International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 6 Issue 3, March-April 2022 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Stock Market Prediction using Machine Learning

Subham Kumar Gupta[1], Dr. Bhuvana J[2], Dr. M N Nachappa[3]

[1]MCA Student, Department of Information Technology, JAIN Deemed-to-be University, Bengaluru, Karnataka, India
[2]Head, Department of Information Technology, JAIN Deemed-to-be University, Bengaluru, Karnataka, India
[3]Head, School of Computer Science & IT, JAIN Deemed-to-be University, Bengaluru, Karnataka, India

ABSTRACT
Stock market prediction is a typical task to forecast the upcoming stock values. It is very difficult to forecast because of unbalanced nature of stocks. In this work, an attempt is made for prediction of stock market trend. This research aims to combine multiple existing techniques into a much more robust prediction model which can handle various scenarios in which investment can be beneficial. By combing both techniques, this prediction model can provide more accurate and flexible recommendations.

However instead of using those traditional methods, we approached the problems using machine learning techniques. We tried to revolutionize the way people address data processing problems in stock market by predicting the behavior of the stocks. In fact, if we can predict how the stock will behave in the short-term future we can queue up our transactions earlier and be faster than everyone else. In theory, this allows us to maximize our profit without having the need to be physically located close to the data sources.

We examined three main models. Firstly we used a complete prediction using a moving average. Secondly we used a LSTM model and finally a model called ARIMA model. The only motive is to increase the accuracy of predictive the stock market price. Each of those models was applied on real stock market data and checked whether it could return profit.

KEYWORDS: *Stock market prediction*

How to cite this paper*:* Subham Kumar Gupta | Dr. Bhuvana J | Dr. M N Nachappa "Stock Market Prediction using Machine Learning" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-6 | Issue-3, April 2022, pp.2046-2061, URL: www.ijtsrd.com/papers/ijtsrd49868.pdf

IJTSRD49868

1. INTRODUCTION
Stock price prediction is very important as it is used by most of the business people as well as common people. People will either gain money or lose their entire savings in stock market activity. Many Stock holders invest more money on this stock market. Without having a clear idea on stock market, many people are losing a lot of money.

1.1. Motivation
Predicting the movement of stocks in the competitive financial markets is a challenge even for the most experienced day trader. Even in a fraction of a second the price of a stock can change so drastically that the first one who is able to see it and act can win huge amount of money while the rest have to face a financial disaster. Through the years many experts used a variety of methods in order to try and predict the unpredictable stock market and earn money.

1.2. Goals and Limitations
In this section we discuss the Goals and the Limitations of this thesis. We explain in detail what we want to achieve through the thesis and what difficulties we had to overcome to make it happen.

1.3. Challenges and Limitations
Stock market is so complicated and many things can affect the change in a price. Not only financial factors can influence the price of a stock. Things like news or the general mood can affect the price in many ways positive or negative. If it was possible to model the stock market with a function it would be a complex function that lives in high-dimensional, maybe infinite dimensional, space. Imagine what would happen if someone knew a away to calculate that function. That someone would be able to profit by taking advantage of it. However the nature of the

space is so complicated that finding that function is an impossible thing to do. The real challenge is to try and approximate that function using neural-networks in a way that we can profit by applying it in the stock market. The focus of this thesis is to try to approximate the stock market as good as possible and try to maximize our profit.

2. Literature Review

The stock returns is an area of study wherein many research scholars have shown immense interest for past several years. A brief review of literature will help in understanding the relevance of the content analysis in the area of stock returns. The first set of articles includes studies that primarily focus on stock market prediction using Moving Average, ARIMA MODEL and LSTM. The researches in social sciences or in the field of economics depend in one way or the other on careful reading of written materials and the research work done by many research scholars on similar subjects. Considering this fact, the importance of content analysis becomes very significant.

The objective of this paper is to construct a model to predict stock worth movement mistreatment the Moving Average, ARIMA MODEL and LSTM to predict National securities market (NSE). It used domain specific approach to predict the stocks from every domain and brought some stock with most capitalization. Topics and connected opinion of shareholders are mechanically extracted from the writings in an exceedingly message board by utilizing our projected strategy aboard uninflected clusters of comparable type of stocks from others mistreatment clump algorithms.

The various areas to which the technique of content analysis can be applied are based on the user's skill and ingenuity in framing valid category formats as discussed in the research. Stock price prediction is a challenging task owing to the complexity patterns behind time series. Autoregressive integrated moving average (ARIMA) model, Moving Average and Long-Short Term Memory (LSTM) model are popular linear and nonlinear models for time series forecasting respectively. The integration of two models can effectively capture the linear and nonlinear patterns hidden in a time series and improve forecast accuracy. In this paper, a new hybrid ARIMA-BPNN model containing technical indicators is proposed to forecast four individual stocks consisting of both main board market and growth enterprise market in software and information services.

Barelson (1952) defined content analysis as a technique of research that is systematic representation of the matter of communication. According to Stone (1964), the content analysis is a methodology or procedure which can be used to access particular information based on the past references. The definition of content analysis requires that the inference be derived from the counts of frequency to place a number of standard methods on the borderline of acceptability (Leites & Poo, 1942).

Enke and Thawornwong (2005) use a machine learning information gain technique to evaluate the predictive relationships for numerous financial and economic variables. By computing the information gain for each model variable, a ranking of the variables is obtained. A threshold is determined to select only the strongest relevant variables to be retained in the forecasting models.

2.1. SCOPE OF THE STUDY

Data forecasting is really convenient topic of research from last few decades and may remain active topic in upcoming years also. Stock price prediction also used data forecasting is basically is one of favorite topic among researchers. A lot of research is already done in this field like Stock price prediction can be using neural, fuzzy, machine learning, R programming and so on. Here we want to predict future stock price using the some predictive services. This will provide help to get more accurate results for predicting stock price prediction. In future we can analyze this stock market historical data in some other way to find more accurate results. We can deal with the not only finding of future stock price prediction but also tried to reduce mismatch value i.e. difference between actual price and predicted price. Threshold value can be reduced to move toward more accurate value.

