Global Scientific JOURNALS
GSJ: Volume 10, Issue 1, January 2022, Online: ISSN 2320-9186
www.globalscientificjournal.com

STOCK MARKET PREDICTION

Saumya tripathi

TABLES OF CONTENTS

1. Abstract:

- Time series forecasting has been widely used to determine the future prices of stock, and the analysis and modeling of finance time series importantly guide investors' decisions and trades.

- This work proposes an intelligent time series prediction system that uses sliding-window optimization for the purpose of predicting the stock prices.

- The system has a graphical user interface and functions as a stand-alone application.

- The proposed model is a promising predictive technique for highly non-linear time series, whose patterns are difficult to capture by traditional models.

2. Introduction:

- Stock market prediction has been an active area of research for a long time. The Efficient Market Hypothesis (EMH) states that stock market prices are largely driven by new information and follow a random walk pattern.

- In this project, we test a hypothesis based on the premise of behave economics, that the emotions and moods of individuals affect their decision making process, thus, leading to a direct correlation between public sentiment and market sentiment.

- Financial markets are highly volatile and generate huge amounts of data daily.

- It is the most popular financial market instrument and its value changes quickly.

- Stock prices are predicted to determine the future value of companies' stock or other financial instruments that are marketed on financial exchanges.

- However, the stock market is influenced by many factors such as political events, economic conditions and traders' expectation.

What is Machine Learning:

- Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.

- More formally, it can defined as,

  - A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

  - Example: playing checkers.

    E = the experience of playing many games of checkers

    T = the task of playing checkers.

    P = the probability that the program will win the next game.

What is Deep Learning

- Deep Learning is an Artificial Intelligence function that imitates the working of the human brain in processing data and creating patterns for use in decision making.

- Deep Learning is a subset of Machine Learning in Artificial Intelligence that has networks capable of learning un-supervised from data i.e., unstructured or unlabeled.

- It is also known as Deep Neural Network or Deep Neural Learning that can be used in LSTM technique to analyze the raw data and to plot predicted result.

LSTM:

- Long Short Term Memory

- LSTM units or blocks are part of a recurrent neural network in deep learning.

- These artificial intelligence programs to more effectively imitate human thought.

3.EXISTING METHOD:

- Time series forecasting consists of a research area designed to solve various problems, mainly in the financial area.

- Support vector regression (SVR), a variant of the SVM, is typically used to solve nonlinear regression problems by constructing the input-output mapping function.

- Long Short Term Memory(LSTM) units or blocks are part of a recurrent neural network in deep learning.

- The Firefly Algorithm (FA), which is a nature-inspired met heuristic method, has recently performed extremely well in solving various optimization problems.

Disadvantages:

- The existing system focuses on the stock price market in Taiwan, but does not generalize for other markets worldwide.
- The system does not allow the import of raw data directly
- The existing system cannot be used to analyze multi-variate time series.
- Lastly, the system does not have a user-interface which can be distributed as a web app to users for personal use.

4.Proposed System:

- To generalize the application of the existing system, our work uses the system to estimate other stocks in similar emerging markets and mature markets.

