DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified)

Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade,

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-078

Mini Project Report

on

"Stock Price Prediction using Machine Learning"

Submitted by

Abhiram G (1DS19CS004)

Adithya N (1DS19CS009)

B Likeith (1DS19CS035)

Datta Shashank C (1DS19CS043)

Sixth Semester B.E (CSE)

Mini Project

19CS6DCMIP

Under the guidance of

Dr. Harish Kumar N
Assistant Professor
Dept. of CSE
DSCE, Bangalore

Department of Computer Science and Engineering

Dayananda Sagar College of Engineering
Bangalore-560078

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

DAYANANDA SAGAR COLLEGE OF ENGINEERING

Shavige Malleshwara Hills, Kumaraswamy Layout, Bangalore - 560078

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled **Stock Price Prediction using Machine Learning** is a bonafide work carried out by **Abhiram G (1DS19CS004), Adithya N (1DS19CS009), B Likeith (1DS19CS035), Datta Shashank C (1DS19CS043)** in partial fulfilment of 6th semester, Bachelor of Engineering in Computer Science and Engineering under Visvesvaraya Technological University, Belgaum during the year 2021-22.

Dr. Harish Kumar N	**Dr. Ramesh Babu**	**Dr.CPS Prakash**
Assistant Professor	Vice principal & Head	Principal,
Department of CSE, DSCE	Department of CSE, DSCE	DSCE
Signature.....	Signature.....	Signature.....

Name of the Examiners:	Signature with date:
1.	.....
2.	.....

ACKNOWLEDGEMENT

We are pleased to have successfully completed the Mini project **"Stock Price Prediction using Machine Learning"**. We thoroughly enjoyed the process of working on this projectand gained a lot of knowledge doing so.

We would like to take this opportunity to express our gratitude to **Dr. C P S Prakash**, Principal of DSCE, for permitting us to utilize all the necessary facilities of the institution.

We also thank our respected Vice Principal, HOD of Computer Science & Engineering, DSCE, Bangalore, **Dr. Ramesh Babu D R**, for his support and encouragement throughout the process.

We are immensely grateful to our respected and learned guide, **Dr. Harish Kumar N,** Assistant Professor CSE, DSCE for his valuable help and guidance. We are indebted to them for their invaluable guidance throughout the process and their useful inputs at all stages of the process.

We also thank all the faculty and support staff of Department of Computer Science, DSCE. Without their support over the years, this work would not have been possible.

Lastly, we would like to express our deep appreciation towards our classmates and our family for providing us with constant moral support and encouragement. They have stood by usin the most difficult of times.

Abhiram G, 1DS19CS004

Adithya N, 1DS19CS009

B Likeith, 1DS19CS035

Datta Shashank C, 1DS19CS043

CONTENTS

ABSTRACT

Time Series forecasting & modelling plays an important role in data analysis. Time series analysis is a specialized branch of statistics used extensively in fields such as Econometrics & Operation Research. Time Series is being widely used in analytics & data science.

Stock prices are volatile in nature and price depends on various factors. The main aim of this project is to predict stock prices using Long short-term memory (LSTM) model.

In this project our task is to predict stock prices to help investors predict stock prices to aid their investments. We use the open, high, low and close prices from the datasets to predict the price of a stock.

The LSTM is a sequential model which contains 4 layers which use ReLU activation function. The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It is given by f(x)=max (0, x). After each layer we included a dropout layer which helps preventing overfitting. The model is then compiled using Adam optimizer, mean squared error as loss function. Then the model is fit using epoch size as 50 and default batch size of 32.

We have built our website using Streamlit Python framework which uses the model as a backend. Streamlit is an open-source app framework for Machine Learning and Data Science teams. The user is able to select a Stock Ticker from a drop down which contains the most commonly traded stocks. The time period for the prediction can also be set by selecting the start and end date. The statistics of the stock in this time period is displayed which contains factors such as count, mean, minimum price, maximum price. We calculate moving average of the first 100 and 200 days. If the moving average of 100 days is more than the moving average of 200 days it indicates an uptrend and if it goes below it's a downtrend. The graph for the same can be viewed on the website.

The prediction of the stocks can be viewed on the website. The price from a certain period of time can also be predicted using the model. The R^2 score, mean square error and mean absolute error is also displayed for the same. The R^2 score generally lies between 0.80 and 0.95. If the stock prices are volatile and changing abruptly it leads to a decrease in accuracy which may cause the R^2 score to go beyond this range.

Chapter 1

INTRODUCTION

The stock market broadly refers to a number of exchanges and other venues in which shares of publicly held companies are bought and sold. Such financial activities are conducted through institutionalized formal exchanges via physical or electronic means and via over-the-counter marketplaces that operate under a defined set of regulations. The principal stock exchanges in India are the National Stock Exchange (NSE) and the Bombay Stock Exchange (BSE). Shares, bonds, Mutual funds and derivatives are traded over the stock exchanges.

