Stock Market Prediction Using Different Machine Learning Techniques

VISHESH JAIN (2K18/CO/391), VISHU DHAMA (2K18/CO/394)

Abstract - **In this paper, we explored different ML models (Multilayer Linear Regression, CNN, LSTM, LSTM with GRU and dropout layers) to predict daily closing price of RIL stocks. Closing price is very important for financial markets, so we analysed the different ML models. The RIL dataset we used to conduct this study consists of daily prices from 2000 to 2020 and the final features used are open, high, low, close prices (OHLC) which are selected using data preprocessing techniques. The model which has LSTM with GRU and dropout layers performed the best among other models with better predictions over different time periods.**

Keywords - stock price prediction, Linear Regression, Long – short term memory (LSTM), Gated Recurrent Unit (GRU), Convolution Neural Network (CNN), Nifty -50 dataset , Sequential dataset , Dropout

I. INTRODUCTION

Stock market is considered to be one of the most unpredictable systems globally. But are the users perfectly rational or irrational entities ? No. There are numerous factors like some restrictions, limitations and psychological differences, etc. that may determine the particular behaviour of users for their respective factors. Although some sudden events might fluctuate the trend, but in general terms it is not completely unpredictable. Studying the factors and analysing them, we can deduce a pattern and predict the trends and future prices.

In this paper, we have created optimised models to study the patterns of the stock prices over different time periods to predict the closing price of RIL stocks. Reliance Industries Ltd., incorporated in the year 1973, is a Large Cap company (having a market cap of Rs. 1428852.36 Crore) operating in diversified sector. India's largest stock by market capitalisation — accounted for about 43 per cent of the benchmark Sensex's rally, hence an important company which has a major effect on the Indian Stock Market. For the study and prediction of RIL stocks, we have used four different models with unique architectures for recalling longer term information. This project shows different architectures of RNN, CNN and Linear Regression and how well these models perform while predicting longer term prices of the RIL stocks.

More specifically, we are using the dataset containing data of more than 5000 days between year 2000 and 2020. The dataset contains features - open, high, low, close, VWAP, volume, turnover, trades, deliverable volume but some of them are discarded as they had lots of null values and some of them are discarded as they were not relevant to our study of prediction of close prices, hence the final feature space of the dataset on which the predictions are made are - Open, High, Low, Close and using these features we have analysed our different ML models and compared their performances.

II. EXPERIMENTAL DESIGN

A. Dataset

The dataset consists of stock market data of 50 top companies whose stocks traded at high rates during the 1960s and 1970s in the Indian Stock Exchange (Total 5141 entries). Out of these 50 companies we selected Reliance in our study. This dataset contains various features related to the stock exchange market from 1st January 2000 to 31st July 2020. These features are Date, Symbol, Series, Prev Close, Open, High, Low, Last, Close, VWAP, Volume, Turnover, Trades, Deliverable Volume, %Deliverble.

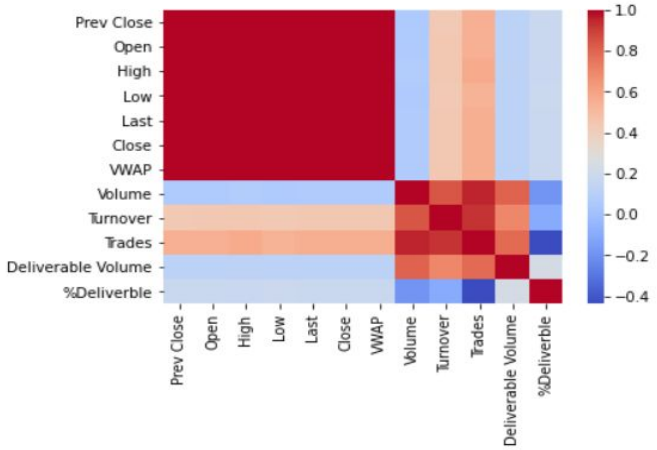
The symbol is the name of the company/ industry which is reliance in our case and the series is EQ (Equity: Delivery + Intraday Trading). Prev Close is the closing price of the previous day. Open is the starting trading

price of the stock and it's different from the closing price of the previous day.
Closing Price and Last Price are two different terminologies, Last is the last price at which a transaction happened and Closing Price is the weighted average of the stock price in the last 30 min before closing.
Volume is the number of stocks traded on a particular day.

