AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

FACULTY OF SCIENCE AND TECHNOLOGY

Stock Market Prediction Using Long Short Term Memory (LSTM)

A Thesis Presented to the

DEPARTMENT OF COMPUTER SCIENCE

In Partial Fulfillment of the Requirements for the Degree BSc. in CSE

Supervised By

Md. Tohedul Islam

Assistant Professor

Department of Computer Science

Faculty of Science and Technology

Submitted By

17-34196-1	MOHD ABU NADIF
17-34181-1	MD. TOWHIDUR RAHMAN SAMIN
17-34600-2	ABDULLAH AL MAMUN
17-34639-2	CHOYON DAS PULOCK

Declaration

We announce that this thesis is our unique work and has not been submitted in any shape for another degree or diploma at any college or other organized tertiary instruction. Data determined from the distributed and unpublished work of others has been recognized within the content and a list of references is given.

MOHD ABU NADIF
17-34196-1
`BSc. CSE

ABDULLAH AL MAMUN
17-34600-2
BSc. CSE

MD. TOWHIDUR RAHMAN SAMIN
17-34181-1
BSc. CSE

CHOYON DAS PULOCK
17-34639-2
BSc. CSE

Approval

The thesis titled "Stock Market Prediction Using LSTM" has been submitted to the taking after regarded individuals of the board of inspectors of the faculty of Computer Science in fractional fulfillment of the necessities for the degree of Bachelor of Science in Computer Science & Engineering on April 2021 by Mohd Abu Nadif (17-34196-1), Abdullah Al Mamun (17-34600-2), Md. Towhidur Rahman Samin (17-34181-1), Choyon Das Pulock (17-34639-2) has been accepted as satisfactory.

Md. Tohedul Islam
Assistant Professor & Supervisor
Department of Computer Science
American International University-Bangladesh

Dr. Dip Nandi
Associate Professor & Director
Department of Computer Science
American International University-Bangladesh

Dr. Md Mahbub Chowdhury Mishu
Assistant Professor and Department Head
Department of Computer Science
American International University-Bangladesh

Professor Dr. Tafazzal Hossain
Dean
Faculty of Science & Information Technology
American International University-Bangladesh

DR. Carmen Lamagna
Vice Chancellor
American International University-Bangladesh

Acknowledgment

To begin with and preeminent, we would like to much oblige the all-powerful Allah for the great wellbeing and favoring required to wrap up this book. We would like to precise our profound and earnest appreciation to our respectable research supervisor, Md. Tohedul Islam, Assistant Professor, Department of CSE, American International University- Bangladesh (AIUB) for his extraordinary back and direction all through the complete work. His dynamism, vision, earnestness, and inspiration offer assistance to us to go through the correct track It was an incredible benefit and respect to work and consider under his direction. We want to thank our external Dr. Dip Nondi Associate Professor & Director, Faculty of science and technology, American International University-Bangladesh (AIUB) for giving us her valuable time. With many thanks to Dr. Carmen Z. Lamagna, honorable Vice Chancellor, American International University- Bangladesh (AIUB) especially for her encouragement. We would like to thank all of our companions and relatives for their love, prayers, caring, and penances for teaching and planning us for the longer term.

Abstract

The stock market is one of the most unpredictable and highly concerned places in the world. There is no fundamental way to forecast stock market share prices. So people think stock market prediction is a gamble. Nevertheless, it is possible to generate a constructive pattern by using different types of algorithms and predict the share price. But when the attributes are complex and the majority of these classification algorithms are linear they perform poorly in predicting class labels. In this paper, we suggest a non-linear technique based on the Long Short-Term Memory (LSTM) architecture. According to studies, LSTM-based models predict the time and sequential models better than other models and RNN is the first algorithm with an internal memory that remembers its input, making it ideal for machine learning problems involving sequential data. For our experiment, we collected the share market data from a particular company named Beximco for the last 11 years. To reassert the effectiveness of the system different test data are used. This work introduces a robust method that can predict stock price accurately based on LSTM.

A. Table of Contents

List of Figures

Chapter 1

1.Introduction

The stock market is a vital organ for the economy in many emerging countries like Bangladesh. Many companies raise their capital by selling shares to the people. So like other countries, our economic growth is closely related to the stock market. Predicting stock market share price is a mysterious and tough task. The prediction can make a huge impact on both the good and bad sides. Most of the emerging countries rely on their stock market for strengthening their economy.

The stock exchange is known for its limited intricacy and unpredictability, and individuals are continually searching for an exact and powerful approach to manage stock exchanging. There are many works done to predict the stock market. Mostly there are four prediction technique types:- 1)Technical Analysis Approach 2)Fundamental Analysis Approach 3)Time-Series Prediction and 4) Machine Learning algorithm methods. Long Short-Term Memory (LSTM) neural networks are created by a recurrent neural network (RNN) and have critical application esteem in numerous fields.[1]. A wavelet change is utilized to denies authentic stock information, concentrate and train its highlights, and build up the forecast model of a stock price based on Long Short-Term Memory(LSTM). The neural network in profound learning has become a mainstream indicator because of its great nonlinear estimation capacity and versatile self-learning. Long Short-Term Memory(LSTM) neural networks have performed well in speech recognition and text processing.[2] At a similar time, since they have the attributes of selectivity, memory cells, LSTM neural networks are reasonable for arbitrary nonstationary arrangements, for example, stock-value time series.

LSTM networks are a perennial neural network which will rely upon the order in sequence prediction. This is a behavior needed in complicated downside domains like AI, speech recognition, and more for predicting or analyzing different types of data. LSTMs area unit a fancy space of deep learning. It will be arduous to induce your hands around what LSTMs area unit and the way terms like biface and sequence-to-sequence relate to the sphere. There are several recent studies on the applying of LSTM neural networks to the exchange. A hybrid model of generalized autoregressive conditional heteroscedasticity (GARCH) combined with LSTM was projected to predict stock value fluctuations[3].CNN was accustomed develop a quantitative stock choice strategy to work out stock trends then predict stock costs exploitation LSTM to push a hybrid neural network model for quantitative temporal order methods to extend profits. In recent times many machine learning algorithms, along with acritical neural networks, gradient enhanced reversion trees, vector support and, random forecasts, have been optimized in a mix of statistics and learning models. In recent times many machine learning algorithms, along with acritical neural networks, gradient enhanced reversion trees, vector support and, random forecasts, have been optimized in a mix of statistics and learning models.

1.1. Problem Statement

Now the days stock market has become the heart of the global financial system. To raise any capital or increase the GDP value stock market plays an important role. Almost in most countries, any businessmen or individuals and other investors access the markets to buy and sell stocks of these businesses. Predicting a stock market result is to determine the future financial value of a company or a stock so that the future price could yield a significant profit. A company's stock price always reflects any investors' ability to earn its profit in the future. If the company is doing well the investors will be happy and they will remain with the company and receive increases in compensation. The value of stocks is continuously changing. Today if the value is high tomorrow it can below. The patterns are always unpredictable so the investors need to always wise to invest in any stock. That's why they will always need the updated value of any stocks. Considering this situation, we are working on an analysis technique to recognize the value and find a pattern so that the investors can predict the upcoming stock value and make a decision before investing in a stock. Long-term short memory (LSTM) networks are a perennial neural network which can depend on the order in sequence prediction. this is often a behavior required in sophisticated drawback domains like AI, speech recognition, and additional for predicting or analyzing different types of information. Without taking a wise decision it is always difficult for investors to find the right time to buy or sell any stocks. Any bad decision an investor can lose a lot of money. Investors always want profits from their stocks, without knowing the right moment when the stock is high priced investors can miss out on a lot of profit. Using this method from previous stock exchange data predicted results will help the investors to know about the upcoming consequences so that investors can know the predicted results and invest their money in the right stocks and make great profits.