3. Prediction Techniques

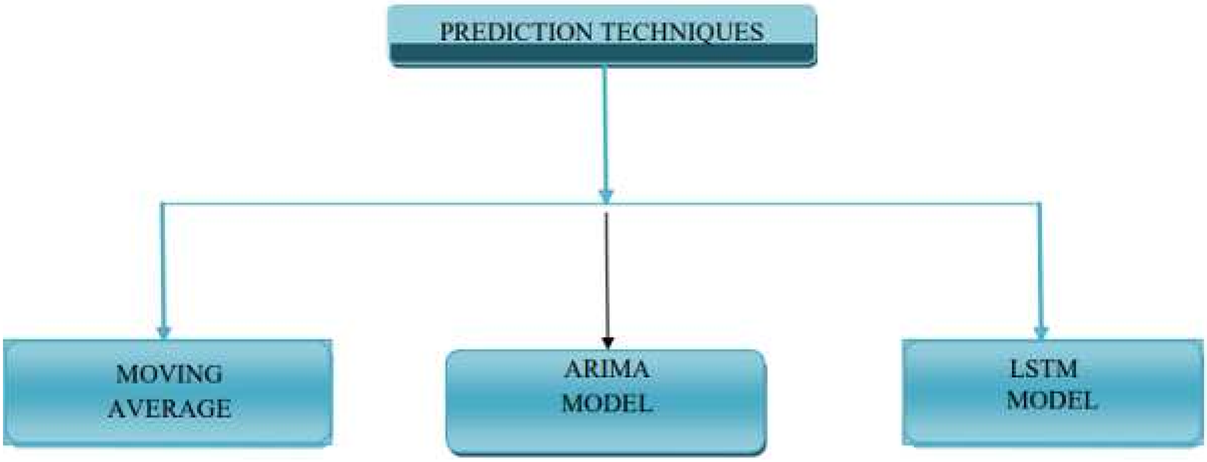

Fig 1: Prediction Techniques

3.1. MOVING AVERAGE

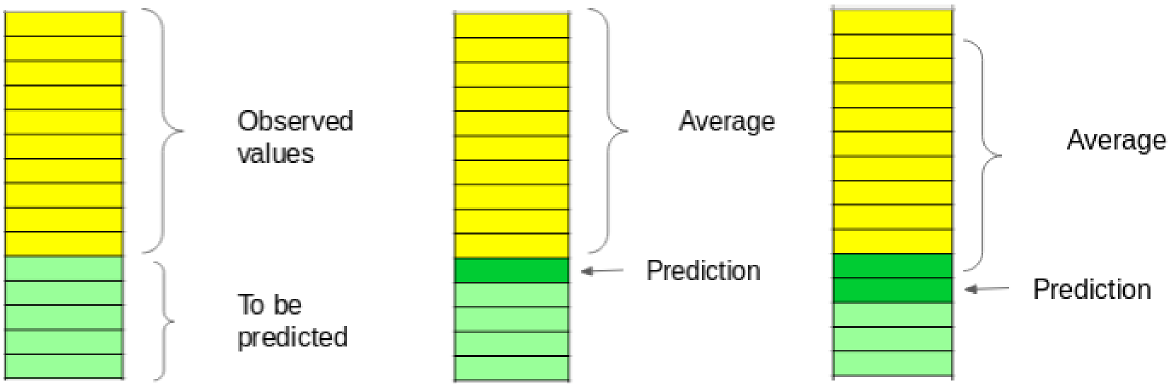

Fig 2: Moving Average

'Average' is easily one of the most common things we use in our day-to-day lives. For instance, calculating the average marks to determine overall performance, or finding the average temperature of the past few days to get an idea about today's temperature – these all are routine tasks we do on a regular basis. So this is a good starting point to use on our dataset for making predictions.

3.2. ARIMA MODEL

Auto-**R**egressive **I**ntegrated **M**oving **A**verage is a model which is used in statistics and econometrics to measure events that happen over a period of time. It's a class of statistical models for analyzing and forecasting time series data. The model understands past data and predicts future data in the series. It's used when a metric is recorded in regular intervals, from fractions of a second to daily, weekly or monthly periods.

AR – Auto-regression: It predicts future values based on past values.

I - integrated: The use of differencing of raw observations in order to make the time series stationary

MA - Moving Average: It is the dependency between an observed value and a residual error from a moving average model applied to previous observations.

In forecasting stock prices, the model reflects the differences between the values in a series rather than measuring the actual values.

ADVANTAGES:

∑ ARIMA model has a fixed structure and is specifically built for time series (sequential) data.

∑ ARIMA works better for relatively short series when the number of observations is not sufficient to apply more flexible methods.

DISADVANTAGES:

∑ ARIMA models can only be highly accurate and reliable under the appropriate conditions and data availability.

∑ The ARIMA model tends to be unstable, both with respect to changes in observations and changes in model specification.

3.3. LSTM – Long Short – Term Memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems as they can store past information. This is important as the previous price of a stock is crucial in predicting its future price. LSTM neural networks are capable of solving numerous tasks that are not solvable by previous learning algorithms like RNNs. Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. LSTM is well-suited to classify, process, and predict time series given time lags of unknown duration. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate.

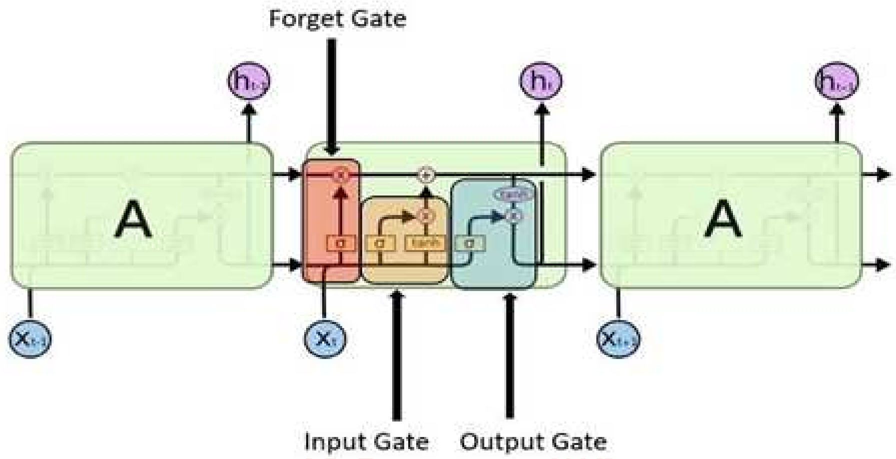

Fig 3: LSTM

ADVANTAGES:

- LSTM cells have a memory that can store previous time step information and use it to train the dataset. It has the ability to bridge very long time lags.
- LSTM doesn't have the vanishing gradient problem which a traditional RNN has.