B. Dataset Cleaning and Preprocessing

Trades, Deliverable Volume, and %Deliverble features are dropped because these features contain approx 10% null values in the dataset and Prev Close doesn't have any significance in our model as we will modify our dataset from time series to sequential dataset.

Other features like Volume, VWAP are not taken into account after analyzing the loss of the model. Finally, the features of the stock market that are considered are Open, High, Low, and Close.



C . Time Series to Sequential Dataset

The aim is to predict the closing price of the coming future day's but the limitation is that the stock market inherits inconsistency, which makes it difficult to make predictions. On the basis of only the current day price, the closing price of the next day can't be predicted accurately. So, the time-series dataset should be converted into a sequential dataset to do the prediction on the basis of the current trend. In this study, sequential length has been taken to be 10 meaning that the OHLC features of the past 10 days will be required for prediction of the closing price of the current day.

D. Assumptions

There are two types of terminology in Predictions: the closing price of one day and multiple days.
Prediction of the closing price of only one day, for example, let's say we have the input of OHLC features from 1st Aug to 10th Aug then we can easily do the prediction of the closing price of 11th Aug and this is what our models do.
But in case of multiple days closing price prediction, for example, now we want to do the prediction of closing price from 11th Aug to 30th Aug using values of OHLC features from 1st Aug to 10th Aug as input. For the prediction of each day, we need information of the past 10 days, so we need to make some assumptions to make multiple day predictions. After going through some research paper, we found that the values of all the features ie Open, High, and Low can be taken the same as the predicted Closing price. So now using our same model, we can predict the closing price of any number of days.

E. Models

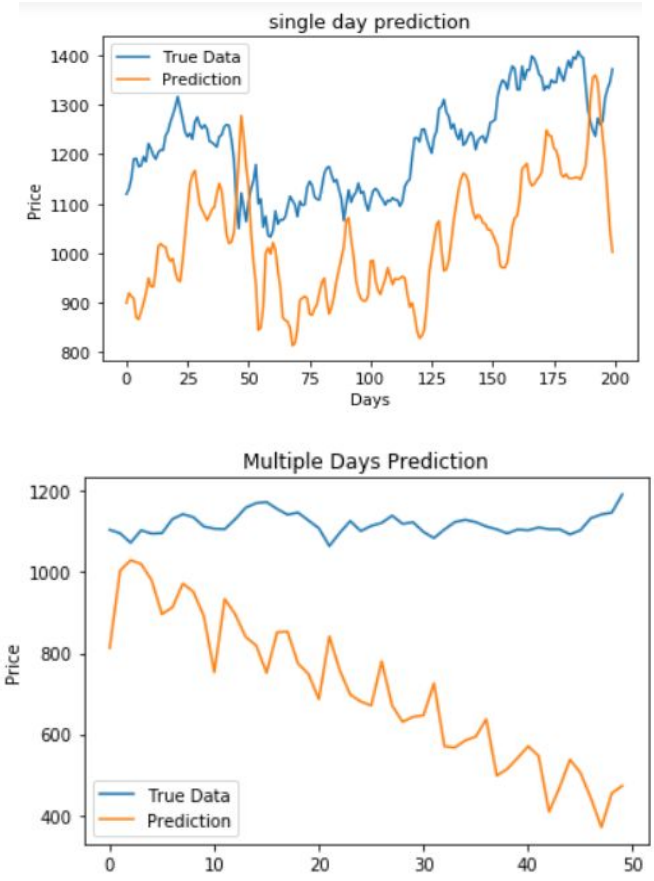
1. Linear Regression

Linear Regression is a technique of finding a linear relationship between one or more independent variables and a dependent variable.

The basic Linear Regression model cannot be trained on a sequential dataset. For each training example, we have a set of 10 days, and each day having 4 features ie OHLC. One way of using Linear Regression is to apply it on a set of 40 features (10 days x 4 features = 40). But it would not be the right way of transforming a sequential dataset into a normal dataset.

Now, MultiLayer Linear Regression can solve this problem, there would be 2 layers in Linear Regression

where the first layer will extract the crux of each day using the OHLC features of given days and then the second layer will take input of the crux of the past 10 days (from layer one) to predict the closing price.