1.2. Research Objectives

Our main objectives of this research are to identify the hidden pattern in the historical stock market data and predict the future scope and also to correct the prediction of stock so that this research can people to get a recommendation of decision making for selling and buying stock and get more profits.

1.3. Research Questions

1. Can we predict the Stock Market data using LSTM?

Chapter 2

2. Literature Review

In this section number of research, surveys and experimental methods will be reviewed based on the stock market prediction processing techniques applying different methodologies and theorems. The review will mainly be focusing on deep learning approaches which are being used for predicting the upcoming result of stock.

Archana Gupta , Pranay Bhatia , Kashyap Dave & Pritesh Jain [4] created a data mining model [KNN] which is applied to the stock information for a time of 5 years to anticipate the stock cost and contrast it and the first developments for a time of 5 years. Incidentally, the proficiency of the test information is around 65-70% if the information isn't to a great extent slanted else the accuracy is around 48 – 53%.

V.Sandhiya & T.Revathi [5] have discussed big data and its characteristics, data mining techniques parameters to analyze the data. To store the huge amount of big data the HDFS was used. To process the enormous data the Map-reduce algorithm is used. This processes the records in a parallel manner. The major advantage of forecasting is they can easily predict the stock exchanges for predicting the future trends so that investors may know about the market to invest their money on profitable trades.

Kulshrestha, Tanisha [6] proposed model distinguishes the example covered up in the chronicled Stock Market information and predicts the future degree. Because of the vulnerabilities in the information stock information, the exactness ingot in this model is almost 80%. The precision of the stock value expectation is influenced by different variables. For the most part, the stock costs are influenced by news channels, supply interest, recorded information, and so on. This model can be improved by considering several other factors such as news which can be categorized using sentiment analysis to enhance the accuracy of the prediction. It can also use Generative Adversarial Networks (GAN) along with Long Short Term Memory (LSTM), a Recurrent Neural Network (RNN) to build the prediction model and use Convolutional Neural Network (CNN) as a discriminator.

Qiu J, Wang B & Zhou C [7] This paper builds up an anticipating system to foresee the initial costs of stocks. They prepared stock information through a wavelet change and utilized a consideration-based LSTM neural organization to anticipate the stock opening cost, with magnificent outcomes. The test results show that contrasted with the generally utilized LSTM, GRU, and LSTM neural organization models with wavelet change, their proposed model has a superior fitting degree and improved precision of the expected results. Subsequently, the model has wide application prospects and is profoundly serious with existing models.

Weng, Bin & Ahmed, Mohamed & Megahed [8] In this research they speculate that joining dissimilar online information sources with conventional time-arrangement and specialized markers for a stock can give a more viable and clever every day exchanging master framework. Three machine learning models, decision trees, neural networks, and support vector machines, serve as the basis for our “inference engine”. To evaluate the performance of their expert system, they presented a case study based on the AAPL (Apple NASDAQ) stock. Their system had an 85% precision in foreseeing the following day AAPL stock development, which outflanks the revealed rates in the writing. Their outcomes propose that: (a) the information base of monetary master frameworks can profit from information caught from nontraditional "specialists" like Google and Wikipedia; (b) broadening the information base by joining information from unique sources can help improve the presentation of monetary expert systems, and (c) the utilization of straightforward AI models for deduction and rule age is proper with their rich information data set.

AnitaYadava, KJhaa & AditiSharanb [9] In this experiment stateful and stateless LSTMs were looked at for four changed organizations. They determined the contrasts among stateful and stateless LSTM at the stock cost expectation issue picked for the test, are genuinely immaterial. A large portion of the distinction in qualities can be represented by the arbitrary cultivating that happens for each LSTM run that produces little varieties in yield. This recommends that stateless LSTM is more steady contrasted with stateful LSTM. The quantity of covered-up layers was shifted from one to seven. The outcomes show that n = 1 has all the earmarks of being the best arrangement to the extent mean RMSE is concerned. A benefit of expanding the quantity of covered-up layers is that the LSTM turns out to be more steady as uncovered by the diminishing standard deviation esteems and the spread in the case and bristle plot graph.

Maksuda Akter Rubi & Md Kamrul Hossain[10] In this research, Artificial Neural Network (ANN) is the most recent device which has been used to foresee the Dhaka Stock Exchange Broad Index (DSEX) for the span from August 2013 to December 2018. DSEX addresses around 97% of the all-out value Market Capitalization. In this paper, Three Layer Feed Forward Neural Network utilizing Back engendering learning calculation is utilized. The precision of the model is analyzed by figuring Root Mean Square Error (RMSE) values. The fitted ANN model showed a better expectation design with more modest blunder esteems.

Desai, N., & Gandhi, N [11] In this research In Data Mining to anticipate securities exchange here we have made NLP based module and factual boundary-based module which results in the sentence extremity and conduct contrasted with last year data. By utilizing this method we get precise and dependable expectation results which give customers better answer for where to contribute their significant cash. These modules assess the news sentences dependent on linguistic examination and with the assistance of recorded information too.

A. S. Al Rafi, T. Rahman, A. R. Al Abir, T. A. Rajib, M. Islam, and M. S. Hossain Mukta [12] In this paper, a new classification technique was proposed by significantly changing the LSTM algorithm. They have split the dataset into two distinct parts: weight task and model structure. Then partitioned every one of the parts into the equivalent size of subsets and efficiently discovered corresponding neighbor subsets. At that point, also applied loads of the subsets to the relating subsets for building the last model. Lastly contrasted the model and six distinctive datasets. In a greater part of the cases, this model beats the conventional straight and non-direct classifiers.