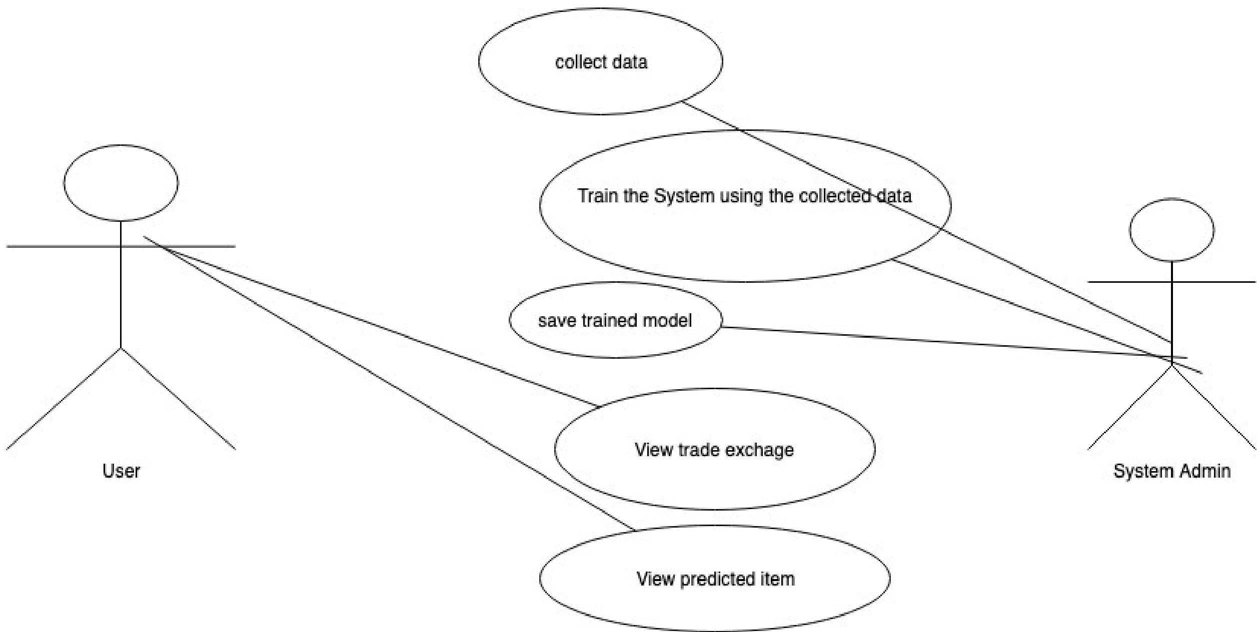
- The system can be extended to analyze multivariate time series data and import raw dataset directly.

- Profit can be maximized even when the corporate stock market is has lower value.

- The development of a web-based application has been considered to improve the user-friendliness and usability of the expert system.

5.System Design:

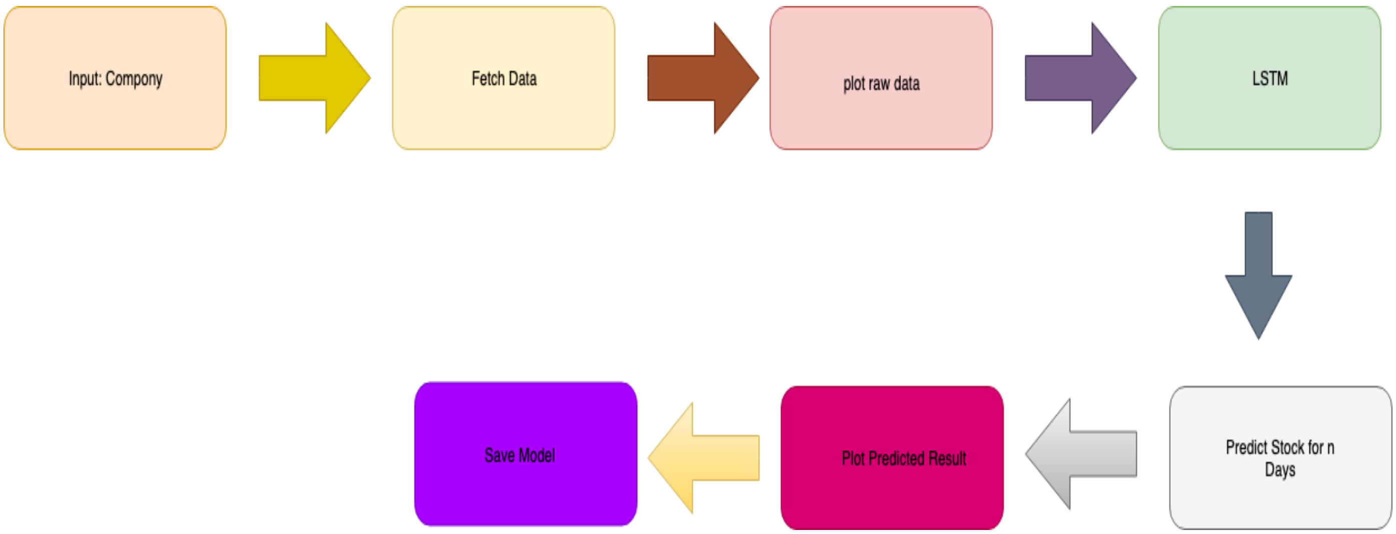
Use case diagram:



Stock market prediction using machine learning

- Data is initially collected from online sources or the stock exchange.
- The data is then used to train the system.
- Trained model is saved.
- User view the trained exchange and stock of companies.
- Using the model, closing prices are predicted.