DISADVANTAGES:

- They require a lot of resources and time to get trained and become ready for real-world applications

What is GCP based on?

What is GCP? GCP is a **public cloud vendor** — like competitors Amazon Web Services (AWS) and Microsoft Azure. With GCP and other cloud vendors, customers are able to access computer resources housed in Google's data centers around the world for free or on a pay-per-use basis.

4. Financial Definition

4.1. Stock Market

Stock prediction is using historical price, related market information and so on to forecast exact price or price trend of the stock in the near future. According to the time granularity of price information, stock trading can be divided into low-latency trading based on daily basis and high-frequency trading, which market exchanges in a matter of hours, minutes, even seconds. High-frequency trading analysis is more common in hedge funds, investment banks, and large institutional investors. It masters the trading signals before prices' ups and downs through analyzing great amount of trading data [20]. In this thesis, only low-latency trading is taking into consideration, which is more common in academia. Its core concept is to increase the accuracy of stock prediction based on the related market information

The Indian stock market mainly studies stocks traded at National Stock Exchange (NSE) and Bombay Stock Exchange (BSE). NSE or National Stock Exchange is located in Mumbai, and it is India's leading stock exchange market. It first came into existence in 1992 and brought with it an electronic exchange system in India, which led to the removal of the paper based system. In 1875, BSE or Bombay Stock Exchange was established, and it was formerly known as 'The native share and stock brokers association'. However, after 1957, Government of India recognized this stock exchange as the premier stock exchange of India, under the Securities Contract Regulation Act, 1956.

The BSE's Sensex comprises of 30 companies, while NSE's Nifty comprises of 50 companies.

Both the stock exchanges, National Stock Exchange and Bombay Stock Exchange, are an important part of Indian Capital Market. Every day, hundreds of thousands of brokers and investors trade on these stock exchanges. And both are established in Mumbai, Maharashtra, and SEBI (Securities and Exchange Board of India) recognized.

4.2. Stock Trend Definition

In this thesis, I will mainly focus on predicting ups and downs of stock. Stocks leave some important trading data after each trading day, such as open price, close price, adjusted close price, highest price, lowest price, volume, etc. Among all, adjusted close price usually represents the stock price of that trading day. In the trading period, there will be a series of adjusted close prices. Let us denote it as:

p1, p2, p3,..., pT .

Here, pt is the close price on t trading day, T is total trading days in this period. Stock price of a certain trading day will rise or drop comparing to previous trading day, thus, here I used the change of closing price of two consecutive trading days as the judgment. Let us denote trading situation as:

yt = 8 ><>: 1 if pt > pt1, 0 if pt □ pt1.

If it is 1, it means the price goes up on the second trading day. Otherwise, if it is 0, it means the price goes down or remains the same.

5. Objective of the Study

Objectives of the study are defined as follow:

- The main objective of this study is to predict the future stock price by analyzing the past historical data that we were going to collect from the National Stock Exchange.
- Predicting the Stock market cost in such a way that it will provide most accurate results.
- Stock market price forecasting should be done in such a way that predicted price should minimize the threshold value (difference between actual value and predicted value also known as mispricing) and close enough to the actual value.
- Process of analyzing the historical data should be simple and easy to understand. For this feature identification can done intelligently to provide most accurate results.
- To increase the efficiency of the data analysis technique by using some cloud based tools.
- To analyze the performance and comparing proposed algorithm with the existing algorithms in terms of predicted price accuracy, close price predicted and accurate close price etc.

6. Methodology

6.1. PROBLEM STATEMENTS

Stock market is so complicated and many things can affect the change in a price. Not only financial factors can influence the price of a stock. Things like news or the general mood can affect the price in many ways positive or negative. If it was possible to model the stock market with a function it would be a complex function that lives in high-dimensional, maybe infinite dimensional, space.

Imagine what would happen if someone knew a away to calculate that function. That someone would be able to profit by taking advantage of it. However the nature of the space is so complicated that finding that function is an impossible thing to do. The real challenge is to try and approximate that function using neural-networks in a way that we can profit by applying it in the stock market. The focus of this thesis is to try to approximate the stock market as good as possible and try to maximize our profit.

We'll dive into the implementation part of this article soon, but first it's important to establish what we're aiming to solve. Broadly, stock market analysis is divided into two parts – Fundamental Analysis and Technical Analysis.

- Fundamental Analysis involves analyzing the company's future profitability on the basis of its current business environment and financial performance.

➢ Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market.

As you might have guessed, our focus will be on the technical analysis part. We'll be using a dataset from data-flair. tranings (you can find historical data for various stocks here) and for this, I have used the data for 'Tata Global Beverages'.

Another difficulty we had to face was the way to determine winnings. We had two different options.

1. Winnings is the difference between portfolio value plus the capital we have on our possession and the initial capital.
2. Winnings is the sum of all the differences in price between sequential transactions.

For example if we bought a stock at price x and sold it at price y, then the winnings are y-x.