Chapter 3

3. Methodology

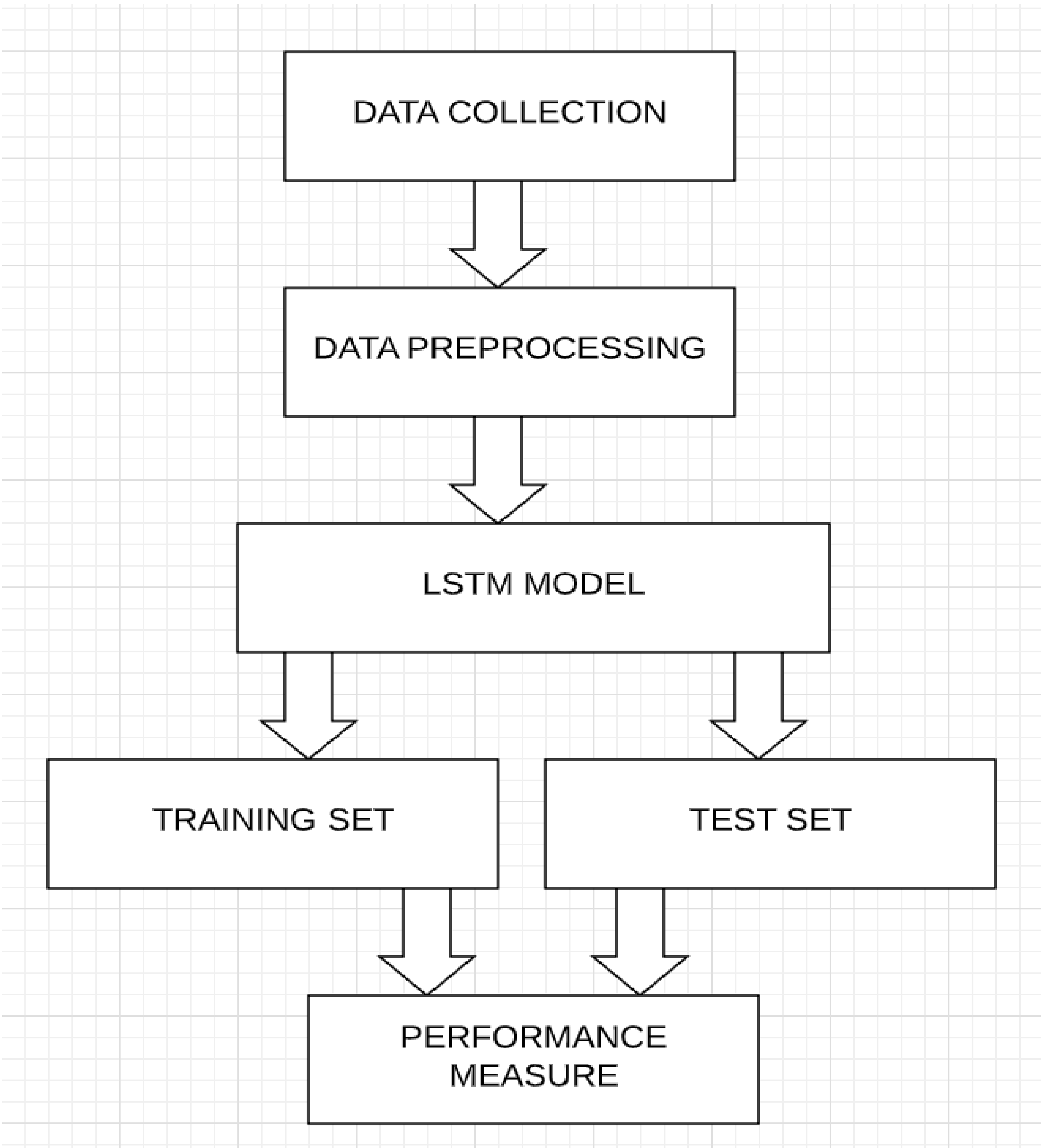


Fig 3: Conceptual model

3.1. Data Collection

In Bangladesh, there is two stock exchange, Dhaka Stock Exchange is one of them. In our paper, we collected some of our data from https://www.dsebd.org, which is the official exchange for the stock market, and the rest of our data collected from Kaggle. For our research, we Choose *Beximco* company stocks. In the dataset, there are Eleven columns. They are Date, Company Name, Ltp(last traded price), Ycp (Yesterday closing price), High, Low, Open (opening price), Close (Closing Price), Trade, Value, Volume.

#	DATE	TRADING CODE	LTP*	HIGH	LOW	OPENP*	CLOSEP*	YCP	TRADE	VALUE (mn)	VOLUME
1	2021-04-13	BEXIMCO	72	72.4	67.5	68.8	72	69	4,040	408.906	5,772,754
2	2021-04-12	BEXIMCO	69	71.2	68.4	68.5	69	68.7	3,884	475.495	6,791,677
3	2021-04-11	BEXIMCO	68.7	71.4	67.7	71	68.7	72.7	4,789	454.072	6,515,447
4	2021-04-08	BEXIMCO	72.7	74.7	72.4	74.7	72.7	74.9	3,509	384.649	5,256,729
5	2021-04-07	BEXIMCO	74.9	76.5	74.3	74.8	74.9	74.2	4,260	508.205	6,724,590
6	2021-04-06	BEXIMCO	74.2	74.6	73	73	74.2	72.3	3,761	443.419	5,992,893
7	2021-04-05	BEXIMCO	72.3	73	68.7	69	72.3	68.5	3,613	328.87	4,588,363
8	2021-04-04	BEXIMCO	68.5	73.1	67.7	70.1	68.5	75.1	6,745	701.496	9,977,315
9	2021-04-01	BEXIMCO	75.1	76	71.6	72.9	75.1	73.8	5,610	706.103	9,608,838
10	2021-03-31	BEXIMCO	73.8	77.4	73.5	77	73.8	77.2	6,888	890.287	11,808,242
11	2021-03-29	BEXIMCO	77.2	79	73.4	73.4	77.2	73.3	9,124	1,016.268	13,230,650
12	2021-03-28	BEXIMCO	73.3	73.6	72	73.1	73.3	73.1	3,454	318.567	4,359,240
13	2021-03-25	BEXIMCO	73.1	74.2	69.2	71.3	73.1	71.5	5,572	611.857	8,438,879
14	2021-03-24	BEXIMCO	71.5	77	70.2	77	71.5	77	7,108	741.194	10,045,480
15	2021-03-23	BEXIMCO	77	78.8	76.2	76.9	77	76	5,967	611.148	7,890,295
16	2021-03-22	BEXIMCO	76	76.7	71.1	74.8	76	74.2	6,911	764.248	10,312,657
17	2021-03-21	BEXIMCO	74.2	77.4	73.6	77.4	74.2	77.2	6,960	842.268	11,243,445
18	2021-03-18	BEXIMCO	77.2	81.4	75.7	81.4	77.2	81.5	8,849	823.462	10,519,937
19	2021-03-16	BEXIMCO	81.5	84	81.2	83.4	81.5	82.5	5,670	736.35	8,913,692
20	2021-03-15	BEXIMCO	82.5	83.8	81.9	83	82.5	83.1	5,250	620.23	7,485,124
21	2021-03-14	BEXIMCO	83.1	85.4	82.5	84.5	83.1	83.6	6,614	819.179	9,733,141
22	2021-03-11	BEXIMCO	83.6	84.5	81.3	82.5	83.6	82.4	7,045	955.878	11,517,991
23	2021-03-10	BEXIMCO	82.4	85.2	81.5	85	82.4	84.4	6,786	731.9	8,764,343

Fig 3.1.1: Dataset from Data Stock Exchange Bangladesh

7/3/2018	BEXIMCO	25.6	26.2	25.5	25.8	25.6	25.7	1568	78.96	3062414
7/2/2018	BEXIMCO	25.8	26.5	25.6	26.4	25.7	26.2	2999	195.816	7555291
6/28/2018	BEXIMCO	26.2	27	26.1	27	26.2	26.5	1826	147.62	5602072
6/27/2018	BEXIMCO	26.6	27	26.4	26.6	26.5	26.5	1629	94.932	3562026
6/26/2018	BEXIMCO	26.5	27.1	26.4	27	26.5	26.9	1757	113.658	4258368
6/25/2018	BEXIMCO	26.7	28.1	26.6	27.5	26.9	27.6	2962	240.924	8748677
6/24/2018	BEXIMCO	27.7	28.1	26.9	27	27.6	26.8	2828	203.547	7428580
6/21/2018	BEXIMCO	26.8	27.4	26.6	27	26.8	26.7	2207	168.704	6258374
6/20/2018	BEXIMCO	26.6	27.3	26.4	26.4	26.7	26.4	1928	127.565	4741865
6/19/2018	BEXIMCO	26.3	27.3	26.2	26.3	26.4	26.3	2263	181.788	6813142
6/18/2018	BEXIMCO	26.6	26.9	26.2	26.7	26.3	26.7	1252	59.518	2251406
6/12/2018	BEXIMCO	26.8	27.2	26.6	26.7	26.7	26.6	1107	58.49	2182505
6/11/2018	BEXIMCO	26.7	27.3	26	26.1	26.6	26.1	1663	106.548	3994790
6/10/2018	BEXIMCO	26.2	27.1	25.9	27	26.1	26.9	2326	135.54	5165601
6/7/2018	BEXIMCO	26.9	27.8	26.8	27.6	26.9	27.5	1954	111.346	4087105
6/6/2018	BEXIMCO	27.5	28.5	27.4	28.1	27.5	27.9	2050	179.05	6406344
6/5/2018	BEXIMCO	28.1	28.4	27.6	27.8	27.9	27.6	2190	137.642	4914880
6/4/2018	BEXIMCO	27.6	28.2	27.4	27.9	27.6	27.7	1440	121.721	4390939
6/3/2018	BEXIMCO	27.6	28.2	27.1	27.4	27.7	27.3	2155	174.788	6305872
5/31/2018	BEXIMCO	27.5	28.3	27.2	28.2	27.3	28.2	2463	151.32	5472256
5/30/2018	BEXIMCO	28	29.4	27.8	28.4	28.2	28.1	4417	274.154	9558373
5/29/2018	BEXIMCO	28.1	28.1	25.4	25.6	28.1	25.6	4849	315.481	11587865
5/28/2018	BEXIMCO	25.4	26.6	25.4	26.6	25.6	26.5	2240	102.284	3938946
5/27/2018	BEXIMCO	26.4	26.9	26.3	26.4	26.5	26.2	1670	87.226	3284243
5/24/2018	BEXIMCO	26.2	26.5	25.8	26.3	26.2	26	2032	117.822	4500869
5/23/2018	BEXIMCO	25.9	27.1	25.7	27	26	27	2668	147.173	5581868
5/22/2018	BEXIMCO	26.8	27.6	26.8	27.4	27	27.2	1896	108.558	3982478
5/21/2018	BEXIMCO	27.2	27.7	26.2	26.6	27.2	26.4	2842	164.006	6019068
5/20/2018	BEXIMCO	26.5	27.5	26.2	27.2	26.4	26.9	1825	99.247	3715132
5/17/2018	BEXIMCO	27	28.1	26.8	28.1	26.9	27.9	2534	136.444	5007389