Data flow diagram:



Stock market prediction using machine learning

6. Implementation:

6.1 program/Algorithm:

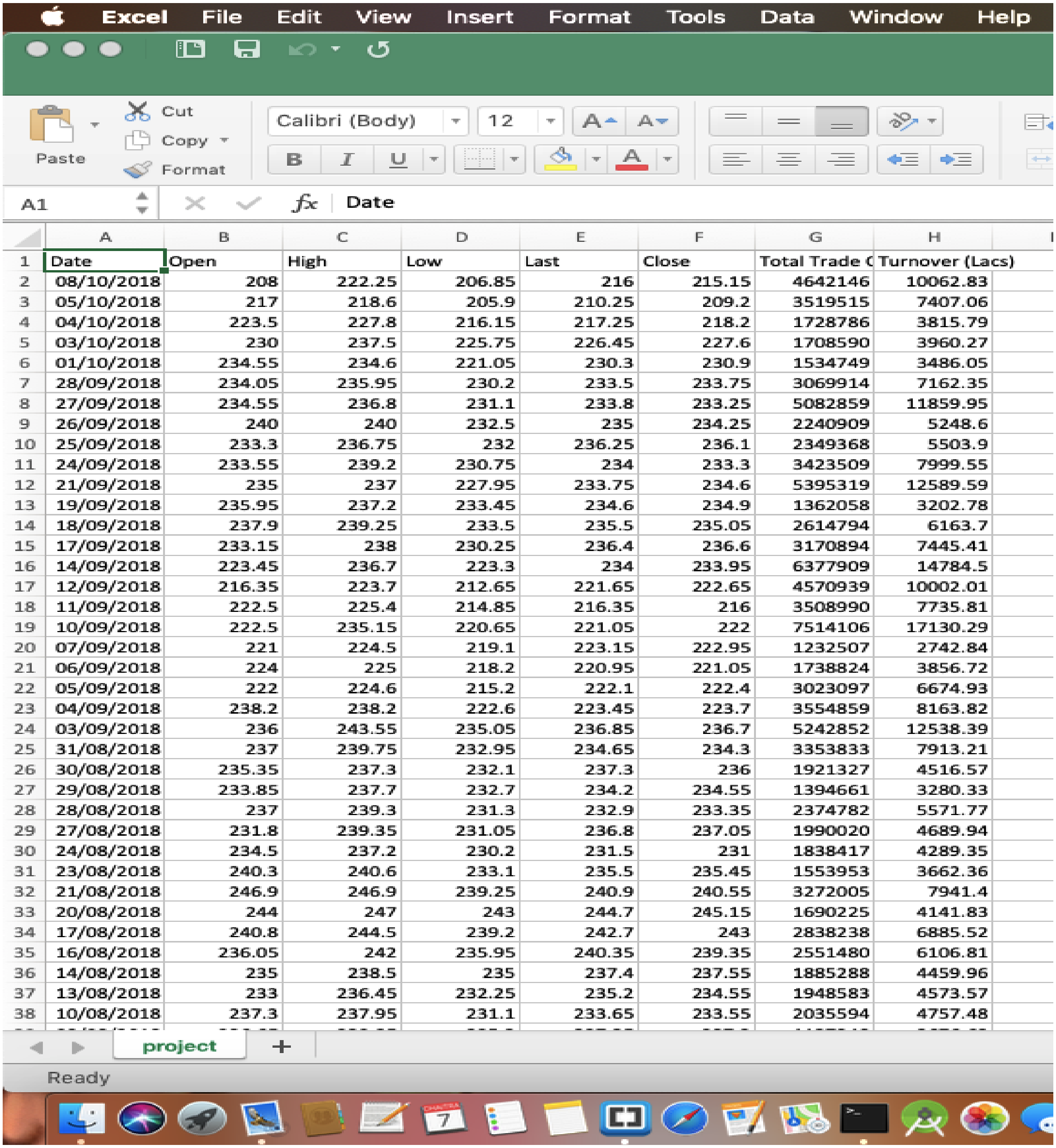
```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM



#to plot within notebook
import matplotlib.pyplot as plt



#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('project.csv')
```

```
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]

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 inrange(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)
rms=np.sqrt(np.mean(np.power((valid-closing_price),2)))
print(rms)
```