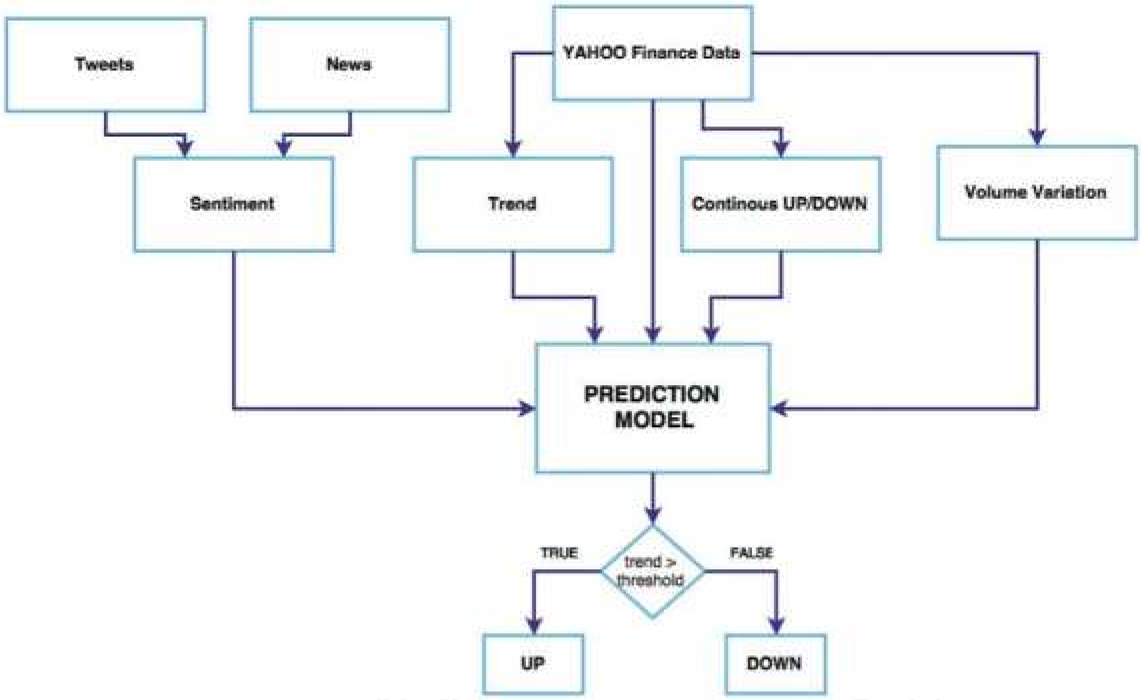

Fig 4: Data Flow

6.2. IMPLEMENTATION

We will first load the dataset and define the target variable for the problem.

Datasets - We will implement this technique on our dataset. The first step is to create a data-frame that contains only the *Date* and *Close* price columns, then split it into train and validation sets to verify our predictions.

```
df= pd.read_csv('NSE-Tata-Global-Beverages-Limited.csv')
df.head(10)
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05
5	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914.0	7162.35
6	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859.0	11859.95
7	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909.0	5248.60
8	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368.0	5503.90
9	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509.0	7999.55

Fig 5: Tata Global Dataset

There are multiple variables in the dataset – date, open, high, low, last, close, total_trade_quantity, and turnover.

- The columns Open and Close represent the starting and final price at which the stock is traded on a particular day.
- High, Low and Last represent the maximum, minimum, and last price of the share for the day.
- Total Trade Quantity is the number of shares bought or sold in the day and Turnover (Lacs) is the turnover of the particular company on a given date.

Another important thing to note is that the market is closed on weekends and public holidays. Notice the above table again, some date values are missing – 2/10/2018, 6/10/2018, 7/10/2018. Of these dates, 2nd is a national holiday while 6th and 7th fall on a weekend. The profit or loss calculation is usually determined by the closing price of a stock for the day; hence we will consider the closing price as the target variable.

6.2.1. MOVING AVERAGE:

In stock market analysis, a 50 or 200-day moving average is most commonly used to see trends in the stock market and indicate where stocks are headed. The MA is used in trading as a simple technical analysis tool that helps determine price data by customizing average price. There are many advantages in using a moving average in trading that can be tailored to any time frame. Depending on what information you want to find out, there are different types of moving averages to use.

The MA is the calculated average of any subset of numbers, using a technique to get an overall idea of the trends in a data set. Once you understand the MA formula, you can start to calculate any subsets to get your MA. It can be calculated for any period of time, making it extremely useful to forecast both long and short-term trends.

Fig 6: Moving Average (SMA 20)

The SMA formula is calculated by taking the average closing price of a security over any period desired. To calculate a moving average formula, the total closing price is divided by the number of periods.

For example, if the last five closing prices are:

28.93+28.48 +28.44+28.91+28.48 = 143.24

The five-day SMA is: 142.24/5= 28.65.

6.2.2. ARIMA MODEL:

We can de-trend the model by differencing each value from a value in the past and modeling these differences. Later, adding the value from the past to arrive at the actual value. We can choose to difference each value from a value at t-1 or t-2 or t-3 … Upon experimentation I found t-2 to give good results. By good results I mean the stationarity of the resulting time series was better. Again, to know what is stationarity and how to measure it please go through the link in the prerequisite section.

Fig 7: Differenced Close Price

Auto Regress or (p) **I**ntegrated (d) **M**oving **A**verage(q).

- p — Number of previous values to consider for estimating the current value
- d — n_diff in the previous code snippet
- q — If we consider a moving average to estimate each value, then q indicates the number of previous errors. i.e., if q= 3 then we will consider e(t-3), e(t-2) and e(t-1) as inputs of the regressor.
- Where e(i) = moving_average(i)- actual_value(i)

6.2.3. LONG-SHORT TERM MEMORY (LSTM):

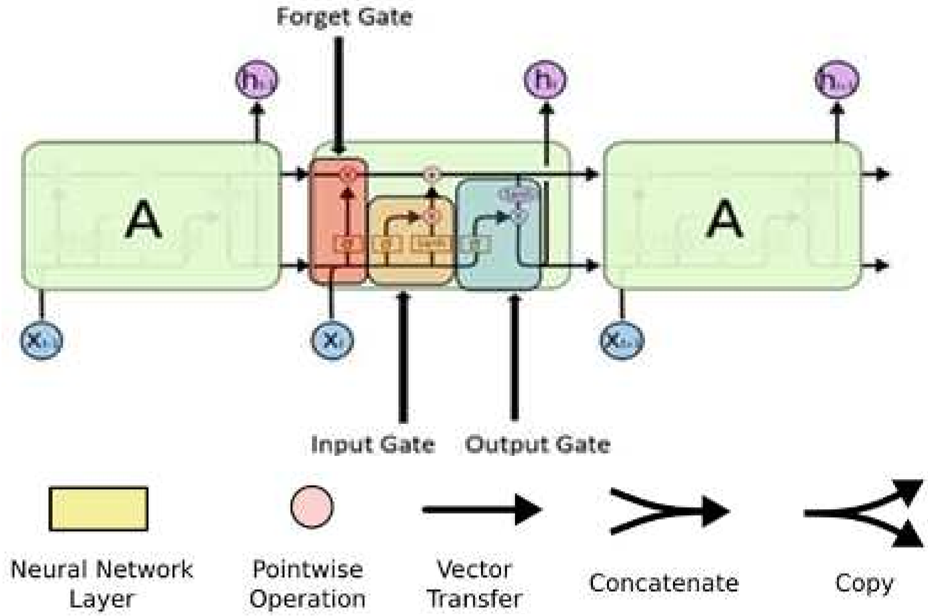

Fig 8: LSTM Cells

The first step in our LSTM is to decide the information to throw away from the cell state. The decision is made by a sigmoid layer called the "forget gate layer."

The next step is to decide what new information is going to store in the cell state. It has two parts. First, a sigmoid layer called the "input gate layer". It decides which values will be updated. Next, a tanh layer creates a vector of new candidate values, C~t C~t, that could be added to the state.

Finally, output is decided and it'll be based on our cell state, but will be a filtered version.

7. Coding

7.1. Stock Price prediction using Moving Average