Stock Price Prediction using machine learning helps to predict the future value of stocks and other financial assets of companies traded on exchanges. The overall idea of predicting stock prices is to make some profit. A correct prediction of stocks can lead to huge profits for the seller and the broker. Predicting how the stock market will develop is a difficult task. There are other factors involved in prediction, such as physical and psychological factors, rational and irrational behavior. All of these factors combine to make stock prices dynamic and volatile. Therefore, accurate stock price prediction is extremely challenging because of multiple factors, such as politics, global economic conditions, unexpected events, a company's financial performance, and so on.

All of this also means that there's a lot of data to find patterns in. So, financial analysts, researchers, and data scientists keep exploring analytics techniques to detect stock market trends. This gave rise to the concept of algorithmic trading, which uses automated, pre-programmed trading strategies to execute orders.

Machine learning is an efficient way to represent such processes. It predicts a market value close to the tangible value, thereby increasing the accuracy. Introduction of machine learning to the area of stock prediction has appealed to many researches because of its efficient and accurate measurements

The vital part of machine learning is the dataset used. The dataset should be as concrete as possible because a little change in the data can perpetuate massive changes in the outcome. In this project, supervised machine learning is employed on a dataset obtained from Yahoo Finance. This dataset comprises of following five variables: open, close, low, high and volume. Open, close, low and high are different bid prices for the stock at separate times with nearly direct names.

Time series analysis is a specific way of analyzing a sequence of data points collected over an interval of time. In time series analysis, analysts record data points at consistent intervals over a set period of time rather than just recording the data points intermittently or randomly. It is an important tool in many stock market prediction methods, and it makes predictions by analyzing observed points in the series.

The moving average is commonly used with time series to smooth random short-term variations and to highlight other components (trend, season, or cycle) present in your data. The moving average is also known as rolling mean and is calculated by averaging data of the time series within k periods of time. Moving averages are widely used in finance to determine trends in the market and in environmental engineering to evaluate standards for environmental quality such as the concentration of pollutants. The reason for calculating the moving average of a stock is to help smooth out the price data over a specified period of time by creating a constantly updated average price.

We use LSTM model to predict the future price of the stocks from the dataset. LSTMs use a series of gates which control how the information in a sequence of data comes into, is stored in and leaves the network. There are three gates in a typical LSTM; forget gate, input gate and output gate. These gates can be thought of as filters and are each their own neural network.

LSTM networks were designed specifically to overcome the long-term dependency problem faced by recurrent neural networks RNNs. LSTMs have feedback connections which make them different to more traditional feed forward neural networks. This property enables LSTMs to process entire sequences of data without treating each point in the sequence independently, but rather, retaining useful information about previous data in the sequence to help with the processing of new data points. As a result, LSTMs are particularly good at processing sequences of data such as time-series data processing, prediction, and classification.

Chapter 2

LITERATURE SURVEY

➢ For the project, Stock Price Prediction, the following 10 research papers were referred before the implementation process. These research papers are studied and tabulated as follows having the title, algorithm/technique used, as well as their performance, accuracy and the dataset used in the implementation.

Name of the project	Algorithms and Datasets used	Performance	Disadvantages and future enhancements	Metrics and accuracy
Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis – Mojtaba Nabipour, Pooyan Nayyeri, Hamed Jabani, Shahab S, Amir Mosavi	This study compares nine machine learning models (Decision Tree, Random Forest, Adaptive Boosting (Adaboost), extreme Gradient Boosting (XGBoost), Support Vector Classifier (SVC), Naïve Bayes, K-Nearest Neighbors (KNN), Logistic Regression and Artificial Neural Network (ANN)) and two powerful deep learning methods (Recurrent Neural Network (RNN) and Long short-term memory (LSTM).	Nine machine learning models and two deep learning methods (were employed as predictors. We supposed two approaches for input values to models, continuous data and binary data, and we employed three classification metrics for evaluations. Our experimental works showed that there was a significant improvement in the performance of models when they use binary data instead of a continuous one. Indeed,	Overall, it is obvious that all the prediction models perform well when they are trained with continuous values (up to 67%), but the models' performance is remarkably improved when they are trained with binary data (up to 83%).	For Continuous data: RNN – F1 Score ~ 0.85 Accuracy ~ 0.84 LSTM – F1 Score ~ 0.85 Accuracy ~ 0.85 SVC – F1 Score ~ 0.74 Accuracy ~ 0.71 Logistic Regression – F1 Score ~ 0.74 Accuracy ~ 0.71 For Binary data: RNN – F1 Score ~ 0.89 Accuracy ~ 0.89 LSTM –