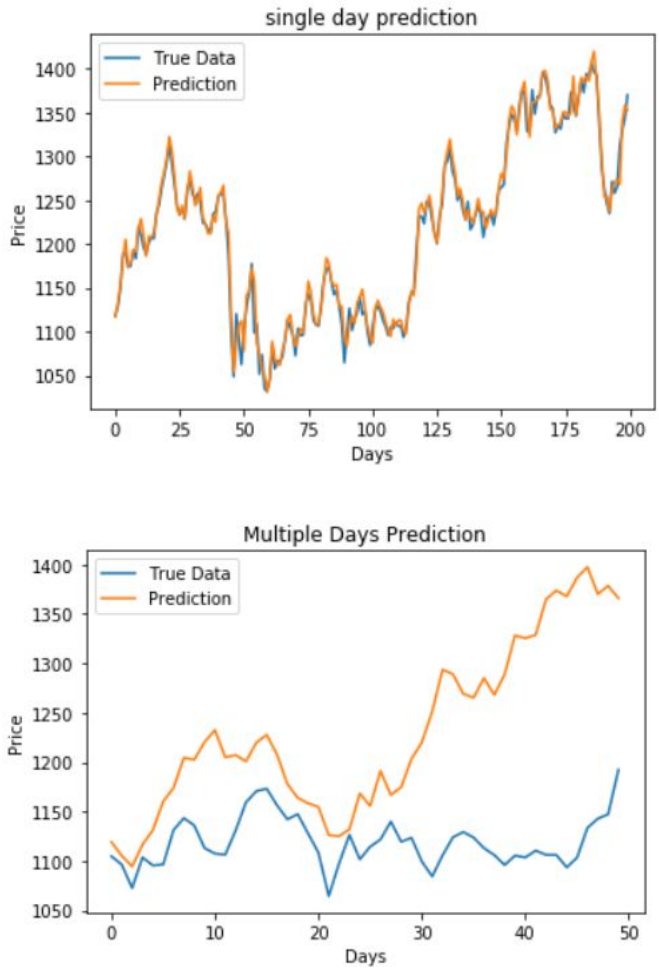


The problem with linear regression is that it can't use the past day's information appropriately for prediction and hence, have a very high testing loss.

2. CNN

Convolution Neural Networks (CNNs) are neural networks that use at least one convolutional layer in place of general matrix multiplication.
CNNs have an advantage over general feed forward networks that there is no need to flatten the input. We can give the input in its original shape which decreases CNN's complexity and CNNs are capable of considering locality of features, hence are more efficient while working on sequential time series data.

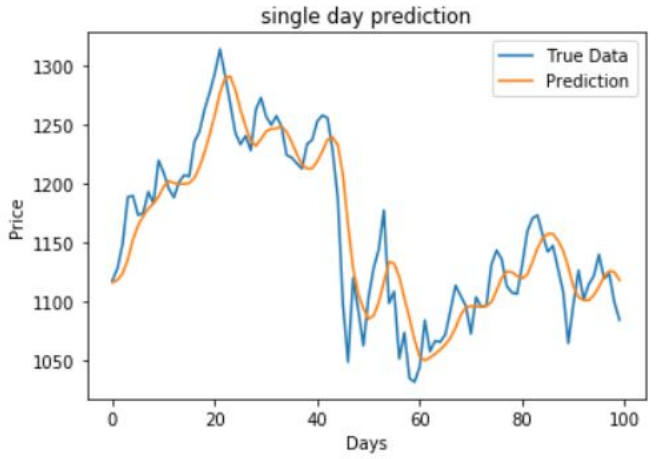
So in this model two convolutional layers are used which have 64 filters with kernel size 2x2 and input shape for the first layer is (10x4x1). After these layers a dropout layer is used and then a few dense layers to get our final output.

