading Strategies: Investors and traders today use stock price forecasts to refine trading strategies that include trend following, mean reversion and momentum trading. Predictive models help identify potential stock buying or selling opportunities based on price changes.

6.1.2 Risk Management: Stock price forecasts help in assessing and managing market risk by providing valuable information about potential asset price movements. Risk managers can use these forecasts to reason against critical market conditions.

6.1.3 Algorithmic Trading: Automated trading systems use stock price predictions to execute trades quickly and efficiently, relying on pre-defined rules or algorithms. These systems analyze large data sets and quickly execute trades to seize short-term market opportunities.

6.1.4 Investment Decision: Making: Individual investors and financial analysts use stock price forecasts to make informed investment decisions. Predictive models provide valuable information about future price trends and help investors identify undervalued or overvalued stocks for investment.

6.2 Scope: The scope of stock price prediction using time series data is wide and evolving, with various possibilities for research, development and applications. Here are some key aspects of its scope

6.2.1 Education and Training: The scope extends to education and training programs aimed at equipping professionals and students with the knowledge and skills to effectively develop and apply stock price prediction models. Training programs cover topics such as data analytics, machine learning, and financial modelling.

6.2.2 Industry Applications: Stock price prediction has practical applications in a variety of sectors, including finance, investment banking, hedge funds, and retail trading. The scope includes the development of tailor-made solutions to address specific industrial needs and challenges.

6.2.3 Risk Management: Stock price forecasting helps in assessing and managing market risk by finding risk mitigation strategies such as portfolio diversification, hedging and risk-adjusted portfolio optimization.

6.3 Conclusion

: At the end, when we designed this project, stock price prediction using time series data is a multifaceted and dynamic field with huge potential for research, development and applications. Using advanced machine learning techniques such as LSTM networks, researchers and practitioners can harness the power of historical data to predict future stock prices with increasing accuracy.

The application of predictive models is also in various areas of the financial industry, including business strategies, portfolio management, risk management. These models provide valuable insights from data to the market, helping investors and financial institutions make informed decisions.

The scope of stock price prediction continues to evolve with continuous methodological advances, interdisciplinary research, and industrial applications. With technological progress and the increasing availability of data, new opportunities are emerging for improving predictive modeling techniques and solving real-world challenges in financial markets.

Finally, the project's share price forecasting using time series data contributes to a deeper understanding of market behavior, facilitates more effective investment strategies, and enables stakeholders to navigate the complexities of financial markets with confidence. As it continues to evolve, it holds promise for driving innovation and opening up new opportunities for wealth creation and risk management in the ever-changing landscape of finance.

REFERENCES

[1]. S. Mehtab and J. Sen, "Stock price prediction using CNN and LSTM-based deep learning models", Proceedings of the IEEE International Conference on Decision Aid Sciences and Applications (DASA'20), pp. 447-453, November 8-9, 2020, Sakheer, Bahrain. DOI: 10.1109/DASA51403.2020.9317207

[2]. J. Sen, S. Mehtab, "Stock price prediction using deep learning and natural language processing", In: J. Sen and S. Mehtab (eds), Machine Learning in the Analysis and Forecasting of Financial Time Series, pp. 1-28, Cambridge Scholars Publishing, UK, May 2022. ISBN: 978-1-5275-8324-5.

[3]. G. Ding and L. Qin, "Study on the Prediction of Stock Price Based on the Associated Network Model of LS TM", *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 6, pp. 1307-1317, 2020.

[4]. M. Nikou, G. Mansourfar and J. Bagherzadeh, "Stock Price Prediction Using Deep Learning Algorithm and Its Comparison with Machine Learning Algorithms", Intelligent Systems in Accounting Finance and Management, vol. 26, no. 4, pp. 164-174, 2019.

[5]. S. Mehtab, J. Sen and A. Dutta, "Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models", *arXiv preprint*, 2020

TURNITIN PLAGIARISM REPORT

Please Insert a Scanned Copy of the Front pages duly signed by the Candidate, Supervisor, Departmental Turnitrin Coordinator, and HoD with Seal