	Dataset: Four stock market groups, namely diversified financials, petroleum, non-metallic minerals and basic metals, from the Tehran stock exchange.	deep learning algorithms were our superior models in both approaches.		F1 Score ~ 0.89 Accuracy ~ 0.88 SVC – F1 Score ~ 0.86 Accuracy ~ 0.86 Logistic Regression – F1 Score ~ 0.86 Accuracy ~ 0.86
Indian Stock Market Prediction using Deep Learning – Ayan Maiti, Pushparaj Shetty D	Long Short-Term Memory (LSTM) model and the Generative Adversarial Network (GAN) model. Dataset: Yahoo Finance is being actively traded on India's National Stock Exchange (NSE).	LSTM and GAN models to predict the stock prices on the Indian market and compare their performance. LSTM outperformed GAN significantly with an error of 0.074 compared to 0.32 obtained by GAN.	The model is not able to predict the direction of the price movement. The increase in the number of look-back days did not have a significant improvement in results on the LSTM model.	The results of the model update cycles on the predictive performance show that M=1200 and N=300 produce the best results of RMSRE=0.073884, obtained at look-back of 90 days.
Stock Market Prediction Web Service Using Deep Learning by LSTM –	Deep learning Long Short-Term Memory (LSTM) model Dataset:	The web-based application demonstrates machine learning outperforms statistical and	To add different parameters to our model for better prediction	Test results have given 70% accurate prediction stats.

Mohammad Mahabubul Hasan, Pritom Roy, Sabbir Sarkar and Mohammad Monirujjaman Khan	DSE Bangladesh	regression techniques in forecasting share prices.	results and add new features to our web service to make this system much more robust and sustainable.	
Deep learning for stock prediction using numerical and textual Information -Ryo Akita, Akira Yoshihara, Takashi Matsubara, Kuniaki Uehara	Long Short-Term Memory (LSTM) model. Representation of textual information by employing the technique of Paragraph Vector. Representation of numerical information using regression. Used MeCab as the morphological Analyser Dataset. We used the morning edition of the Nikkei newspaper published from 2001 to 2008 for our experiments, with the news from year 2001 to 2006 as the training data, 2007 as validation data, and 2008 as	Experimental results showed that distributed representations of textual information are better than the numerical-data-only methods and Bag-of-Words based methods, LSTM was capable of capturing time series influence of input data than other models, and considering the companies in the same industry was effective for stock price prediction.	For future work, we would like to incorporate more technical indices such as the moving average (MA) and the moving average convergence divergence (MACD) for better profit-making capabilities.	This approach was able to make the more profits from all industries compared to considering a separated single company. Hence, it is effective for predicting stock prices to consider multiple companies together.

	test data. The target 10 companies were chosen from Nikkei 225 and the same industries			
Stock Market Prediction Using Machine Learning – Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena	Regression-Based Model Long Short-Term Memory (LSTM) Network-Based Model Dataset: Yahoo Finance (Five variables: open, close, low, high and volume)	A. Regression-Based Model: Regression based Model is used for predicting continuous values through some given independent values. Regression uses a given linear function for predicting continuous values B. Long Short-Term Memory (LSTM) Network-Based Model: LSTM is the advanced version of Recurrent-Neural Networks (RNN) where the information belonging to the previous state persists. LSTM regulates error by giving aid to the RNNs	In the future, the accuracy of the stock market prediction system can be further improved by utilizing a much bigger dataset than the one being utilized currently. Furthermore, other emerging models of Machine Learning could also be studied to check for the accuracy rate resulting from them.	Regression: The R-square confidence test resulted in a confidence score of 0.86625 LSTM: The model resulted in a Train Score of 0.00106 MSE (0.03 RMSE) and a Test Score of 0.00875 MSE (0.09 RMSE).

		by retaining information for older stages making the prediction more accurate. Thus proving itself as much more reliable compared to other methods		
Stock price prediction using machine learning techniques - Sumeet Sarode, Harsha G. Tolani, Prateek Kak, Lifna C S	LSTM (Long Short-Term Memory) is used for predicting Dataset: Real-time news that is collected from various websites providing financial news.	The system combines price prediction based on historical and real-time data along with news analysis. It takes the latest trading information and analysis indicators as its input. For news analysis, only the relevant and live news is collected from a large set of business news. The filtered news is analysed to predict sentiment around companies. The results of both analyses are integrated together to get a response that gives a recommendation for future increases.	The future quant funds will obviate risks that are seized by unforeseen news events and make it more pliable and sturdy.	
Machine	Partial Least	Explores the	The model	For Partial Least