- ```
  #import dataset
  from google.colab import files
  uploaded = files.upload()
  ```
  ```

```
#import packages
import pandas as pd
import numpy as np

#to plot within notebook
import matplotlib.pyplot as plt
%matplotlib inline

#setting figure size
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 20,10

#for normalizing data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))

#read the file
df = pd.read_csv('NSE-Tata-Global-Beverages-Limited.csv')

#print the head
df.head()
```

```
#setting index as date
df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')
df.index = df['Date']

#plot
plt.figure(figsize=(16,8))
plt.plot(df['Close'], label='Close Price history')
```

```
# importing libraries
import pandas as pd
import numpy as np

# reading the data
df = pd.read_csv('NSE-Tata-Global-Beverages-Limited.csv')

# looking at the first five rows of the data
print(df.head())
print('\n Shape of the data:')
print(df.shape)

# setting the index as date
df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')
df.index = df['Date']

#creating dataframe with date and the target variable
data = df.sort_index(ascending=True, axis=0)
new_data = pd.DataFrame(index=range(0,len(df)),columns=['Date', 'Close'])

for i in range(0,len(data)):
new_data['Date'][i] = data['Date'][i]
new_data['Close'][i] = data['Close'][i]

# NOTE: While splitting the data into train and validation set, we cannot use random splitting since t
hat will destroy the time component. So here we have set the last year's data into validation and the 4
 years' data before that into train set.
# splitting into train and validation
train = new_data[:987]
valid = new_data[987:]

# shapes of training set
print('\n Shape of training set:')
print(train.shape)
```

```
# shapes of validation set
print('\n Shape of validation set:')
print(valid.shape)

# In the next step, we will create predictions for the validation set and check the RMSE using the act ual values.
# making predictions

preds = []
for i in range(0,valid.shape[0]):
a = train['Close'][len(train)-248+i:].sum() + sum(preds)
b = a/248
preds.append(b)

# checking the results (RMSE value)
rms=np.sqrt(np.mean(np.power((np.array(valid['Close'])-preds),2)))
print('\n RMSE value on validation set:')
print(rms)
```

➢
```
#plot the graph
valid['Predictions'] = 0
valid['Predictions'] = preds
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
```

7.2. Stock Price prediction using LSTM :

➢
```
#import dataset
from google.colab import files
uploaded = files.upload()
```

➢
```
#import packages
import pandas as pd
import numpy as np

#to plot within notebook
import matplotlib.pyplot as plt
%matplotlib inline

#setting figure size
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 20,10

#for normalizing data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))

#read the file
df = pd.read_csv('NSE-Tata-Global-Beverages-Limited.csv')

#print the head
df.head()
```

➢
```
#setting index as date
df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')
df.index = df['Date']

#plot
plt.figure(figsize=(16,8))
plt.plot(df['Close'], label='Close Price history')
```

➢
```
#importing required libraries
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM
```

```
#creating dataframe
data = df.sort_index(ascending=True, axis=0)
new_data = pd.DataFrame(index=range(0,len(df)),columns=['Date', 'Close'])
for i in range(0,len(data)):
new_data['Date'][i] = data['Date'][i]
new_data['Close'][i] = data['Close'][i]

#setting index
new_data.index = new_data.Date
new_data.drop('Date', axis=1, inplace=True)

#creating train and test sets
dataset = new_data.values

train = dataset[0:987,:]
valid = dataset[987:,:]

#converting dataset into x_train and y_train
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(dataset)

x_train, y_train = [], []
for i in range(60,len(train)):
x_train.append(scaled_data[i-60:i,0])
y_train.append(scaled_data[i,0])
x_train, y_train = np.array(x_train), np.array(y_train)

x_train = np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))

# create and fit the LSTM network
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1],1)))
model.add(LSTM(units=50))
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x_train, y_train, epochs=1, batch_size=1, verbose=2)

#predicting 246 values, using past 60 from the train data
inputs = new_data[len(new_data) - len(valid) - 60:].values
inputs = inputs.reshape(-1,1)
inputs  = scaler.transform(inputs)

X_test = []
for i in range(60,inputs.shape[0]):
X_test.append(inputs[i-60:i,0])
X_test = np.array(X_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
closing_price = model.predict(X_test)
closing_price = scaler.inverse_transform(closing_price)
```

➢
```
#calculating RMS value
rms=np.sqrt(np.mean(np.power((valid-closing_price),2)))
rms
```

➢
```
#for plotting
train = new_data[:987]
valid = new_data[987:]
valid['Predictions'] = closing_price
plt.plot(train['Close'])
plt.plot(valid[['Close','Predictions']])
```