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

6.2. Raw Data Set:



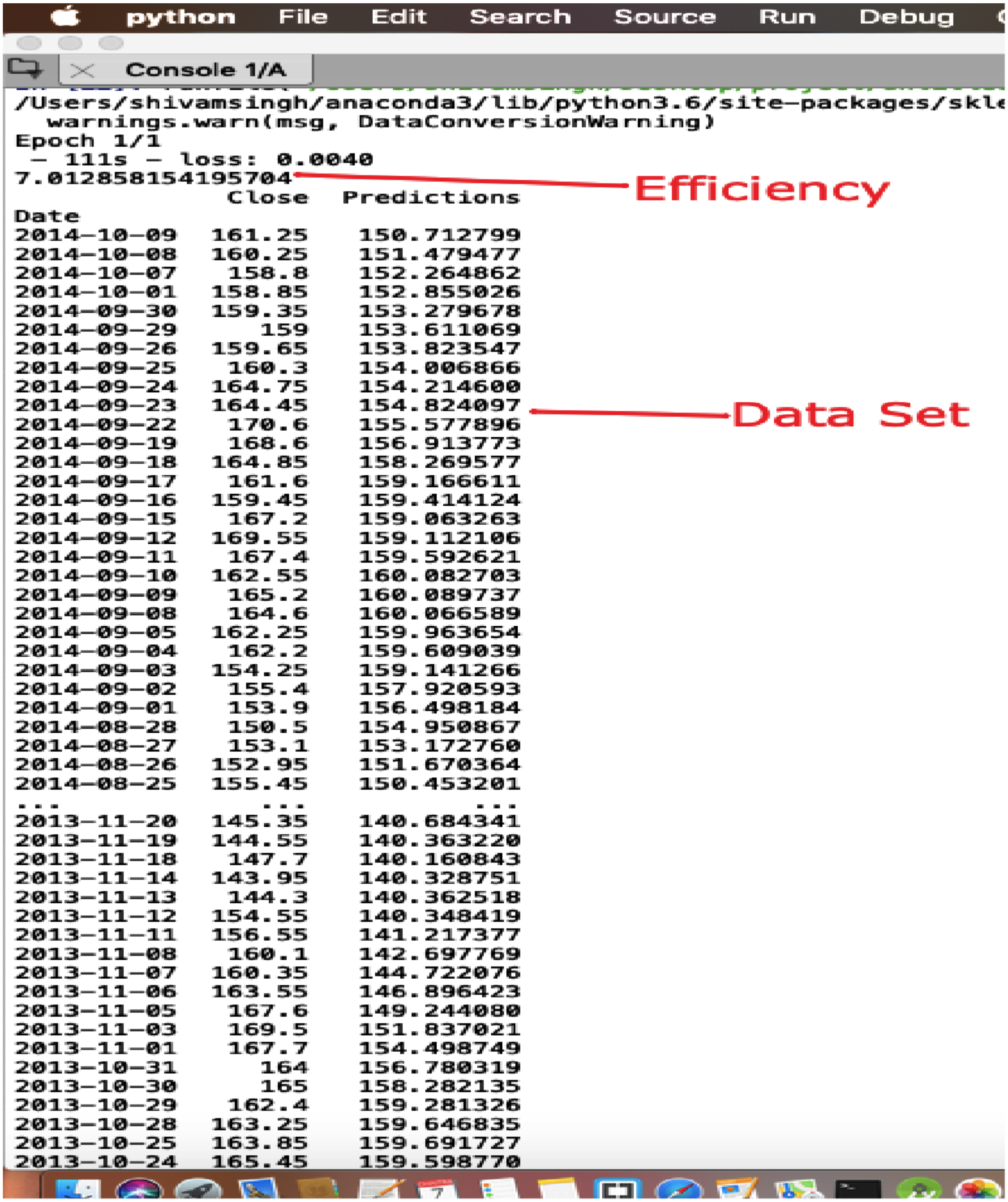
Date	Open	High	Low	Last	Close	Total Trade (	Turnover (Lacs)
08/10/2018	208	222.25	206.85	216	215.15	4642146	10062.83
05/10/2018	217	218.6	205.9	210.25	209.2	3519515	7407.06
04/10/2018	223.5	227.8	216.15	217.25	218.2	1728786	3815.79
03/10/2018	230	237.5	225.75	226.45	227.6	1708590	3960.27
01/10/2018	234.55	234.6	221.05	230.3	230.9	1534749	3486.05
28/09/2018	234.05	235.95	230.2	233.5	233.75	3069914	7162.35
27/09/2018	234.55	236.8	231.1	233.8	233.25	5082859	11859.95
26/09/2018	240	240	232.5	235	234.25	2240909	5248.6
25/09/2018	233.3	236.75	232	236.25	236.1	2349368	5503.9
24/09/2018	233.55	239.2	230.75	234	233.3	3423509	7999.55
21/09/2018	235	237	227.95	233.75	234.6	5395319	12589.59
19/09/2018	235.95	237.2	233.45	234.6	234.9	1362058	3202.78
18/09/2018	237.9	239.25	233.5	235.5	235.05	2614794	6163.7
17/09/2018	233.15	238	230.25	236.4	236.6	3170894	7445.41
14/09/2018	223.45	236.7	223.3	234	233.95	6377909	14784.5
12/09/2018	216.35	223.7	212.65	221.65	222.65	4570939	10002.01
11/09/2018	222.5	225.4	214.85	216.35	216	3508990	7735.81
