SUMMER INTERNSHIP REPORT

Area Of Online Internship	**Machine learning And Deep Learning**
Intern Name	**Anchal Rathore**
Name Of Institution	**INDIAN INSTITUTE OF TECHNOLOGY,INDORE**
Faculty Mentor Name	**Dr. Vimal Bhatia**
Project Title	**Stock Prediction Using ML**
Duration	**1 month (from 16/06/2022 to 17/07/2022)**
Date of Submission	**31/08/2022**

Table Of Contents

Abstract

In Stock Market Prediction, the main aim is to predict the future value of the stocks of NSE-TATA GLOBAL. The trend in market prediction technologies is the use of machine learning which makes predictions based on the values of current stocks by training the data of previous values. Machine learning employs different models of training data with its efficient form.

Introduction

A correct prediction of stocks can lead to huge profits for the seller and the broker. Frequently, it is brought out that prediction is chaotic rather than random, which means it can be predicted by carefully analyzing the history of the respective stock market. The vital part of machine learning is the dataset used. The dataset should be as concrete as possible because a little change in the data can perpetuate massive changes in the outcomes they assessing a company's historical performance as well as the reliability of its accounts. This dataset comprises of following five variables: open, close, low, high, and volume. Open, close, low, and high are different bid prices for the stock at separate times with nearly direct names. The model is then tested on the test data. Regression and LSTM models are engaged for this conjecture separately. Regression involves minimizing error and LSTM contributes to remembering the data and results for the long run. Finally, the graphs for the fluctuation of prices with the dates (in the case of Regression based model) and between actual and predicted prices (for the LSTM-based model) are plotted.

Methodology

The stock market prediction seems a complex problem because there are many factors that have yet to be addressed and it doesn't seem statistical at first. But by proper use of machine learning techniques, one can relate previous data to the current data and train the machine to learn from it and make appropriate assumptions.

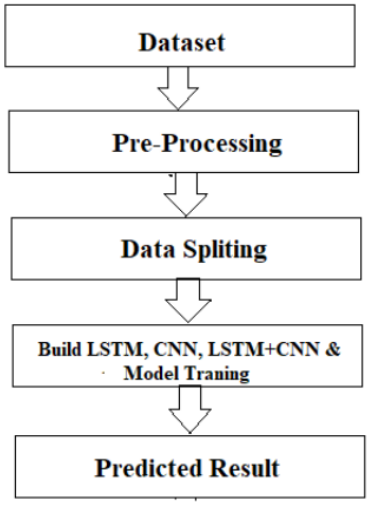


Fig 1. Proposed working

Working of the LSTM model:

Long Short-Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input within the present step. It tackled the matter of long-term dependencies of RNN within which the RNN will not predict the word hold on within the long term memory however can offer additional accurate forecasts from the recent info. Because the gap length will increases RNN does not offer an economical performance. LSTM will by default retain the knowledge for a long period of time. It is used for processing, predicting, and classifying on the basis of time-series data.

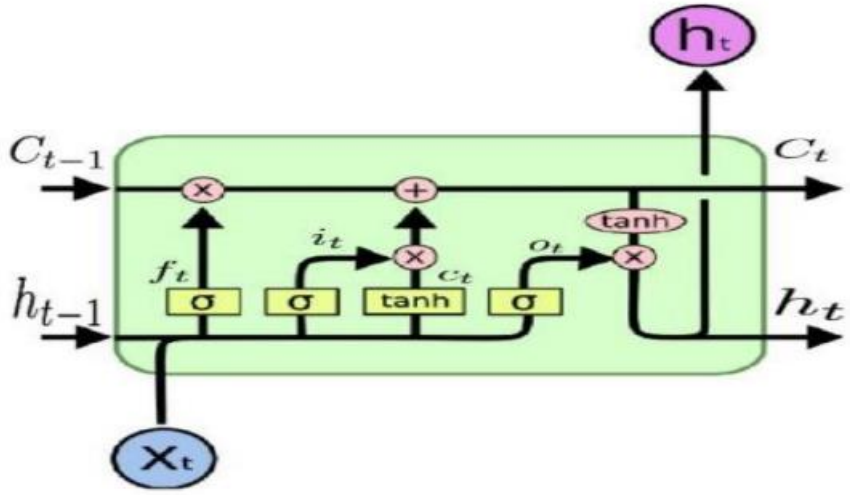


Fig. 2 LSTM Architecture

Working of CNN model:

In the hybrid approach, the Convolutional Neural Networks (CNNs) offer benefits in choosing sensible options, and Long Short-Term Memory (LSTM) networks have proven sensible skills to find out to learn sequential data. Each approach is reported to produce an improved result. CNN's possess convolute filters over every input layer so as to get simple options and CNNs have shown enhancements in computer vision, natural language processing, and different tasks. CNN may be a powerful tool to pick out features in order to improve the prediction accuracy.