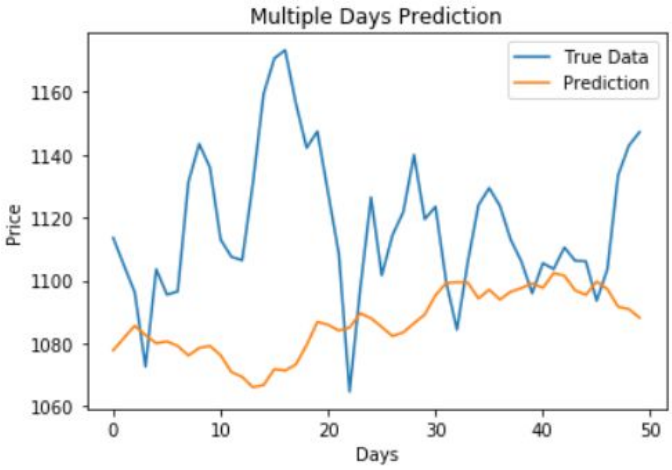


3. LSTM

Long Short-Term Memory (LSTM) is an advanced form of recurrent neural network. It is composed of a cell state, an input gate, an output gate and a forget gate. The cell state has the information from previous time instances while the gates help to regulate the information into and out of the cell state.
LSTM network can process entire sequences of data and it can handle the vanishing gradient problem due to which we used LSTM network in this model over traditional recurrent neural networks.

In this sequential model we used two LSTM network layers, the first layer has 64 LSTM units and the second layer has 32 LSTM units followed by a dense layer with one neuron i.e. our output layer.

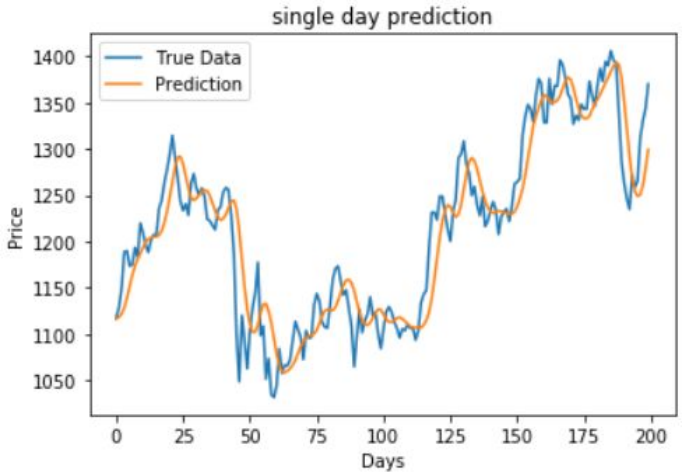


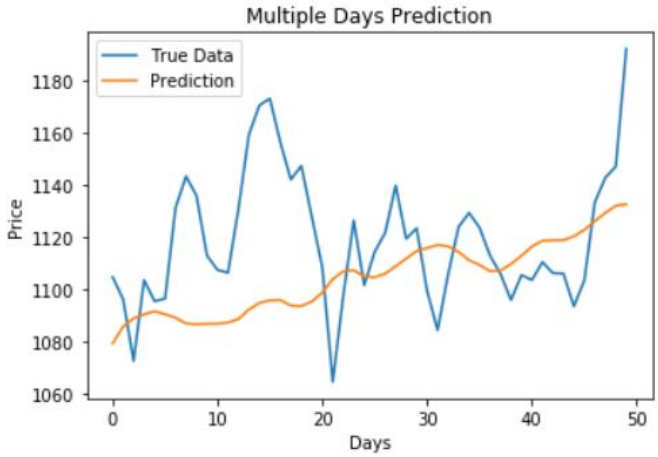


4. LSTM and GRU

This model is an enhancement of the basic lstm model along with GRU (Gated Recurrent Unit) and Dropout layers. Simple lstm model not performing efficiently in case of multiple days prediction , introducing a dropout layer and extending the depth of the model reduced the loss of long term predictions to a satisfactory level.

Dropout layer is used to prevent overfitting as our study is very prone to overfitting and can overfit for single day prediction. This model contains three lstm layers , one gru layer, two dropout layers and final output dense layer.