7.3. Stock Price prediction using ARIMA Model:

- ```
  #import dataset
  from google.colab import files
  uploaded = files.upload()
  ```
- ```
  #import packages
  import pandas as pd
  import numpy as np

  #to plot within notebook
  import matplotlib.pyplot as plt
  %matplotlib inline

  #setting figure size
  from matplotlib.pylab import rcParams
  rcParams['figure.figsize'] = 20,10

  #for normalizing data
  from sklearn.preprocessing import MinMaxScaler
  scaler = MinMaxScaler(feature_range=(0, 1))

  #read the file
  df = pd.read_csv('NSE-Tata-Global-Beverages-Limited.csv')

  #print the head
  df.head()
  ```
- ```
  #setting index as date
  df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')
  df.index = df['Date']

  #plot
  plt.figure(figsize=(16,8))
  plt.plot(df['Close'], label='Close Price history')
  ```
- ```
  #installing new lib files
  pip install pyramid
  pip install pmdarima
  ```
- ```
  #import package and setting the training
  from pmdarima import auto_arima

  data = df.sort_index(ascending=True, axis=0)

  train = data[:987]
  valid = data[987:]

  training = train['Close']
  validation = valid['Close']

  model = auto_arima(training, start_p=1, start_q=1,max_p=3, max_q=3, m=12,start_P=0, seasonal=T
  rue,d=1, D=1, trace=True,error_action='ignore',suppress_warnings=True)
  model.fit(training)

  forecast = model.predict(n_periods=248)
  forecast = pd.DataFrame(forecast,index = valid.index,columns=['Prediction'])
  ```
- ```
  #calculating RMS value
  rms=np.sqrt(np.mean(np.power((np.array(valid['Close'])-np.array(forecast['Prediction'])),2)))
  rms
  ```
- ```
  #plot the graph
  plt.plot(train['Close'])
  plt.plot(valid['Close'])
  plt.plot(forecast['Prediction'])
  ```
  ```

8. Results

8.1. Moving Average Predicted Price

Let's visualize this to get a more intuitive understanding. So here is a plot of the predicted values along with the actual values.

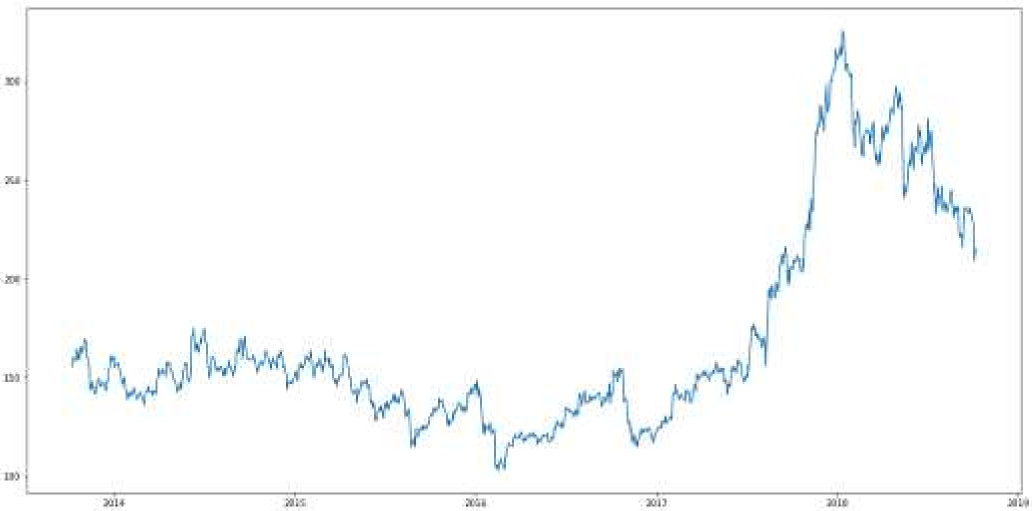

Fig 9: Close price (MA)

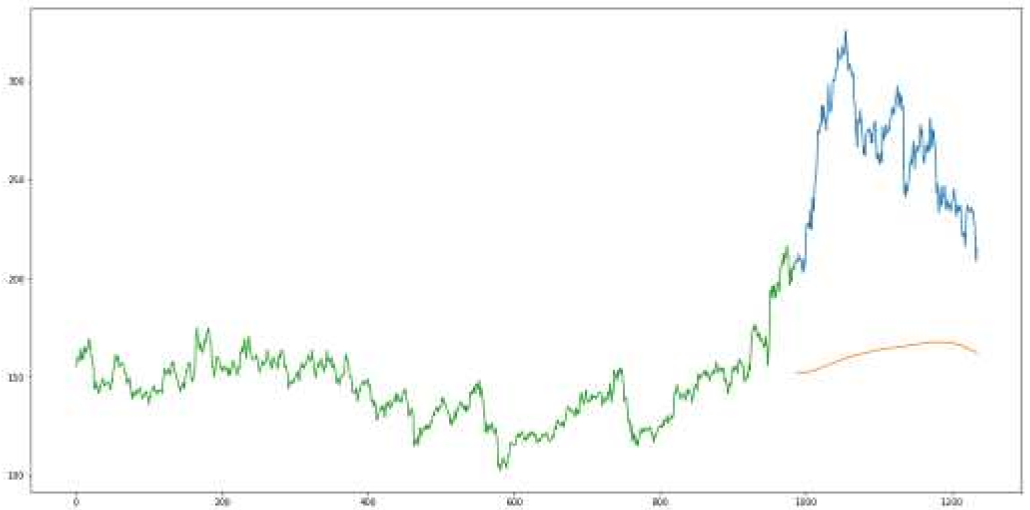

Fig 10: Predicted Price (MA)

8.2. ARIMA Model Predicted Price

ARIMA model uses past data to understand the pattern in the time series. Using these values, the model captured an increasing trend in the series. Although the predictions using this technique are far better than that of the previously implemented machine learning models. As it's evident from the plot, the model has captured a trend in the series, but does not focus on the seasonal part.

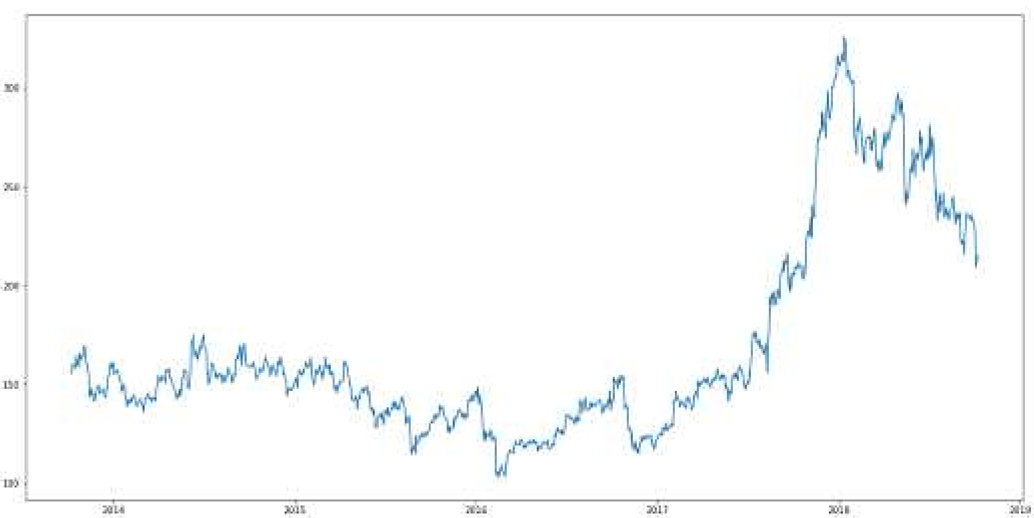

Fig 11: Close Price (ARIMA)

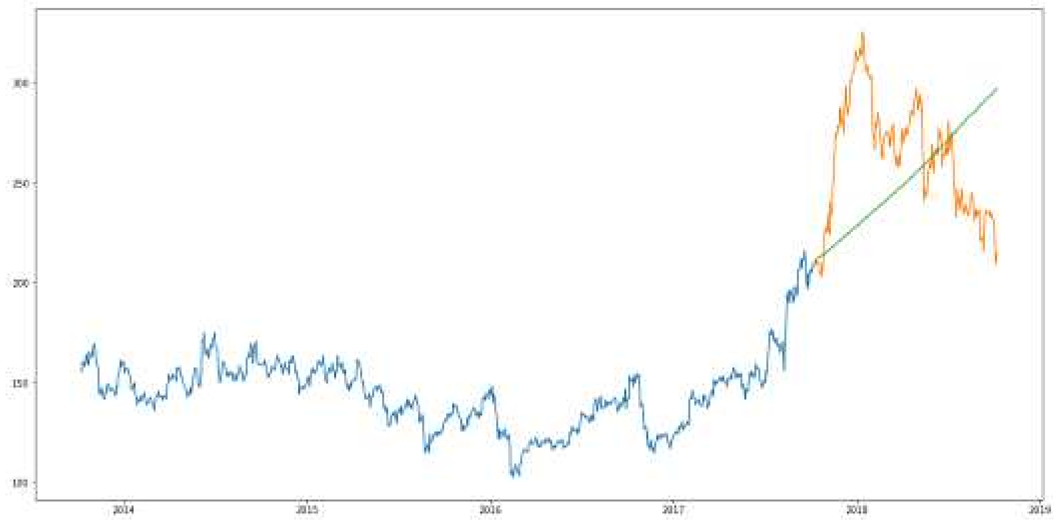

Fig 12: Predicted Price (ARIMA)

8.3. LSTM (Long-Short Term Memory) Predicted Price

The LSTM model can be tuned for various parameters such as changing the number of LSTM layers, adding dropout value or increasing the number of epochs. At the start of the article, stock price is affected by the news about the company and other factors like demonetization or merger/demerger of the companies. There are certain intangible factors as well which can often be impossible to predict beforehand.

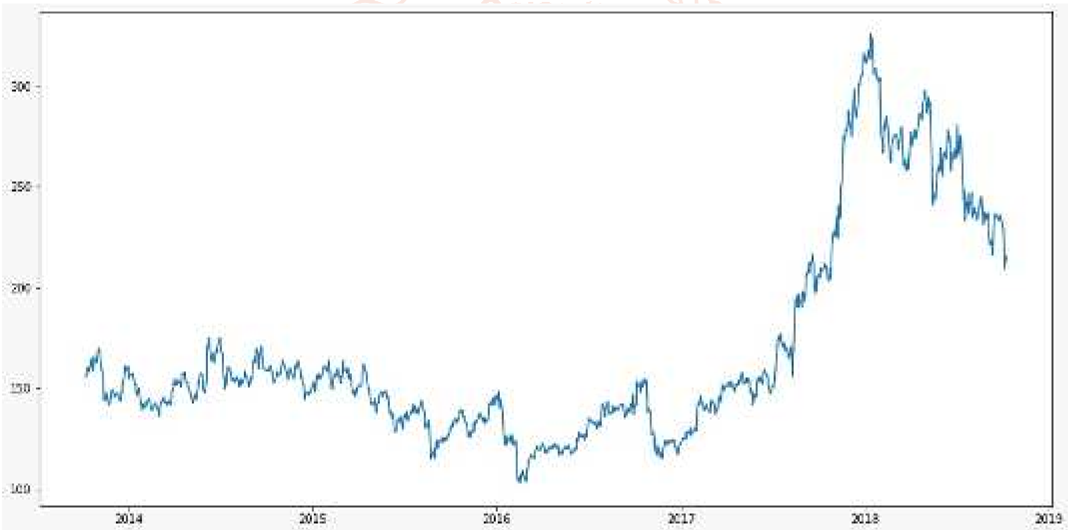

Fig 13: Close Price (LSTM)

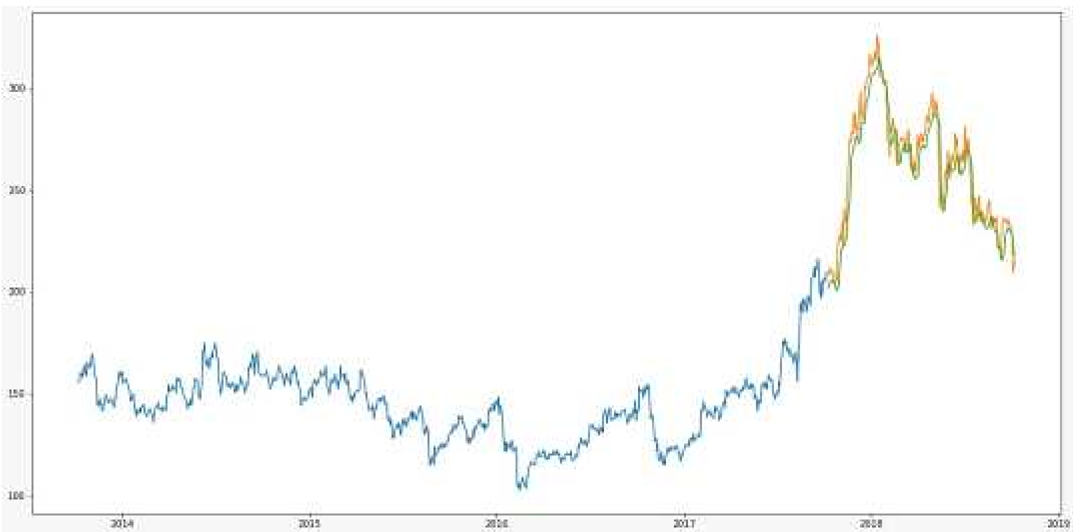

Fig 14: Predicted Price (LSTM)

8.4. Comparative Analysis of Machine Learning Techniques:

S. No.	Techniques	Advantages	Disadvantages	Parameter Used
1	Moving Average	The SMA is the most straightforward calculation, the average price over a chosen time period.	The SMA's weakness is that it is slower to respond to rapid price changes that often occur at market reversal points.	Close Price of stocks.