Applications of CNN include:

- Decoding Facial Recognition
- Analyzing Documents

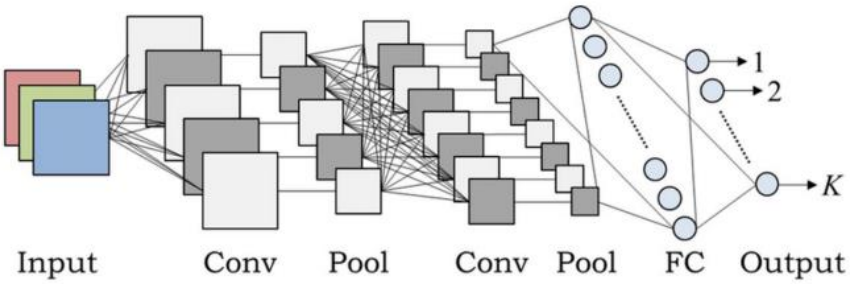


Fig. 3 Architecture of CNN Model

Regression

Regression is a technique for investigating the relationship between independent variables or features and a dependent variable or outcome. It's used as a method for predictive modeling in machine learning, in which an algorithm is used to predict continuous outcomes.

Some of the most common regression techniques in machine learning can be grouped into the following types of regression analysis:

- Simple Linear Regression
  Simple Linear regression is a linear regression technique that plots a straight line within data points to minimize the error between the line and the data points. It is one of the most simple and basic types of machine learning regression.

- Multiple linear regression
  Multiple linear regression is a technique used when more than one independent variable is used. Polynomial regression is an example of a multiple linear regression techniques.
- Logistic regression
  Logistic regression is used when the dependent variable can have one of two values, such as true or false, or success or failure. Logistic regression models can be used to predict the probability of a dependent variable occurring.

Regression Tree

A regression tree is basically a decision tree that is used for the task of regression which can be used to predict continuous valued outputs instead of discrete outputs.

Dataset Details

The dataset consists of the stock historical data from the National stock exchange (NSE) and captures the daily information of each stock from the National Stock Exchange. It collects data of NSE TATA GLOBAL which includes the opening price, the highest price, the lowest price, the closing price, the adjusted closing price, and the volume of stock.

Tool & Technologies used in the project:

Python:

The language of selection for this project was Python.

- Python [is an abundance of powerful tools ready for scientific computing Packages. The packages like NumPy, Pandas, and SciPy area units are freely available and well documented. These Packages will intensely scale back, and vary the code necessary to write a given program. This makes repetition fast.

NUMPY :

Numpy is a python package that provides scientific and higher-level mathematical abstractions wrapped in python. Numpy's array type augments the Python language with an efficient data structure used for numerical work, e.g., manipulating matrices.

SCIKIT LEARN:

Scikit-learn could be a free machine learning library for Python. It features numerous classification, clustering and regression algorithms like random forests, k-neighbours, support vector machine, and it furthermore supports Python scientific and numerical libraries like SciPy and NumPy.

COMPILER OPTION:

Anaconda is [19] free premium open-source distribution of the R and Python programming languages for scientific computing, predictive analytics, and large-scale process that aim is to modify package management and deployment. Package versions units are managed by the package management system code.

JUPITER NOTEBOOK :

The Jupyter Notebook is an open-source web application that enables to making and sharing of documents that contain visualizations, narrative text, live code, and equations. Uses include: data, data visualization, data transformation, statistical modeling, machine learning, numerical simulation, data cleaning, and much more.

Problem - Predict the stock market price of next few days using previous stock market data (equity or indices) using machine learning or Deep learning.