Learning Approaches in Stock Price Prediction: A Systematic Review - Payal Soni, Yogya Tewari and Prof. Deepa Krishnan	Squares Classifier (PLS Classifier) LSTM coupled with sentiment analysis ARIMA Dataset: Yahoo and NSE-India	different techniques that are used in the prediction of share prices from traditional machine learning and deep learning methods to neural networks and graph-based approaches. The best performing model out of the ones created in is the GCN based graph, modelled from news co-mentions. A reason for this could be that the graph is causation based instead of being correlation-based.	considered 12 technical indicators to identify patterns in the stock market.However, the accuracy level lies between 50-70%, thus to increase the level of accuracy, a higher number of technical indicators can be used. Focus on combining the sentiment analysis of stocks related information and the numeric value associated with the historical value of stocks in predicting stock prices can be done.	Squares Classifier (PLS Classifier) - Average Error Value = 0.81225 LSTM coupled with sentiment analysis - Matthews Correlation Coefficient (MCC) = 0.04092 Accuracy = 52.27% ARIMA - For 3 time steps ahead: RMSE: ~15% MAPE: 20-25% MAE: ≤15% For 9 time steps ahead: RMSE: 15-20% MAPE: 15-20% MAE: 10-15%
A Prediction Approach for Stock Market Volatility Based on Time Series Data. – Sheikh Mohammad Idrees, M. Afshar Alam, Parul Agarwal	ARIMA Model. (Autoregressive integrated moving average) The publicly available time series data of Indian stock market has been used for this study.	Auto-regressive processes have a certain degree of unpredictability or randomness built in, that occasionally makes it capable to predict future trends pretty well. ARIMA approach is good enough for handling time	The predicted time series has been compared with the actual time series, which shows roughly a deviation of 5% mean percentage error for both Nifty and Sensex on average.	For Nifty: p-value = 0.9099 For Sensex: p-value = 0.8682.

		series data, and as such can be very constructive in various real world problems like that of health sector, education, finance and other practical domains for prediction.		
Stock Closing Price Prediction using Machine Learning Techniques - Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar	Linear Regression, Random Walk Theory (RWT), Moving Average Convergence / Divergence (MACD) and also using some linear models like Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), for predicting stock prices. The dataset includes 10 year data from 4/5/2009 to 4/5/2019 of Nike, Goldman Sachs, Johnson and Johnson, Pfizer and JP Morgan Chase	The comparative analysis based on RMSE, MAPE and MBE values clearly indicate that ANN gives better prediction of stock prices as compared to RF. Results show that the best values obtained by ANN model gives RMSE (0.42), MAPE (0.77) and MBE (0.013).	The historical dataset available on company's website consists of only few features like high, low, open, close, adjacent close value of stock prices, volume of shares traded etc., which are not sufficient enough. For future work, deep learning models could be developed which consider financial news articles along with financial parameters such as a closing price, traded volume, profit and loss statements etc., for possibly	For ANN, RMSE ~ 1.10 MAPE ~ 1.07% MBE ~ -0.0522 For RN, RMSE ~ 1.29 MAPE ~ 1.14% MBE ~ -0.0521

	and Co.		better results.	
STOCK PRICE PREDICTION USING LSTM, RNN AND CNN-SLIDING WINDOW MODEL -Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P	We have used three different deep learning architectures, RNN, LSTM and CNN for this work. The data set consists of minute wise stock price for 1721 NSE listed companies for the period of July 2014 to June 2015. We trained the model using the data of Infosys and was able to predict stock price of Infosys ,TCS and Cipla.	For comparison we have used ARIMA, which is a linear model used for forecasting. From error percentage obtained it is clear that deep learning models are outperforming ARIMA.	The changes occurring in the stock market may not always be in a regular pattern or may not always follow the same cycle. CNN architecture is capable of identifying the changes in trends. For the proposed methodology CNN is identified as the best model compared to the other two models.	ERROR PERCENTAGE RNN – 5.12 LSTM – 5.31 CNN – 4.98

Chapter 3

METHODOLOGY

We use the dataset picked from Yahoo Finance. The dataset consisted of approximately 9 lakh records of the required stock prices and other relevant values. The data reflected the stock prices at certain time intervals for each day of the year. It consisted of various sections namely date, symbol, open, close, low, high and volume. We use pandas data reader to scrape the data from the website. Following this normalization of the data was performed through usage of the sklearn library in Python and the data was divided into training and testing sets. The test set was kept as 30% of the available dataset.

LSTM is the advanced version of Recurrent-Neural-Networks (RNN) where the information belonging to previous state persists. These are different from RNNs as they involve long term dependencies and RNNs works on finding the relationship between the recent and the current information. This indicates that the interval of information is relatively smaller than that to LSTM.

The main purpose behind using this model in stock market prediction is that the predictions depend on large amounts of data and are generally dependent on the long-term history of the market. So, LSTM regulates error by giving an aid to the RNNs through retaining information for older stages making the prediction more accurate. Thus, proving itself as much more reliable compared to other methods.

Since stock market involves processing of huge data, the gradients with respect to the weight matrix may become very small and may degrade the learning rate. This corresponds to the problem of Vanishing Gradient. LSTM prevents this from happening. The LSTM consists of a remembering cell, input gate, output gate and a forget gate. The cell remembers the value for long term propagation and the gates regulate them.