2	ARIMA MODEL	ARIMA works better for relatively short series when the number of observations is not sufficient to apply more flexible methods.	It is suitable for short-term prediction only.	Open, High, Close, Low prices and moving average.
3	LSTM(Long-Short Term Memory)	LSTM cells have a memory that can store previous time step information and use it to train the dataset. It has the ability to bridge very long time lags.	They require a lot of resources and time to get trained and become ready for real-world applications.	Open, High, Close, Low prices.

8.5. RMSE : Root-Mean-Square Error

RMSE is defined as the square root of the average squared distance between the actual score and the predicted

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

RMSE is used to evaluate the Machine Learning models. So here,
Moving average – 104.51415465984348
ARIMA model – 44.954584993246954
LSTM – 9.707045241044804

9. Conclusion

This work summarizes necessary techniques in machine learning that square measure relevant to stock prediction. This model may be improved upon by process refined fuzzy rules. By coaching data's scale and timeframe may result in higher prediction. Technical indicators square measure accustomed construct the relation between stock exchange index and their variables. Implementing victimization Moving Average slow so as to perform computations compared to alternative techniques, whereas ARIMA model is good for short-run prediction. Every technique has its own benefits and downsides. Differing types of techniques are accustomed predict the stock exchange and to forecast the longer term stock values up to some extent. Combining Moving Average, ARIMA Model and LSTM Model might end in high accuracy.

9.1. General Comment

We were able to come with a way to successfully predict the stock market and in combination with a good trading strategy we were able to profit from stock trading using historical data. The reason we used historical data and not real time data for testing was time efficiency but also the ability to compare models and trading strategies using the same testing data.

How would the model behave with real-time data?

We treated our historical data as real time data. We can use the same methods and be able to predict the stock price in real time. We can achieve this by collecting the transactions in real time and converting them into 5-min intervals and just passing them forward to our network, point by point. One of the goals of this thesis was that we should be able to utilize the stock market on real time and all the simulations were done in way that would make it easy to transition from historical to real time data.

As an investment it can be characterized as really profitable. However the neural network cannot predict sudden changes in the price that happen during the time that the stock market is closed. An example is when a company or their direct competitors announce their term results. Those kinds of events can skyrocket the stock price or make it lose considerable value.

9.2. Limitations