Fig 3.1.2: Dataset from Kaggle

3.2. Data Preprocessing

The dataset we collected from Kaggle is in JSON format. So, we use JSON and CSV libraries in python from converting our dataset from JSON to CSV. After that, we manually check missing values and zero values. Then, we use Pearson's Correlation Coefficient for measuring the association of our predicted attribute with the other attribute. Finally, we find they are correlated.


```
{"date": "2009-01-19 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 267.0, "high": 268.0, "low": 265.2, "o
{"date": "2009-01-18 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 270.0, "high": 272.0, "low": 270.0, "o
{"date": "2009-01-15 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 271.5, "high": 280.0, "low": 270.0, "o
{"date": "2009-01-14 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 273.0, "high": 274.1, "low": 272.5, "o
{"date": "2009-01-13 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 274.0, "high": 280.0, "low": 274.0, "o
{"date": "2009-01-12 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 274.0, "high": 281.9, "low": 272.0, "o
{"date": "2009-01-11 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 274.2, "high": 285.0, "low": 271.5, "o
{"date": "2009-01-07 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 278.1, "high": 284.3, "low": 278.0, "o
{"date": "2009-01-06 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 280.0, "high": 282.5, "low": 276.0, "o
{"date": "2009-01-05 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 283.0, "high": 299.8, "low": 282.5, "o
{"date": "2009-01-04 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 285.0, "high": 287.9, "low": 271.0, "o
{"date": "2009-01-01 00:00:00", "trading_code": "BERGERPBL", "last_traded_price": 281.9, "high": 285.0, "low": 273.1, "o
{"date": "2009-12-30 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 309.9, "high": 310.0, "low": 304.0, "ope
{"date": "2009-12-29 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 305.5, "high": 306.9, "low": 304.2, "ope
{"date": "2009-12-27 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 306.1, "high": 308.0, "low": 304.0, "ope
{"date": "2009-12-24 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 306.7, "high": 311.0, "low": 306.1, "ope
{"date": "2009-12-23 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 307.6, "high": 309.1, "low": 303.5, "ope
{"date": "2009-12-22 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 303.5, "high": 304.7, "low": 299.0, "ope
{"date": "2009-12-21 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 300.6, "high": 303.9, "low": 298.0, "ope
{"date": "2009-12-20 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 302.3, "high": 309.7, "low": 300.9, "ope
{"date": "2009-12-17 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 305.0, "high": 313.0, "low": 304.5, "ope
{"date": "2009-12-15 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 308.6, "high": 313.0, "low": 304.8, "ope
{"date": "2009-12-14 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 309.6, "high": 316.3, "low": 309.2, "ope
{"date": "2009-12-13 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 311.7, "high": 312.0, "low": 304.2, "ope
{"date": "2009-12-10 00:00:00", "trading_code": "BEXIMCO", "last_traded_price": 304.1, "high": 306.0, "low": 299.3, "ope
```


Fig 3.2.1: Kaggle data in JSON format

```python
import csv
import json

cols=['date','trading_code','last_traded_price','high','low','opening_price','closing_price',
'yesterdays_closing_price','trade','value_mn','volume']

with open('Data/A/archive/New folder/prices_2019.json') as json_file:
    data = json.load(json_file)

path = "Data/A/prices_2019.csv"
with open(path, 'w') as f:
    wr = csv.DictWriter(f, fieldnames = cols)
    wr.writeheader()
    wr.writerows(data)
```

Fig 3.2.2: Code for converting the dataset from JSON to CSV

Out[12]:

	r	p
#	-0.479371	5.993713e-27
LTP*	0.997239	0.000000e+00
HIGH	0.995977	0.000000e+00
LOW	0.995989	0.000000e+00
OPENP*	0.994337	0.000000e+00
CLOSEP*	1.000000	0.000000e+00
YCP	0.996378	0.000000e+00
TRADE	0.858124	2.469770e-130
VALUE (mn)	0.915686	1.728602e-177
VOLUME	0.672270	7.924030e-60

Fig 3.2.3: Pearson's Correlation Coefficient

```python
df=df.dropna()
```
[11] Python

```python
from scipy import stats as sc
corr_df=pd.DataFrame(columns=['r','p'])
for col in df:
    if pd.api.types.is_numeric_dtype(df[col]):
        r,p = sc.pearsonr(df['CLOSEP*'],df[col])
        corr_df.loc[col]=[r,p]
corr_df
```
[12] Python

Fig 3.2.4: Pearson's Correlation Coefficient code

3.3. Data Mining Technique

Recurrent Neural Network (RNN) is one type of neural network where the output of one hidden layer is used as an input of the next layer. Long short term memory is a special kind of RNN that can remember information for longer periods. This is the cell state which holds all the information in a vector form.

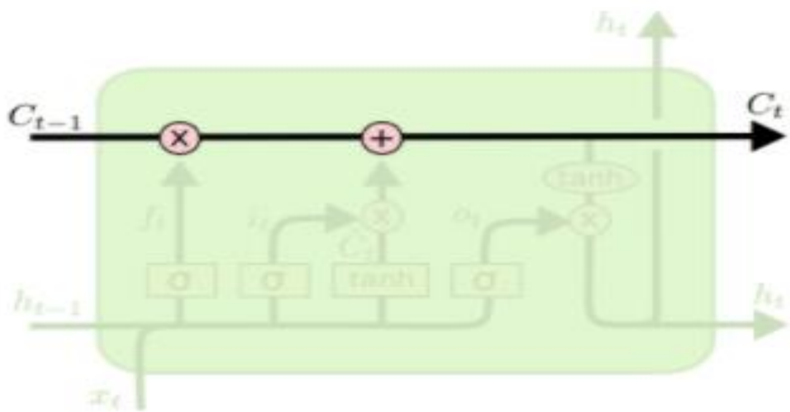
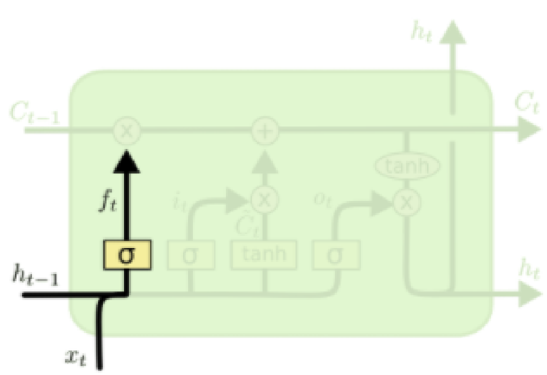