1. Use News headlines as Data for prediction.
2. Use previous Equity data of Day open, close, low, high for prediction.
3. Any other stock Relative data.

Solution Code

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
plt.style.use('bmh')

stock = pd.read_csv(r'NSE-TATAGLOBAL11.csv')
print(stock)

stock = pd.read_csv(r'NSE-TATAGLOBAL11.csv')
stock.head()

stock.shape

stock["Date"]=pd.to_datetime(stock.Date,format="%Y-%m-%d")
stock.index=stock['Date']
plt.figure(figsize=(16,8))
plt.title('TATAGLOBAL')
```

```
plt.xlabel('Days')
plt.ylabel('Close Price in Rs')
plt.plot(stock['Close'],label='Close Price')
plt.show()

stock = stock[['Close']]
stock.tail(4)
future_days = 25
stock['Prediction'] = stock[['Close']].shift(-future_days)
stock.head(25)

X = np.array(stock.drop(['Prediction'],1))[:-future_days]
print(X)
y= np.array(stock['Prediction'])[:-future_days]
print(y)

x_train , xtest, y_train, y_test = train_test_split(X, y, test_size = 0.25)
tree = DecisionTreeRegressor().fit(x_train,y_train)

lr = LinearRegression().fit(x_train,y_train)
x_future = stock.drop(['Prediction'],1)[:-future_days]

x_future = x_future.head(future_days)
x_future = np.array(x_future)

x_future
tree_prediction = tree.predict(x_future)
print(tree_prediction)
print()
```

```
lr_prediction = lr.predict(x_future)
print(lr_prediction)
stock[:future_days]
predictions = tree_prediction

valid = stock[:future_days]
valid['Predictions']= predictions

plt.figure(figsize=(16,8))
plt.title('Model')
plt.xlabel('Days',fontsize=18)
plt.ylabel('Close price in rs',fontsize=18)
plt.plot(stock['Close'])
plt.plot(valid[['Close','Predictions']])

plt.show()
predictions = lr_prediction

valid = stock[:future_days]
valid['Predictions']= predictions
plt.figure(figsize=(16,8))
plt.title('Model')
plt.xlabel('Days',fontsize=18)
plt.ylabel('Close price in rs',fontsize=18)
plt.plot(stock['Close'])
plt.plot(valid[['Close','Predictions']])
plt.legend(['orig','Val','Pred'])
plt.show()
```

Results

Step 1:Data analysis

```
            Date    Open    High     Low    Last   Close  \
0     2018-10-08  208.00  222.25  206.85  216.00  215.15
1     2018-10-05  217.00  218.60  205.90  210.25  209.20
2     2018-10-04  223.50  227.80  216.15  217.25  218.20
3     2018-10-03  230.00  237.50  225.75  226.45  227.60
4     2018-10-01  234.55  234.60  221.05  230.30  230.90
...          ...     ...     ...     ...     ...     ...
1230  2013-10-14  160.85  161.45  157.70  159.30  159.45
1231  2013-10-11  161.15  163.45  159.00  159.80  160.05
1232  2013-10-10  156.00  160.80  155.85  160.30  160.15
1233  2013-10-09  155.70  158.20  154.15  155.30  155.55
1234  2013-10-08  157.00  157.80  155.20  155.80  155.80

      Total Trade Quantity  Turnover (Lacs)
0                4642146.0         10062.83
1                3519515.0          7407.06
2                1728786.0          3815.79
3                1708590.0          3960.27
4                1534749.0          3486.05
...                    ...              ...
1230             1281419.0          2039.09
1231             1880046.0          3030.76
1232             3124853.0          4978.80
1233             2049580.0          3204.49
1234             1720413.0          2688.94