The main limitation we had during this thesis was the time constraints due to the lack of available data. We had to wait at least three months to run experiments as we had to collect our own data. Also trying to train a neural network takes quite long. If we were lucky a neural network would converge in a couple of hours but sometimes could take up to 15 hours. Especially in the Reinforcement Learning (RL) models training would take quite a few days. The whole process it was a trial and error in order to come up with the optimal hyper parameters and we had to train numeral different networks just to be able to select them.

Stock market cannot be accurately predicted. The future, like any complex problem, has far too many variables to be predicted. The stock market is a place where buyers and sellers converge. When there are more buyers than sellers, the price increases

9.3. Future Work

We have to test the existing methods with more data as the keep coming. We want to ensure that the results we got it is not just a random event that happened as result of the time period. We have to test with even more data as time passes and make sure that our model can generalize. Another thing we can do is check how our model works with stocks outside the OMSX30 index. We can try train and evaluate out model with stocks that belong to smaller companies that do not have as many transactions as the big ones. In that case we will be able to see if we can expand our work to other stocks and maybe even other stock markets as well.

The concept behind this idea is that we will have few neural networks trained in different time intervals. For example, we can have the prediction of the stock price in 5-min, 10-min and 30-min intervals. Using this information we can decide when is the best time to place our action. If we know how the stock will move within the next thirty minutes we will be able to increase our profit even more. The more information, we have the more profit we can achieve.

REFERENCES

[1] Prashant S. Chavan,"Parameters for Stock Market Prediction" IJCTA, MAR 2013.

[2] Mayank Kumar B Patel, Sunil R Yalamalle "Stock price prediction using Artificial Neural Network "IJIRSET, June 2014.

[3] Mrityunjay Sharma "Survey on Stock Market Prediction and Performance Analysis" IJARCET, January 2014.

[4] Gholap Rahul Mansing "Indian stock market prediction using neural network technique" IJARCET, March 2014.

[5] Shom Prasad Das, Sudarsan Padhy "Support Vector Machines for Prediction of Futures Prices in Indian Stock Market" IJCA, March 2014.

[6] Author: Yusuf Perwej, Asif Perwej Research paper: Prediction of the Bombay Stock Exchange (BSE) Market Returns Using Artificial Neural Network and Genetic Algorithm Journal: Scientific Research

[7] Author: K. Senthamarai Kannan, P. Sailapathi Sekar, M. Mohamed Sathik and P. Arumugam Research paper: Financial Stock Market Forecast using Data Mining Techniques Journal: International Multi-Conference of Engineers and Computer Scientists 2010 Vol I, IMECS 2010, March 17-19, 2010, Hong Kong. ISSN:2078-0966

[8] Author: Zahid Iqbal, R. Ilyas, W. Shahzad, Z. Mahmood and J Anjum Research paper: Efficient Machine Learning Techniques for Stock Market Prediction Journal: Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp.855-867.

[9] Author: Marc-André Mittermaye Research paper: Forecasting Intraday Stock Price Trends with Text Mining Techniques Journal: Hawaii International Conference on System Sciences – 2004.

[10] Hellstrom, T. and Holmstrom, K. 1998, 'Predicting the Stock Market', Technical Report Series Ima-TOM-1997-07.

[11] M. Usmani, S. H. Adil, K. Raza and S. S. A. Ali, "Stock market prediction using machine learning techniques," 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2016, pp. 322-327.