Fig 3.3.1: Cell State

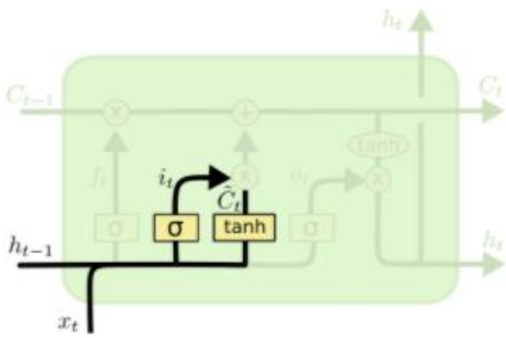
Next, it has a forget gate layer, which adds or removes information in the cell state. It gets the output from the previous layer and concatenates with the current input layer and passes it in the sigmoid function. From the function the output will be 0 or 1 .1 means information will be kept and 0 means it will be discarded.



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

Fig 3.3.2: Forget gate layer

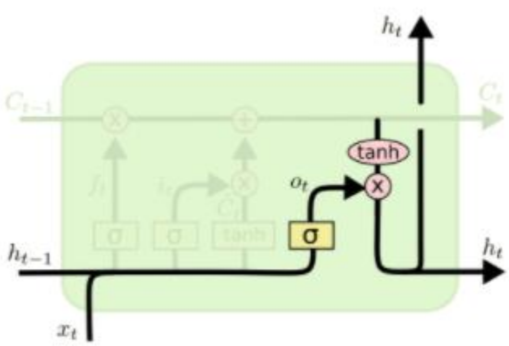
This step will be used for adding new information in the cell state. It has two parts, one is the sigmoid layer which holds the new values and another is the layer that will contain the vector of the new information. Then, the output of this two-layer will be combined and added to the memory cell.



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

Fig 3.3.3: Input layer

Finally, now the total output of this layer will be decided in two phases. First, the input and the previous layer will be passed in a sigmoid layer again and second cell state information will be passed in tanh and then these two layers' results will be multiplied. This result will be passed in the next layer.



$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Fig 3.3.4: Output Layer

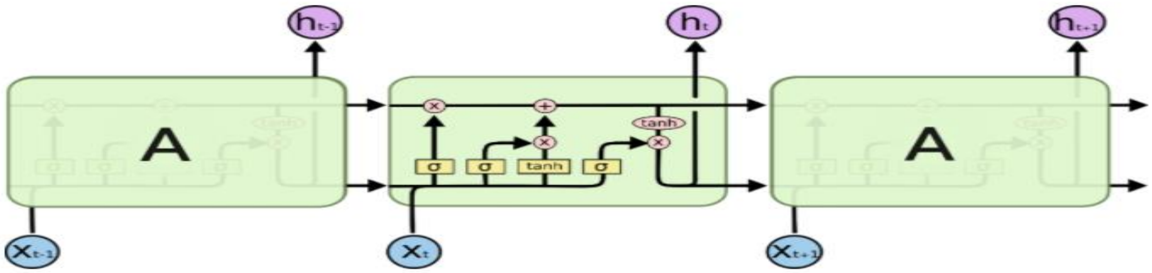


Fig 3.3.5: Long short term memory

In the model, stacked Lstm was used. Stacked Lstm means the combination of multiple Lstm layers. One Lstm gives a sequence without giving a value and then the sequence will be feed in the next stacked Lstm layer. Then, after completing all the Lstm layers, in the output layer, it gives a specific value. This technique is very much stable for predicting sequential data. The model has four hidden layers and a dense output layer. In each layer, there are fifty neurons. The number of epochs is fifty with a batch size of 32. Then, the adam optimizer was used and the dropout will be 0.2 and for loss mean square root was used. This model tries to predict the closing price of the next 30 days based previous 30 days of data.

ThesisCode.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[105]
```
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
```

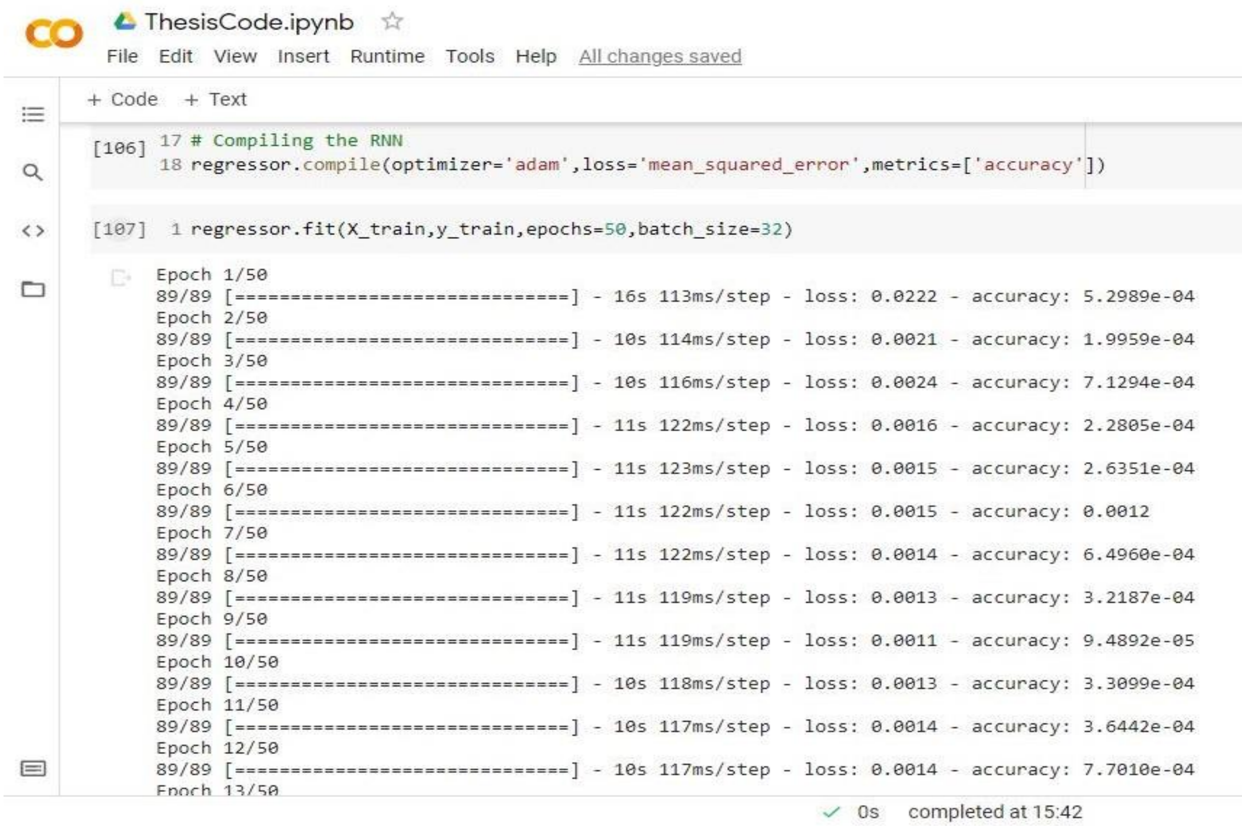
[106]
```
regressor = Sequential()
# First LSTM layer with Dropout regularisation
regressor.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1],1)))
regressor.add(Dropout(0.2))
# Second LSTM layer
regressor.add(LSTM(units=50, return_sequences=True))
regressor.add(Dropout(0.2))
# # Third LSTM layer
regressor.add(LSTM(units=50, return_sequences=True))
regressor.add(Dropout(0.2))
# Fourth LSTM layer
regressor.add(LSTM(units=50))
regressor.add(Dropout(0.2))
# The output layer
regressor.add(Dense(units=1))

# Compiling the RNN
regressor.compile(optimizer='adam',loss='mean_squared_error',metrics=['accuracy'])
```