[1235 rows x 8 columns]
```

Fig. 4 Dataset information

Firstly, I have performed Data analysis for stock price of companies. Fig. represent the date, open, close, high, low, adjusted close and volume of stocks details.

Step 2: Read Dataset

	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

Fig. 5: Read Dataset

After performing data analysis, I have read the dataset. It shows the dataset information table starting from the head.

Step 3: Graph of Close Price history

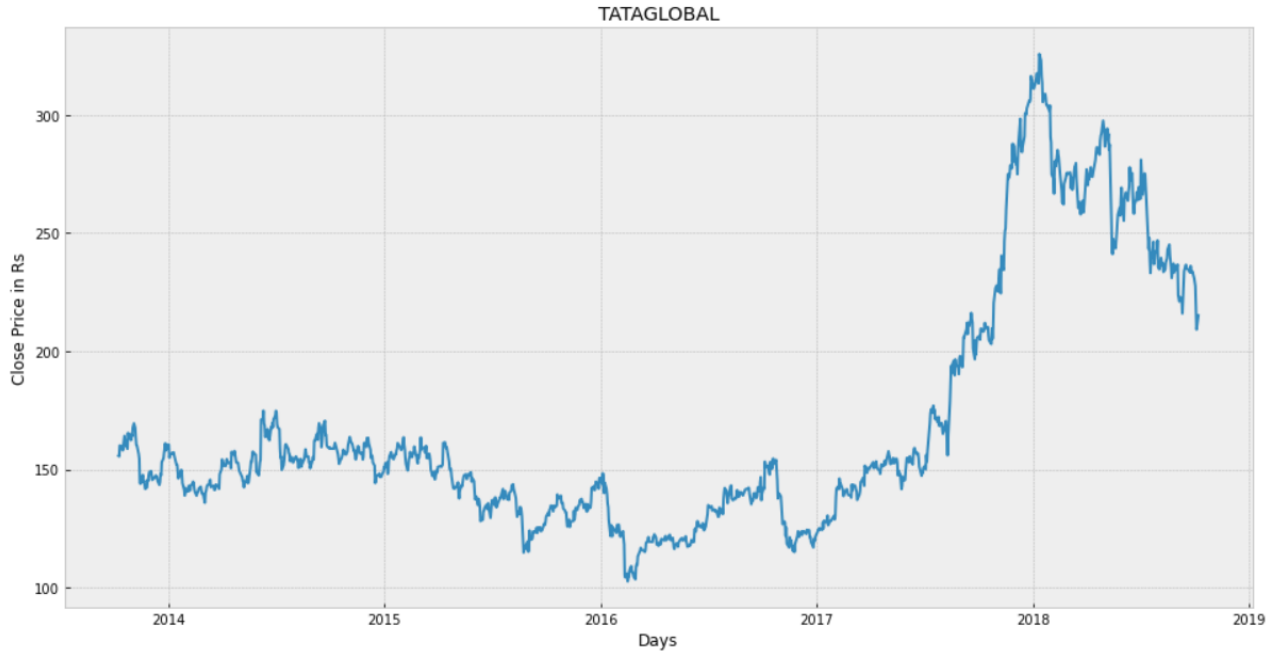


Fig 6: Graph of Close Price history

Step 4: Preprocessing

```
[[215.15]
 [209.2 ]
 [218.2 ]
 ...
 [147.7 ]
 [143.95]
 [144.3 ]]
```

Fig 7: Data Scaling

Step 5:Prediction Model:

```
[234.55 233.35 237.05 231.   238.65 266.3  237.55 236.9  239.35 237.55
 269.35 233.55 233.55 236.35 237.6  239.5  234.65 235.45 235.65 213.7
 242.2  242.25 239.1  236.9  246.3 ]