In this model, a sequential model has been made which involves stacking four LSTM layers on top of each other with increasing units of 50, 60, 80 and 120 respectively. Each layer is followed by a dropout layer which drops out 0.2, 0.3, 0.4 and 0.5 units respectively after each LSTM layer. The dropout layer prevents overfitting of the model. At last, the core dense layer where each neuron is connected to every other in the next layer gives output as 1. The model is compiled with Adam optimizer and a mean square cost function to maintain the error throughout the process and mean absolute error is chosen as a metric for the prediction.

SYSTEM DESIGN

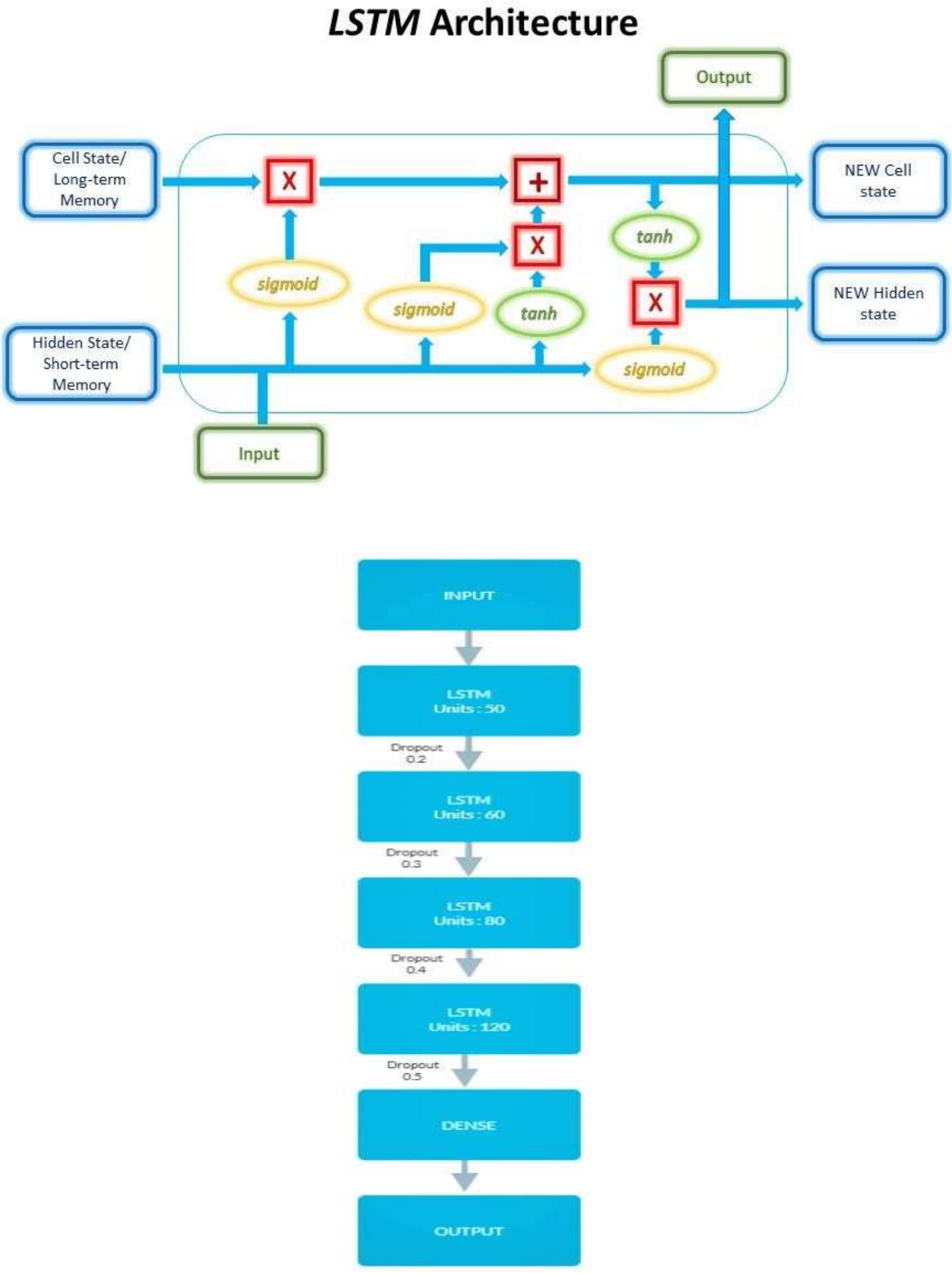

Fig. 3.1 Design of the LSTM model

Fig. 3.1 shows the LSTM model being used for the prediction of stock prices. It is a sequential model to which four LSTM layers are added with increasing units of 50, 60, 80 and 120 respectively. ReLU activation function is used in each layer which stands for Rectified Linear Unit. The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It is given by f(x)=max (0, x). It is the most commonly used activation function in deep learning models. Each LSTM layer is followed by a dropout layer which drops out 0.2, 0.3, 0.4 and 0.5 units respectively after each LSTM layer. The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. Finally, the core dense layer is used as the last layer. A dense layer also referred to as a fully connected layer is a layer that is used in the final stages of the neural network. This layer helps in changing the dimensionality of the output from the preceding layer so that the model can easily define the relationship between the values of the data in which the model is working. The model is compiled with Adam optimizer and a mean square cost function to calculate the error and mean absolute error is chosen as a metric for the prediction.