[107]
```
regressor.fit(X_train,y_train,epochs=50,batch_size=32)
```

```
Epoch 1/50
```

✓ 0s completed at 15:42

Fig 3.3.6: Code Snippet of Lstm model



ThesisCode.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[106] 17 # Compiling the RNN
      18 regressor.compile(optimizer='adam',loss='mean_squared_error',metrics=['accuracy'])

[107]  1 regressor.fit(X_train,y_train,epochs=50,batch_size=32)

Epoch 1/50
89/89 [==============================] - 16s 113ms/step - loss: 0.0222 - accuracy: 5.2989e-04
Epoch 2/50
89/89 [==============================] - 10s 114ms/step - loss: 0.0021 - accuracy: 1.9959e-04
Epoch 3/50
89/89 [==============================] - 10s 116ms/step - loss: 0.0024 - accuracy: 7.1294e-04
Epoch 4/50
89/89 [==============================] - 11s 122ms/step - loss: 0.0016 - accuracy: 2.2805e-04
Epoch 5/50
89/89 [==============================] - 11s 123ms/step - loss: 0.0015 - accuracy: 2.6351e-04
Epoch 6/50
89/89 [==============================] - 11s 122ms/step - loss: 0.0015 - accuracy: 0.0012
Epoch 7/50
89/89 [==============================] - 11s 122ms/step - loss: 0.0014 - accuracy: 6.4960e-04
Epoch 8/50
89/89 [==============================] - 11s 119ms/step - loss: 0.0013 - accuracy: 3.2187e-04
Epoch 9/50
89/89 [==============================] - 11s 119ms/step - loss: 0.0011 - accuracy: 9.4892e-05
Epoch 10/50
89/89 [==============================] - 10s 118ms/step - loss: 0.0013 - accuracy: 3.3099e-04
Epoch 11/50
89/89 [==============================] - 10s 117ms/step - loss: 0.0014 - accuracy: 3.6442e-04
Epoch 12/50
89/89 [==============================] - 10s 117ms/step - loss: 0.0014 - accuracy: 7.7010e-04
Epoch 13/50
```

✓ 0s completed at 15:42

ThesisCode.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[107] 89/89 [==============================] - 10s 115ms/step - loss: 8.6290e-04 - accuracy: 1.5020e-04
Epoch 37/50
89/89 [==============================] - 10s 115ms/step - loss: 9.5522e-04 - accuracy: 3.5599e-05
Epoch 38/50
89/89 [==============================] - 10s 115ms/step - loss: 8.2812e-04 - accuracy: 6.0974e-04
Epoch 39/50
89/89 [==============================] - 10s 115ms/step - loss: 0.0010 - accuracy: 0.0014
Epoch 40/50
89/89 [==============================] - 10s 117ms/step - loss: 8.1826e-04 - accuracy: 3.1666e-04
Epoch 41/50
89/89 [==============================] - 11s 118ms/step - loss: 7.6543e-04 - accuracy: 7.2809e-04
Epoch 42/50
89/89 [==============================] - 10s 117ms/step - loss: 9.0576e-04 - accuracy: 4.8523e-04
Epoch 43/50
89/89 [==============================] - 10s 115ms/step - loss: 8.7989e-04 - accuracy: 7.4044e-04
Epoch 44/50
89/89 [==============================] - 10s 117ms/step - loss: 9.1909e-04 - accuracy: 0.0012
Epoch 45/50
89/89 [==============================] - 10s 115ms/step - loss: 7.6126e-04 - accuracy: 2.0762e-04
Epoch 46/50
89/89 [==============================] - 10s 117ms/step - loss: 7.9479e-04 - accuracy: 2.4781e-04
Epoch 47/50
89/89 [==============================] - 10s 114ms/step - loss: 6.8785e-04 - accuracy: 4.5441e-04
Epoch 48/50
89/89 [==============================] - 10s 117ms/step - loss: 7.9367e-04 - accuracy: 6.9293e-04
Epoch 49/50
89/89 [==============================] - 10s 117ms/step - loss: 9.6585e-04 - accuracy: 2.3130e-04
Epoch 50/50
89/89 [==============================] - 10s 118ms/step - loss: 7.7240e-04 - accuracy: 4.0216e-04
<tensorflow.python.keras.callbacks.History at 0x7fc59c7f20d0>
```

✓ 0s completed at 15:42

Fig 3.3.7: Epoch and decreasing loss

Chapter 4

4. Result and Discussion

4.1. Result Analysis

For testing the model, the dataset was divided into two parts. One was a training set and another was a test set. In the training set, data from 1 January 2009 to 2 February 2021, a total number of 2893 data were used to train the model. In the test set, a full month of March 2021, the total number of 31 data were used. For performance metrics, the root means square error was applied. The RMSE value was 0.33. The accuracy was 65.93%.