[210.16209705 204.67732716 212.97361775 221.63863237 224.68060559
 227.30776427 226.84685924 227.76866931 229.47401793 226.89294974
 228.09130283 228.36784585 228.50611736 229.93492296 227.49212629
 217.07567254 210.94563561 216.476496   217.35221556 215.60077644
 216.84522003 218.04357311 230.02710397 227.81475981 229.38183692]
```

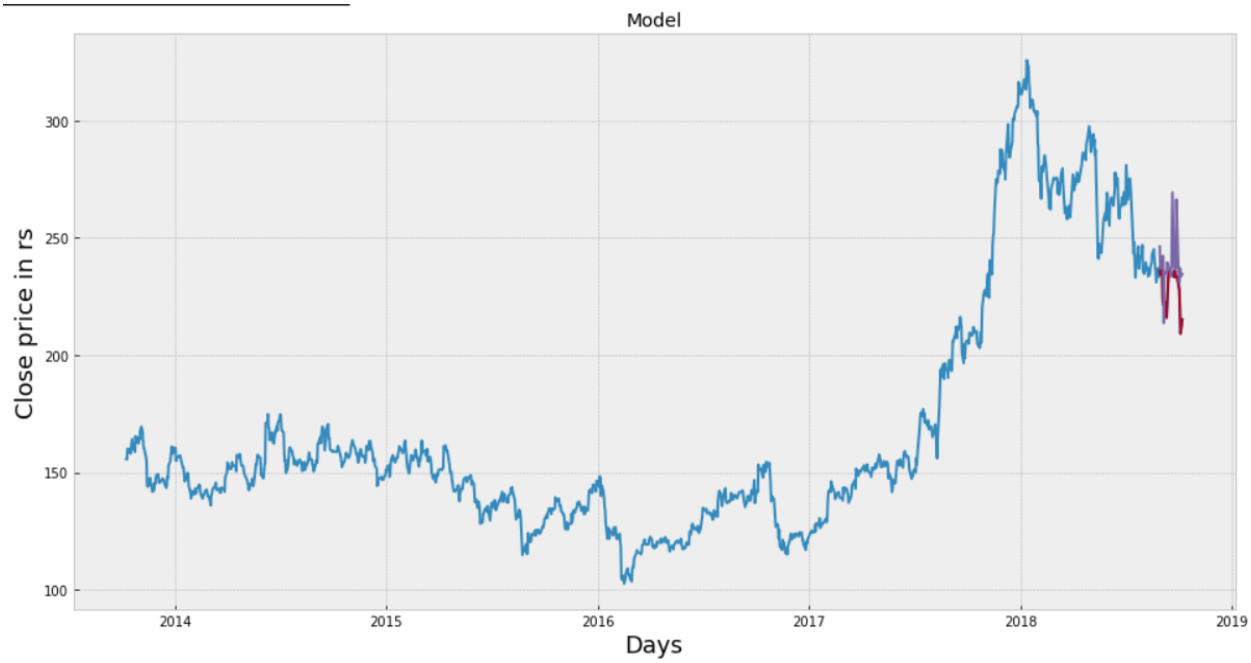
Fig 8 Dataset of Prediction model

Step 6: Predicted Result:

Date	Close	Prediction
2018-10-08	215.15	234.55
2018-10-05	209.20	233.35
2018-10-04	218.20	237.05
2018-10-03	227.60	231.00
2018-10-01	230.90	235.45
2018-09-28	233.75	240.55
2018-09-27	233.25	245.15
2018-09-26	234.25	243.00
2018-09-25	236.10	239.35
2018-09-24	233.30	237.55
2018-09-21	234.60	234.55
2018-09-19	234.90	233.55
2018-09-18	235.05	237.30
2018-09-17	236.60	236.35
2018-09-14	233.95	237.60
2018-09-12	222.65	239.50
2018-09-11	216.00	234.65
2018-09-10	222.00	235.45

Fig 9 *Predicted Close Price*

Step 7: Predicted Graph



Plot for Real vs Predicted value tree prediction

Plot for Real vs Predicted value linear regression

Conclusions

In the report, we will compare machine learning models like LSTM model, the CNN model, and also the hybrid approach of the LSTM + CNN model. We have a tendency to train the model using the data of NSE-listed companies to predict the stock's future value. This shows the proposed method is capable of distinctive interrelation with the data.

Based on the companies and sectors, the existence of the trends and the period of their existence will differ. The analysis of this type of cycle and trends can offer aore profit to the investors. In future work, we add more stock market data and compare more model to improve the accuracy of predicted stock prices. In the future, for better accuracy models can be trained with more varied and detailed data. Also, other algorithms along with the proposed can be used to create a new hybrid model.

References

- “Stock price prediction using LSTM, RNN and CNN-sliding window model - IEEE Conference Publication.” https://ieeexplore.ieee.org/document/8126078 (accessed Dec. 27, 2019).
- https://www.seldon.io/machine-learning-regression-explained#:~:text=Regression%20is%20a%20technique%20for,used%20to%20predict%20continuous%20outcomes.
- https://www.geeksforgeeks.org/videos/stock-price-prediction-in-machine-learning/
- https://www.kaggle.com/code/samyakjhaveri/stock-prediction-using-lstm-on-tata-global-stock/notebook
- https://randerson112358.medium.com/predict-stock-prices-using-machine-learning-python-f554b7167b36