10/09/2018	222.5	235.15	220.65	221.05	222	7514106	17130.29
07/09/2018	221	224.5	219.1	223.15	222.95	1232507	2742.84
06/09/2018	224	225	218.2	220.95	221.05	1738824	3856.72
05/09/2018	222	224.6	215.2	222.1	222.4	3023097	6674.93
04/09/2018	238.2	238.2	222.6	223.45	223.7	3554859	8163.82
03/09/2018	236	243.55	235.05	236.85	236.7	5242852	12538.39
31/08/2018	237	239.75	232.95	234.65	234.3	3353833	7913.21
30/08/2018	235.35	237.3	232.1	237.3	236	1921327	4516.57
29/08/2018	233.85	237.7	232.7	234.2	234.55	1394661	3280.33
28/08/2018	237	239.3	231.3	232.9	233.35	2374782	5571.77
27/08/2018	231.8	239.35	231.05	236.8	237.05	1990020	4689.94
24/08/2018	234.5	237.2	230.2	231.5	231	1838417	4289.35
23/08/2018	240.3	240.6	233.1	235.5	235.45	1553953	3662.36
21/08/2018	246.9	246.9	239.25	240.9	240.55	3272005	7941.4
20/08/2018	244	247	243	244.7	245.15	1690225	4141.83
17/08/2018	240.8	244.5	239.2	242.7	243	2838238	6885.52
16/08/2018	236.05	242	235.95	240.35	239.35	2551480	6106.81
14/08/2018	235	238.5	235	237.4	237.55	1885288	4459.96
13/08/2018	233	236.45	232.25	235.2	234.55	1948583	4573.57
10/08/2018	237.3	237.95	231.1	233.65	233.55	2035594	4757.48