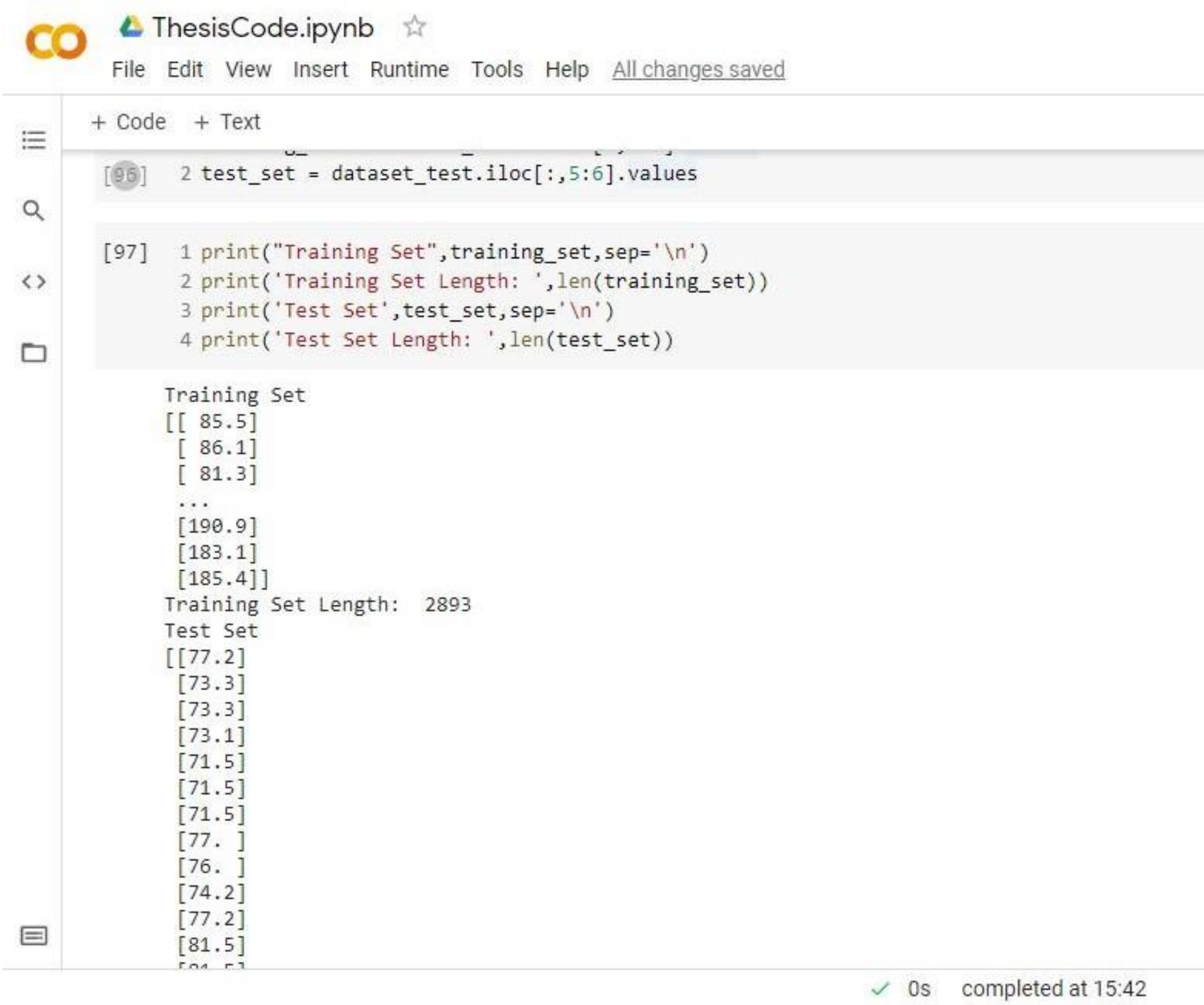


Fig 4.1.1: Training Set

ThesisCode.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[71.5]
[71.5]
[77. ]
[76. ]
[74.2]
[77.2]
[81.5]
[81.5]
[81.5]
[82.5]
[82.5]
[83.1]
[83.6]
[82.4]
[82.4]
[82.4]
[84.4]
[81.9]
[81.9]
[82.1]
[84. ]
[84. ]
[84. ]
[85.4]
[83.3]
[80.5]]
Test Set Length:  31
```

```
[98]  import matplotlib.pyplot as plt
```

✓ 0s completed at 15:42

Fig 4.1.2: Test Set

ThesisCode.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[92]  import math
      from sklearn.metrics import mean_squared_error

[101] def plot_predictions(test,predicted):
          plt.plot(test, color='red',label='Real Beximco Stock Price')
          plt.plot(predicted, color='blue',label='Predicted Beximco Stock Price')
          plt.title('Beximco Stock Price Prediction')
          plt.xlabel('Time')
          plt.ylabel('Beximco Stock Price')
          plt.legend()
          plt.show()

[102] def return_rmse(test,predicted):
          rmse = math.sqrt(mean_squared_error(test, predicted))
          print("The root mean squared error is {}.".format(rmse/100))
          print("Accuracy is:",rmse*1.96,"%")
          return rmse

[100] from sklearn.preprocessing import MinMaxScaler

[103] sc = MinMaxScaler(feature_range=(0,1))
      training_set_scaled = sc.fit_transform(training_set)

[59]  import numpy as np
```

✓ 0s completed at 15:42

```
The root mean squared error is 0.33642527356947527.
Accuracy is: 65.93935361961715 %
```

✓ 0s completed at 15:42

Fig 4.1.3: Model performance (RMSE & Accuracy)