Impact of Sequence Length

Initially we assumed the sequence length of 10 for our study , but to find the optimal length of sequence we converted our dataset with different sequence lengths of length 5 , 10 , 15 , 20 and 25 . For evaluating the best sequence length , we used the model that comes out be the best i.e. LSTM and GRU .
Following are the Performance of different sequences :
M1 : Model with Sequence Length 5
M2 : Model with Sequence Length 10
M3 : Model with Sequence Length 15
M4 : Model with Sequence Length 20
M5 : Model with Sequence Length 25

Mean Squared Error :

Model	1	2	3	4	5	10	50
M1	0.01	0.01	0.01	0.01	0.02	0.04	0.32
M2	0.01	0.01	0.01	0.01	0.02	0.03	0.22
M3	0.01	0.01	0.01	0.01	0.02	0.03	0.22
M4	0.01	0.01	0.01	0.01	0.02	0.04	0.23
M5	0.0	0.01	0.01	0.01	0.02	0.04	0.24

Mean Absolute Error :

Model	1	2	3	4	5	10	50
M1	0.06	0.07	0.08	0.09	0.1	0.15	0.42
M2	0.06	0.06	0.07	0.08	0.09	0.13	0.33
M3	0.06	0.06	0.07	0.08	0.09	0.13	0.35
M4	0.05	0.06	0.07	0.09	0.1	0.14	0.36
M5	0.05	0.06	0.07	0.09	0.1	0.14	0.36

Root Mean Squared Error

Model	1	2	3	4	5	10	50
M1	0.09	0.09	0.1	0.12	0.13	0.2	0.56
M2	0.08	0.09	0.1	0.11	0.13	0.18	0.47
M3	0.08	0.09	0.1	0.11	0.12	0.18	0.47
M4	0.07	0.09	0.11	0.12	0.13	0.19	0.48
M5	0.06	0.08	0.1	0.12	0.13	0.19	0.49

R2 Score :

Model	1	2	3	4	5	10	50
M1	0.91	0.9	0.87	0.83	0.78	0.55	-0.48
M2	0.92	0.9	0.87	0.84	0.8	0.61	-0.03
M3	0.92	0.91	0.88	0.85	0.81	0.64	-0.03
M4	0.93	0.9	0.86	0.82	0.78	0.58	-0.09
M5	0.95	0.91	0.87	0.83	0.79	0.6	-0.12

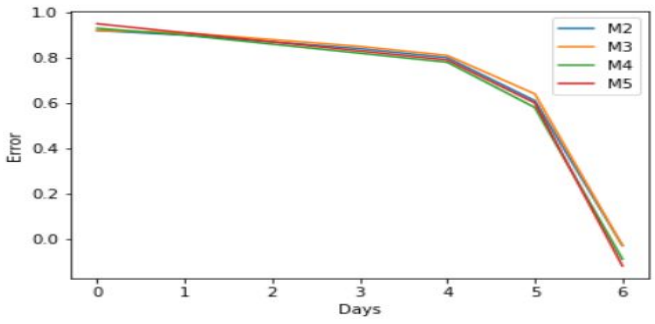


Figure : Mean Square Error Vs Day Predicted

After training the model for different length ,we collected the performance measures and found that model with sequence length of 10 and 15 are performed best with sequence length of 15 having an edge over sequence length of 10 .

III. RESULTS AND ANALYSIS

In our study, we have used four different performance measures to analyze the different models.
M1 : Linear Regression
M2 : LSTM
M3 : Convolutional Neural Network
M4 : LSTM and GRU
Following performance measures are used :

1. Mean Squared Error (MSE) is the mean of squared prediction errors (Actual Value - Predicted Value).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

In the above formula,
$\widehat{Y}$ = Predicted Values
Y = Observed Values
n = Total number of data points

	1	2	3	4	5	10	50
M1	0.195	0.027	0.029	0.05	0.067	0.199	1.929
M2	0.004	0.007	0.01	0.014	0.019	0.046	0.35
M3	0.001	0.004	0.007	0.01	0.014	0.031	0.326
M4	0.006	0.008	0.01	0.013	0.016	0.034	0.22

After comparing MSE from the above table for each model, we observed that the minimum value of MSE for predicting next day closing price was obtained by using M3 and by increasing the number of days M4 obtained the minimum MSE.

2. Mean Absolute Error (MAE) is the average of absolute values of prediction error (Actual Value - Predicted Value).

	1	2	3	4	5	10	50
M1	0.388	0.136	0.141	0.194	0.228	0.407	1.309
M2	0.048	0.062	0.077	0.091	0.105	0.166	0.433
M3	0.02	0.045	0.06	0.072	0.083	0.128	0.466
M4	0.057	0.062	0.071	0.082	0.092	0.135	0.333

After comparing MAE from the above table for each model, we observed that the minimum value of MAE for predicting next day closing price was obtained by using M3 and by increasing the number of days M4 obtained the minimum MAE.