Chapter 4

SNAPSHOTS AND RESULTS

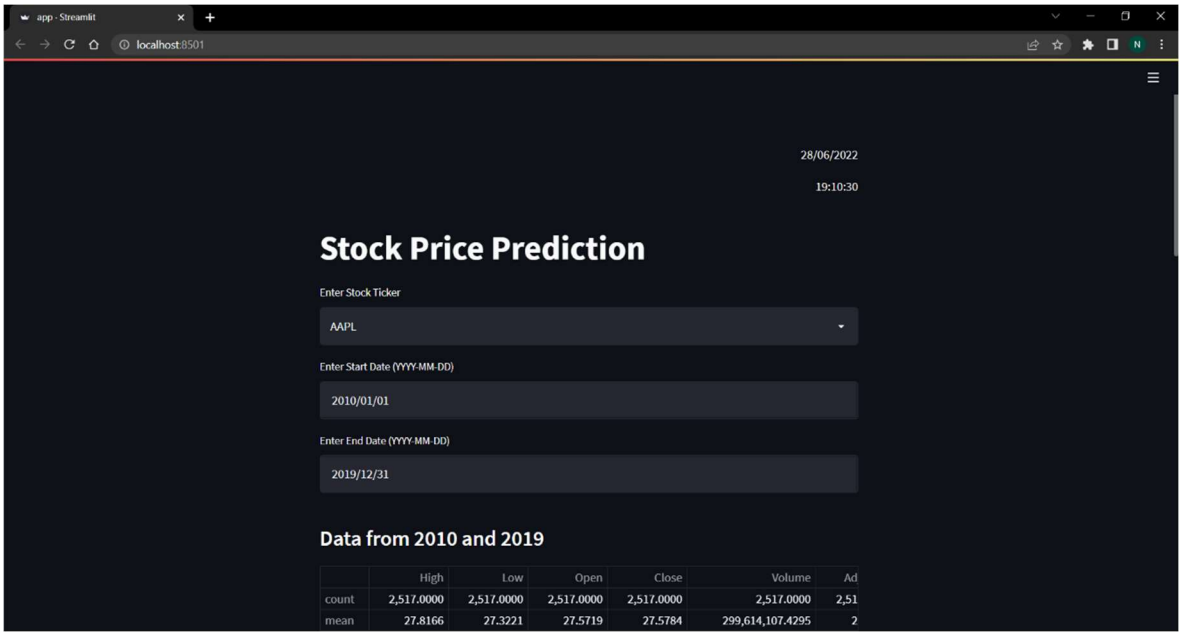

Fig. 4.1 User Input for Stock Ticker and Date

Fig. 4.2 Data Summary and Moving Average graph

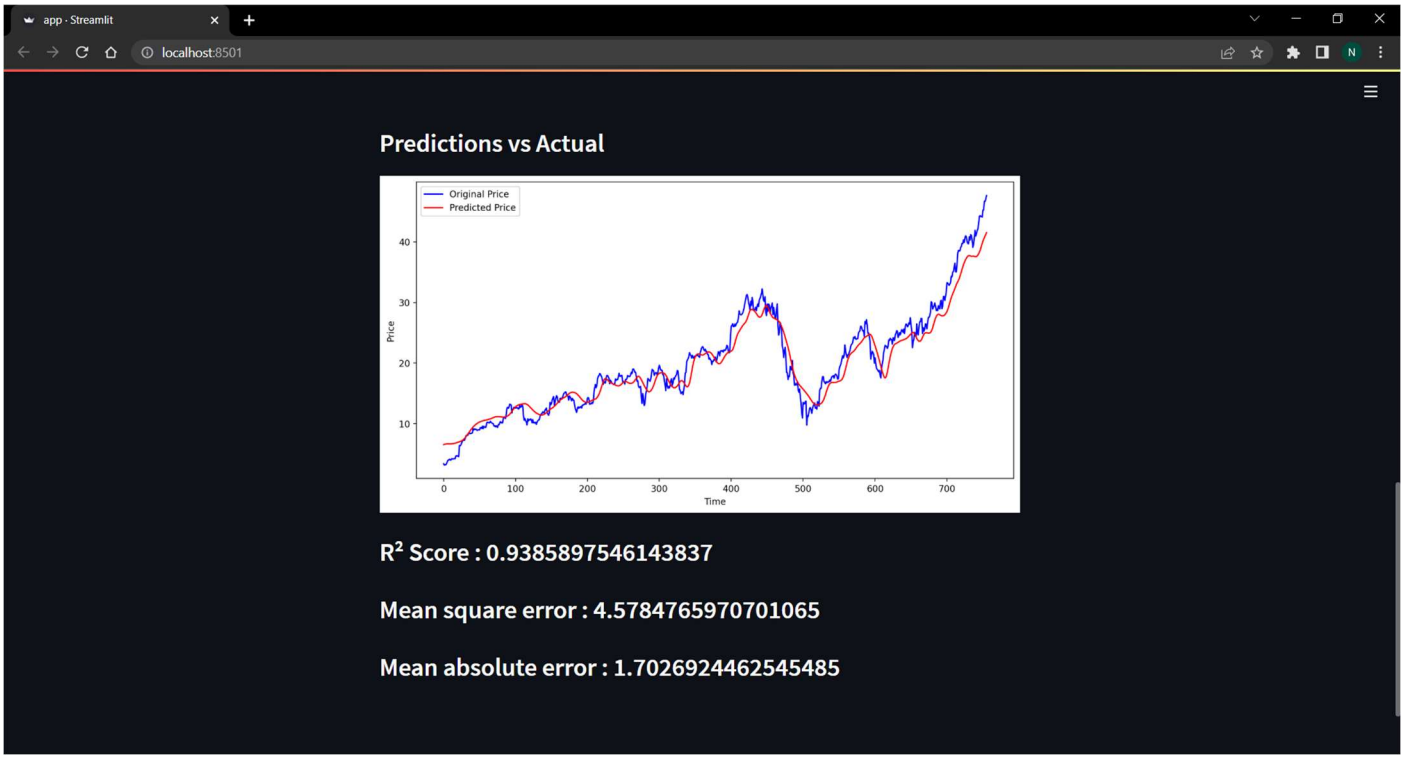

Fig. 4.3 Predictions vs Original Price Graph & Accuracy and Loss values

Chapter 5

CONCLUSION

We propose an LSTM based model for stock price prediction. It is seen that deep neural network architectures are capable of capturing hidden dynamics and are able to make predictions.

This project shows the predictions made by machine learning models to retail investors through a web application. It helps investors navigate through the stock markets with additional analysis and help them make more informed decisions. The findings concluded the usefulness of evolution algorithm in lowering the mean squared error when predicting stock prices, which is helpful for improving the trend prediction for retail investors.

Therefore, with the application and research findings, to large extent the project team achieved the aim of creating an user-friendly system for retail investors whom does not have previous technical knowledge to navigate the machine model predictions result with useful benchmarks.

FUTURE ENHANCEMENTS

We can improve the efficiency of the model by incorporating the loss function for direction accuracy used in the model. Also, new loss function can be designed to achieve better results. We have used stock price data of daily frequency. Higher frequency data of every minute stock price can be obtained to train and test the performance of the models implemented in this project. We use only the close price as the predicting variable. The model can also be enhanced by using other variables such as open and adjusted close.

The stock application that you are looking at should be compatible with the present smartphone OS (Be it Android or iPhone).