7. Result and Discussion:

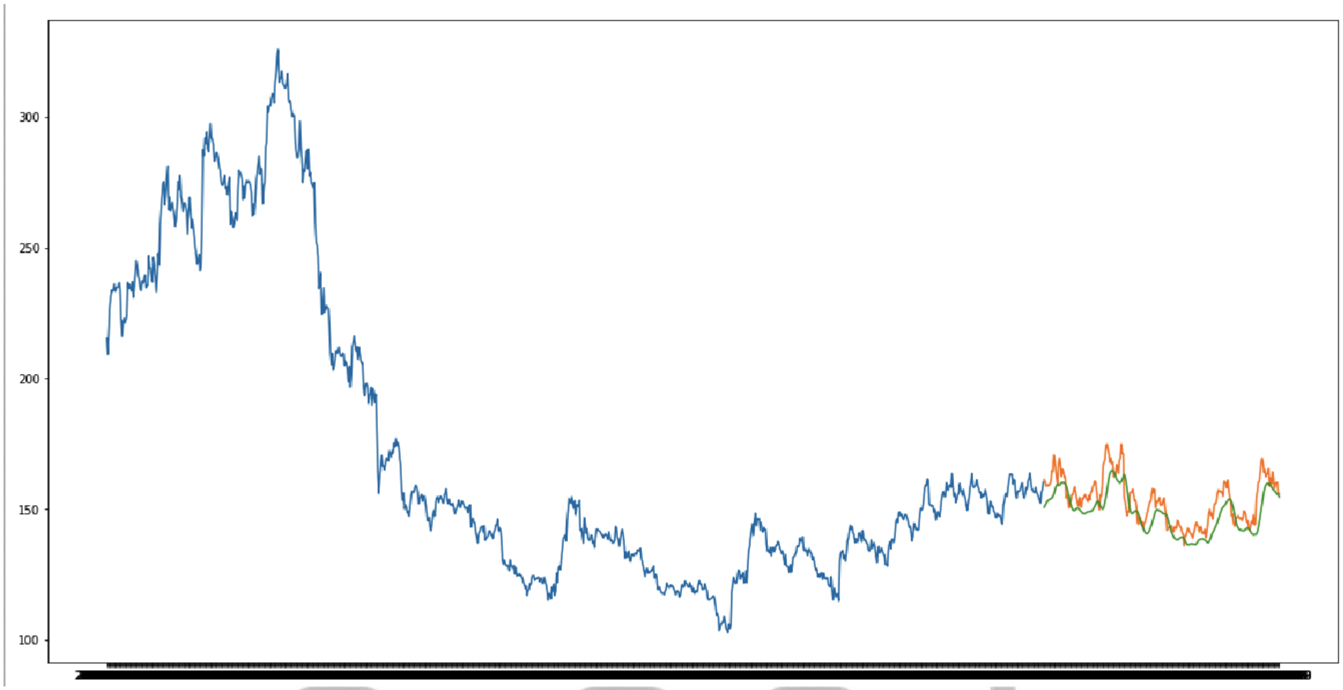
Average Close value, Average predicted value And Efficiency (in percentage) :

```
[248 rows x 2 columns]
average valid
153.5778225806452
average prediction
148.70277404785156
print percentage
96.82568195662905
/Users/shivamsingh/Desktop/project/untitled1.py:80: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```


Data After Prepressing And Root Mean Square Value :



Performance via Graph:



- In this graph blue lines are representing the "**Training Data**".

- Green lines representing the "**Actual Stock**" these are taken from last 246 rows from our data set file (project.csv).

- Yellow lines in the Graph represent the "**predicted Future Stock**".

- In this we got **Accuracy of 96.8** %.

8.Future Enhancement

- The limitation of the proposed system is its computational speed, especially with respect to sliding-window validation as the computational cost increases with the number of forward day predictions.

- The proposed model does not predict well for sudden changes in the trend of stock data.

- This occurs due to external factors and real-world changes affecting the stock market.

- We can overcome this by implementing Sentiment Analysis and Neural Networks to enhance the proposed model.

- We can modify the same system to an online-learning system that adapts in real-time.

9.Conclusion:

- Thus, as we can see above in our proposed method, we train the data using existing stock dataset that is available. We use this data to predict and forecast the stock price of n-days into the future.
- The average performance of the model decreases with increase in number of days, due to unpredictable changes in trend.
- The current system can update its training set as each day passes so as to detect newer trends and behave like an online-learning system that predicts stock in real-time.

8: REFERENCES –

www.draw.io,

www.google.com

www.wikipedia.org

www.udemy.com

www.datacamp.com

www.youtube.com

www.stackoverflow.com

https://anaconda.org/conda-forge/keras