3. Root Mean Squared Error (RMS) is calculated by taking the square root of mean squared error.

	1	2	3	4	5	10	50
M1	0.442	0.165	0.17	0.224	0.256	0.446	1.389
M2	0.065	0.082	0.1	0.119	0.137	0.214	0.591
M3	0.031	0.062	0.082	0.1	0.117	0.177	0.571
M4	0.08	0.088	0.1	0.114	0.127	0.184	0.469

After comparing RMSE from the above table for each model, we observed that the minimum value of RMSE for predicting next day closing price was obtained by using M3 and by increasing the number of days M4 obtained the minimum RMSE.

4. R-Squared (R2) is equal to 1 minus ratio of sum of residual squares to the total sum of squares.

	1	2	3	4	5	10	50
M1	-1.513	0.645	0.633	0.375	0.174	-1.288	-8.101
M2	0.938	0.898	0.84	0.761	0.664	-0.162	-60.89
M3	0.988	0.951	0.911	0.866	0.818	0.587	-1.693
M4	0.917	0.902	0.875	0.839	0.803	0.615	-0.025

After comparing the R2 score from the above table for each model, we observed that the maximum value of R2 score for predicting next day closing price was obtained by using M3 and by increasing the number of days M4 obtained the maximum R2 score.

Linear regression model has high Mean Squared Error and negative R2 score but it can be seen that this model performs better for long term prediction as compared to next day prediction .
Model with only Lstm and dense layers (M2) performed very well for next day prediction but when multiple days are predicted it performed even worse than Linear Regression Model .
M3 model that is model with Conv2d layer outperformed for within 5 days prediction but on predicting the days further ahead it's loss in prediction increased.

Our last model(M4) with additional dropout layers , and giving satisfactory performance for more than one days prediction and the reason may be that dropout layers can reduce the impact of recent days during prediction by dropping some percentage of neurons and that is why having a bit less accurate for first 5 days prediction .
By comparing the performance of all the ML models using different performance measures, we analyzed that for just next day predictions M3 is performing better but for longer predictions M4 seems to perform better among all the models used in this study.

CONCLUSION

This study attempts to perform Stock Price Prediction on Reliance Stock Market Data from Nifty 50 dataset , stock market predictions are considered to be very important and complex things for investors. In this paper, we have covered all the steps that could be taken to preprocess the stock market dataset like feature selection , standardization and finally converting the time series dataset to Sequential dataset .

Four models, namely Linear Regression ,LSTM with dense layers , Convolutional Neural Network and lstm with gru and dropout layers , have been trained on the specified dataset. After determining the best parameters and validating their results using hold out validation, their performances have been compared using mean squared error , mean absolute error , root mean squared error and finally R2 score . From the results obtained, it

can be concluded that the LSTM with gru and dropout layers outperforms the others in terms of performance measures .

REFERENCES :

[1] “Nifty -50 Stock Market Dataset” https://www.kaggle.com/rohanrao/nifty50-stock-market-data accessed:2020-09-20.
2020-09-10.

[2] URL :
“https://www.investopedia.com/terms/o/ohlcchart.asp”
accessed: 2020-09-25

[3] Akshay Sachdeva , Geet Jethwani , Chinthakunta Manjunath , Balamurugan M , Adapalli V N Krishna ; “An Effective Time Series Analysis for Equity Market Prediction Using Deep Learning Model ” , IEEE

[4] Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P ; “STOCK PRICE PREDICTION USING LSTM,RNN AND CNN-SLIDING WINDOW MODEL” , IEEE

[5] Raul Girbal , Glib Stronov ; “Predicting Stock Price Dynamics using stacked GRU’s and LSTM’s”

[6] Jerome T. Connor, R. Douglas Martin ; ”Recurrent Neural Networks and Robust Time Series Prediction” , IEEE

[7]Chen, J.-F., Chen, W.-L., Huang, C.-P., Huang, S.-H., & Chen, “*Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks”. 2016 7th International Conference on Cloud Computing and Big Data*