4.2. Visualization

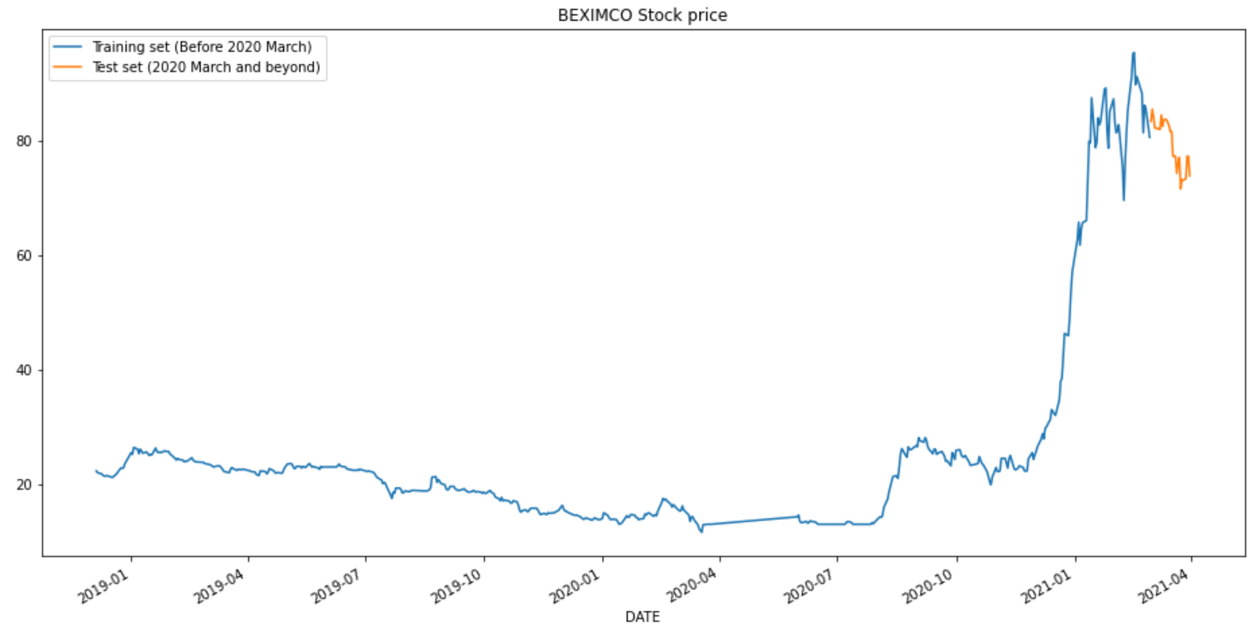


Fig 4.2.1: Training and Test Set Plot

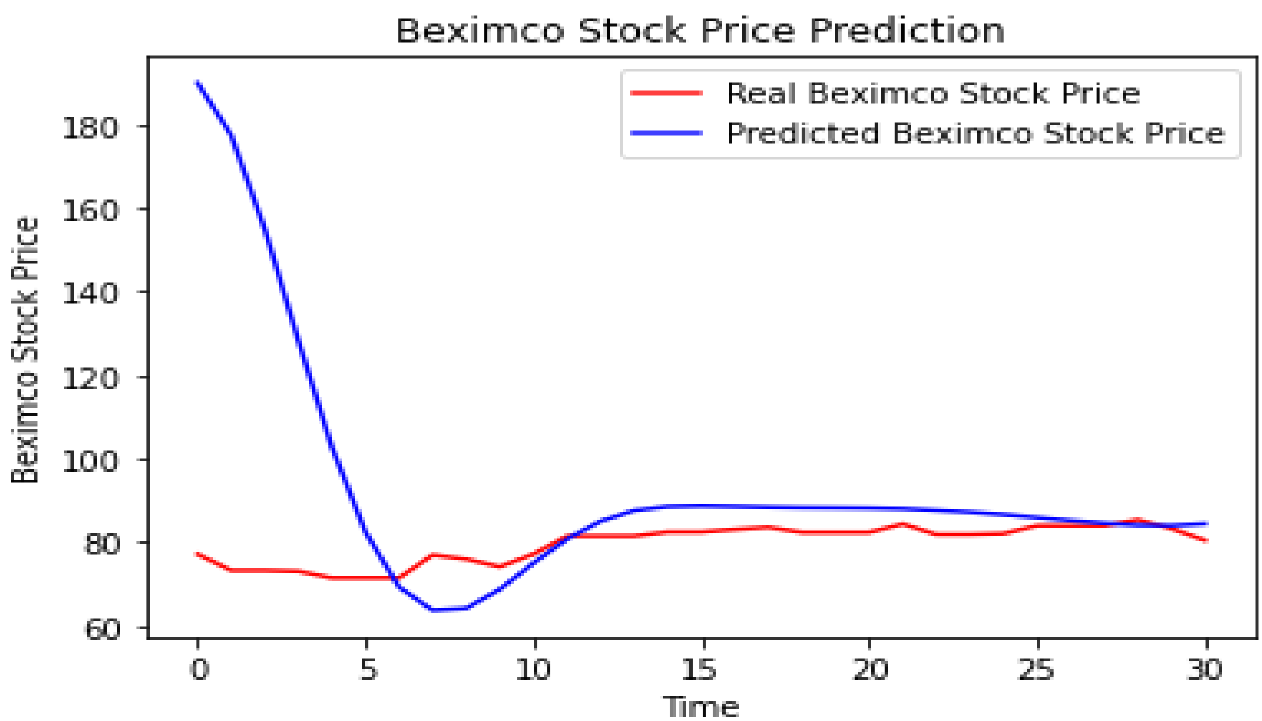


Fig 4.2.2: Real Vs Predicted Close Price

4.3. Discussion

Datasets are collected from *Kaggle* and *Dhaka Stock Exchange* official sites as mentioned early. We use the data from both sources because it will be more accurate and reliable for better output. We have a total number of 2924 instances and 11 attributes. For this research, we used all the data some as a training set and others as the test set.

After applying the Lstm model, we get sixty-five percentage accuracy. The root means squared error value was less than 0.5. From the plot and graph, we can say that the values which are predicted by the model are close enough. As a result, our model perfectly predicts the closing price will be down or up. After analyzing the result we get our desired output. Many investment sites can apply this method for analyzing the price of the stocks.

In the end, visualization in the plot helps the user to understand the outcome. For predicting this type of prediction which involves with the sequence of numbers this data mining technique will be helpful for further analysis.

Chapter 5

5. Conclusion

Stocks data is a vital field in which successful data preprocessing has played a very important role in predicting the prices which can help the investor to take a proper decision of their finance. As the numbers of people are getting interested in this field exponentially. For this new company offer their market shares for growing their business. Many startups use these data mining techniques for forecasting all the stock attributes, by using this technique they can build a robust system that can help someone to manage their personal finance by using the system. The more time and data the machine gets, it will be more accurate. To analyze stock data from a database, data mining methods can be used. The outcomes of this data mining could theoretically be used in the next few years to mitigate and even deter stock prices. We assume there is a bright future for stock data mining to enhance the efficiency and effectiveness of investigative and intelligence research.

Chapter 6

6. Future Work

The stock market is a very much important sector for investment. Number of peoples associated with this industry is increasing rapidly. Machine learning has the potential to find the pattern and predict these stock prices and can improve reliability day by day. Stock Information not as it depended on the Drift within the previous Information, it too primarily depends on the item esteem or the fulfillment of the clients with the company's showcase[13].In the future, we want to extend our model with various ranges and apart from the closing price, we also try to predict other parameters as well. The prices of the stocks depend on the satisfaction of the customer on the product, we will try to develop a system where it will predict the prices in terms of the news and some reviews of the product from various resources.

References

[1] Qiu J, Wang B, Zhou C (2020) Forecasting stock prices with long-short term memory neural network based on attention mechanism. PLoS ONE 15(1): e0227222. https://doi.org/10.1371/journal.pone.0227222.

[2] Xu T, Zhang J, Ma Z, et al. Deep LSTM for Large Vocabulary Continuous Speech Recognition[J]. 2017.

[3] Ha Young Kim,Chang Hyun Won. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models[J]. Expert Systems With Applications,2018,103.

[4] Stock Market Prediction Using Data Mining Techniques 1.Archana Gupta 2.Pranay Bhatia 3.Kashyap Dave 4.Pritesh Jain.

[5] Stock Market Prediction on Bigdata Using Machine Learning Algorithm V.Sandhiya1, T.Revathi2, A.Jayashree3, A.Ramya4, S.Sivasankari5 B.Tech Student1, 2, 3, 4, Assistant Professor5 Department of Information Technology Kingston Engineering College, Tamilnadu, India.

[6] Kulshrestha, Tanisha. “Pattern Recognition in Stock Market.” Tanisha Kulshrestha (2020).

[7] Qiu J, Wang B, Zhou C (2020) Forecasting stock prices with long-short term memory neural network based on attention mechanism.

[8] Weng, Bin & Ahmed, Mohamed & Megahed, Fadel. (2017). Stock Market One-Day Ahead Movement Prediction Using Disparate Data Sources. Expert Systems with Applications. 79. 10.1016/j.eswa.2017.02.041.

[9] Anita Yadav, C K Jha, Aditi Sharan,Optimizing LSTM for time series prediction in Indian stock market,Procedia Computer Science,Volume 167,2020.Maksuda Akter Rubi & Md Kamrul Hossain

[10] Maksuda Akter Rubi & Md Kamrul Hossain Forecasting dse broad index: an application of multi-layer feed-forward neural network.

[11] Desai, N., & Gandhi, N. (2014). Stock Market Prediction Using Data Mining. International Journal of Engineering Development and Research, 2, 2780- 2784.

[12] S. Al Rafi, T. Rahman, A. R. Al Abir, T. A. Rajib, M. Islam, and M. S. Hossain Mukta, "A New Classification Technique: Random Weighted LSTM (RWL)," 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2020.

[13] Yang, Y., & Hasuike, T. (2017) Construction of Investor Sentiment Index in the Chinese Stock Market. In Advanced Applied Informatics (IIAI–AAI), 2017 6th IIAI International Congress on (pp. 23–28). IEEE

[14] Nti, I.K., Adekoya, A.F. Weyori, B.A. A systematic review of fundamental and technical analysis of stock market predictions. Artif Intell Rev 53, 3007–3057 (2020).

[15] Kulshrestha, Tanisha. "Pattern Recognition in Stock Market." Tanisha Kulshrestha (2020)

[16] Stock Market Prediction on Bigdata Using Machine Learning Algorithm V.Sandhiya 1. T.Revathi 2. A.Jayashree 3. A.Ramya 4. S.Sivasankari B.Tech Student,Assistant Professor, Department of Information Technology Kingston Engineering College, Tamilnadu, India

[17] Evaluating and understanding text-based stock price prediction models,Enric Junqué de Fortuny a,, Tom De Smedt b, David Martens a, Walter Daelemans b .a Faculty of Applied Economics, University of Antwerp, Prinsstraat 13, B-2000 Antwerp, Belgium .b) Faculty of Arts, University of Antwerp, Prinsstraat 13, B-2000 Antwerp, Belgium

[18] Predicting stock prices using data mining techniques 1.Qasem A. Al-radaideh 2 Adel Abu Assaf 3 Eman Alnagi.

[19] Predicting Stock Trends Using Tsk-fuzzy Rule-Based System 1.A. D. Adebayo 2.A. F. Adekoya 3.M. T. Rahman.

[20] Stock Market Prediction Using Data Mining Techniques 1.Archana Gupta 2.Pranay Bhatia 3.Kashyap Dave 4.Pritesh Jain.