The stock analysis app can provide you the option to create multiple watchlists with as many stocks as possible in a single list. This way one can track the number of stocks from a single window. The watch list should accompany tools like stock screeners and real-time alerts to make informed decisions & strategies.

The app can have the historical charts, time-interval, intraday charts for periods ranging from a few minutes to 5 years. The presence of indicators, overlays and drawing tools helps in performing technical studies.

REFERENCES

1. "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis" – Mojtaba Nabipour, Pooyan Nayyeri, Hamed Jabani, Shahab S, Amir Mosavi
2. "Indian Stock Market Prediction using Deep Learning" – Ayan Maiti, Pushparaj Shetty D
3. "Stock Market Prediction Web Service Using Deep Learning by LSTM" – Mohammad Mahabubul Hasan, Pritom Roy, Sabbir Sarkar and Mohammad Monirujjaman Khan
4. "Stock Market Prediction Using Machine Learning" – Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena
5. "Deep learning for stock prediction using numerical and textual Information" - Ryo Akita, Akira Yoshihara, Takashi Matsubara, Kuniaki Uehara
6. "Stock price prediction using machine learning techniques" -Sumeet Sarode, Harsha G. Tolani, Prateek Kak, Lifna C S
7. "Machine Learning Approaches in Stock Price Prediction: A Systematic Review" -Payal Soni, Yogya Tewari and Prof.Deepa Krishnan
8. "A Prediction Approach for Stock Market Volatility Based on Time Series Data." –Sheikh Mohammad Idrees, M. Afshar Alam, Parul Agarwal
9. "Stock Closing Price Prediction using Machine Learning Techniques" - Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar
10. "Stock Price Prediction Using Lstm, Rnn And Cnn Sliding Window Model" - Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P
11. https://keras.io/api/layers/recurrent_layers/lstm/
12. https://towardsdatascience.com/a-practical-guide-to-rnn-and-lstm-in-keras-980f176271bc

APPENDIX

Code:

I. Streamlit frontend:

1. Importing Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas_datareader as data
%matplotlib notebook
from sklearn.preprocessing import MinMaxScaler
from keras.models import load_model
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
import streamlit as st
import datetime
with open('styles.css') as f:
  st.markdown(f'<style>{f.read()}</style>', unsafe_allow_html=True)
```

2. User input and loading dataset:

```
today=datetime.date.today()
now=datetime.datetime.now()
st.write(today.strftime("%d/%m/%Y"))
st.write(now.strftime("%H:%M:%S"))
st.title("Stock Price Prediction")
user_input = st.selectbox('Enter Stock
Ticker',('AAPL','TSLA','MSFT','GOOG','GOOGL','AMZN','HDB','WIT','INFY','MMYT','AZRE','TTM'))

start = st.date_input('Enter Start Date (YYYY-MM-DD)',datetime.date(2010,1,1))
end = st.date_input('Enter End Date (YYYY-MM-DD)',datetime.date(2019,12,31))
df = data.DataReader(user_input,data_source='yahoo', start=start, end=end)
columns = st.columns((1,1))
```

3. Describing Data:

```
st.subheader('Data from '+str(start.year)+' and '+str(end.year))
```

```
st.write(df.describe())
```

4. Visualisations:

```
st.subheader("Closing Price vs Time chart with 100MA & 200MA")
ma100 = df.Close.rolling(100).mean()
ma200 = df.Close.rolling(200).mean()
fig = plt.figure(figsize=(12,6))
plt.plot(ma100,'r',label='Moving Average for 100 days')
plt.plot(ma200,'g',label='Moving Average for 200 days')
plt.plot(df.Close,'b',label='Close Price')
plt.xlabel('Years')
plt.ylabel('Price')
plt.legend()
st.pyplot(fig)
```

5. Data Split and scaling:

```
data_train = pd.DataFrame(df['Close'][0:int(len(df)*0.70)])
data_test = pd.DataFrame(df['Close'][int(len(df)*0.70):int(len(df))])
print(data_train.shape)
print(data_test.shape)
scaler = MinMaxScaler(feature_range=(0,1))
data_training_array = scaler.fit_transform(data_train)
```

6. Loading the model:

```
model = load_model('keras_model.h5')
past_100_days = data_train.tail(100)
final_df = past_100_days.append(data_test,ignore_index=True)
input_data = scaler.fit_transform(final_df)
x_test = []
y_test = []
for i in range(100,input_data.shape[0]):
  x_test.append(input_data[i-100:i])
  y_test.append(input_data[i,0])
x_test, y_test = np.array(x_test),np.array(y_test)
```

7. Making Predictions:

```
y_predicted = model.predict(x_test)
scaler = scaler.scale_
scale_factor = 1/scaler[0]
y_predicted = y_predicted*scale_factor
y_test = y_test*scale_factor
```

8. Plotting the prediction:

```
st.subheader("Predictions vs Actual")
fig2=plt.figure(figsize=(12,6))
plt.plot(y_test,'b',label='Original Price')
plt.plot(y_predicted,'r',label='Predicted Price')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
st.pyplot(fig2)
```

9.Accuracy:

```
mse = mean_squared_error(y_test,y_predicted)
r2 = r2_score(y_test,y_predicted)
mae = mean_absolute_error(y_test,y_predicted)
st.subheader("R\u00b2 Score : "+str(r2))
st.subheader("Mean square error : "+str(mse))
st.subheader("Mean absolute error : "+str(mae))
```

II. LSTM MODEL:

1.Importing Libraries:

```
from keras.layers import Dense,Dropout,LSTM
from keras.models import Sequential
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
```

2.Adding layers to the model:

```
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))
model.summary()
```

3.Compiling the model:

```
model.compile(optimizer='adam',loss='mean_squared_error',metrics=['mae'])
history=model.fit(x_train,y_train,epochs=50)
```

4.Preparing data to test:

```
past_100_days = data_train.tail(100)
final_df = past_100_days.append(data_test,ignore_index=True)
input_data = scaler.fit_transform(final_df)
x_test = []
y_test = []
for i in range(100,input_data.shape[0]):
  x_test.append(input_data[i-100:i])
  y_test.append(input_data[i,0])
```

```
x_test, y_test = np.array(x_test),np.array(y_test)
```

5. Making predictions:

```
y_predicted = model.predict(x_test)
scale_factor = 1/0.02099517
y_predicted = y_predicted*scale_factor
y_test = y_test*scale_factor
```

6. Calculating Accuracy and Loss:

```
mae = history.history['mae'][-1]
mse=history.history['loss'][-1]
mse = mean_squared_error(y_test,y_predicted)
r2 = r2_score(y_test,y_predicted)
mae = mean_absolute_error(y_test,y_predicted)
```

PROJECT SOURCE CODE LINK:

https://github.com/adithya-n11/Stock-Prediction