Twitter Sentiment Analysis to improve Stock Price Prediction using LSTM

Supervisor : Dr. Suman Kundu

Anmol Gupta (B18EE060), Yash Chaubey (B18EE013)

ABSTRACT

This project aims at prediction of stock market closing prices of a particular firm by analyzing the various underlying factors, uncovering various patterns affecting the volatility, predictability and the trends of the same. We also focus on the external factors that might influence the future value of stocks through scrapped news regarding the organization. A model has been developed to predict the stock prices of a company and has been trained, validated and tested on the stock data of Microsoft Corporation (MSFT). Furthermore, the accuracy of the prediction with and without incorporation of external factors is compared in terms of certain pre-defined metrics and plots.

1 INTRODUCTION

Any financial success lies in preparing in the present by predicting the future. Stock market prediction[1],[2], [3] has been a major attraction within the data scientist and researcher community for decades, with only aim to make some sense out of millions of unrelated raw data thereby making informed and rational financial decisions.
This problem falls under the category of time-series analysis[4],[5] and the most trivial procedure would be to linearly fit the stock prices[6]. The stock prices being non-linear and extremely volatile[7], the researchers moved on to regression models [8], moving average based models and its variants[9]. Soon the research community was attracted by different Machine Learning models[10],[11]. However such models would fail to identify long term

trends. In the past few years, with increasing popularity and effectiveness of Deep Learning Techniques, the researchers have been attracted to Deep Learning based methods for addressing this regression problem[12],[13] The neural networks outperform the previous methods, as they can deal with non-linear and non-stationary data[14]. The Recurrent Neural Networks[15], particularly Long Short Term Memory(LSTMs)[16] networks, which excel in identifying long term patterns and are capable of predicting with more accuracy than the above mentioned techniques. Hence we have used LSTM networks for the purpose of our analysis.
As evident, the public sentiment about the organization has always been an important external factor which causes sudden change in stock prices[17], hence we tried to incorporate these factors into our model by scraping twitter news[18] and including sentiment scores of that news into our model. Microsoft being a popular firm and having a great community on twitter, we decided to do our analysis on their stock market data.

2 METHODOLOGY

We collected data regarding stock history and scrapped tweets related to the organization. Then we used a pre-trained sentiment analyzer "VADER"[19] to evaluate sentiment of those tweets after appropriate pre-processing of scrapped tweets. There was a desired correlation between the tweet's sentiment and financial data. Finally we incorporated those sentiments in our model and trained and tested it which gave a significant increase in the R2 Score[20] of the predicted values. The block diagram in figure 2.1 summarizes the complete methodology.

2.1 DATA COLLECTION

2.1.1 FINANCIAL DATA

We collected the Microsoft Corporation's financial history from Yahoo Finance[21] with the following parameters - Opening Price , Closing Price, Highest , Lowest trading price of the day, Volume of stocks traded and Adjusted Closing Price. The data collected ranged from 21st September, 2009 to 28th May, 2020. These were trusted and reliable data excluding the days when markets were closed (weekends).

2.1.2 SENTIMENT DATA FROM TWITTER

We used twitter APIs to collect bulk tweets and managed the scrapped tweets using different scraping techniques. A third party open source library scrappy[22] is used. We started by scraping all tweets from the organization's timeline (Microsoft's twitter), but there was almost no correlation (correlation : -0.4196%)(figure 2.2) of sentiment score of these tweets as calculated in section 2.2.2, with closing price of financial data.
This happened due to many irrelevant tweets not related to stock and also the company never posting negative tweets. Even if a setback was posted it was portrayed as such that overall sentiment of the tweet turned out to be neutral.
So we scraped all the tweets with keyword "#microsoft" and "#MSFT"(Registered stock listed name of microsoft) but there were too many irrelevant tweets not related to stock and daily scraped tweets count crossed about 30,000. Hence, we filtered the scraped tweets with the

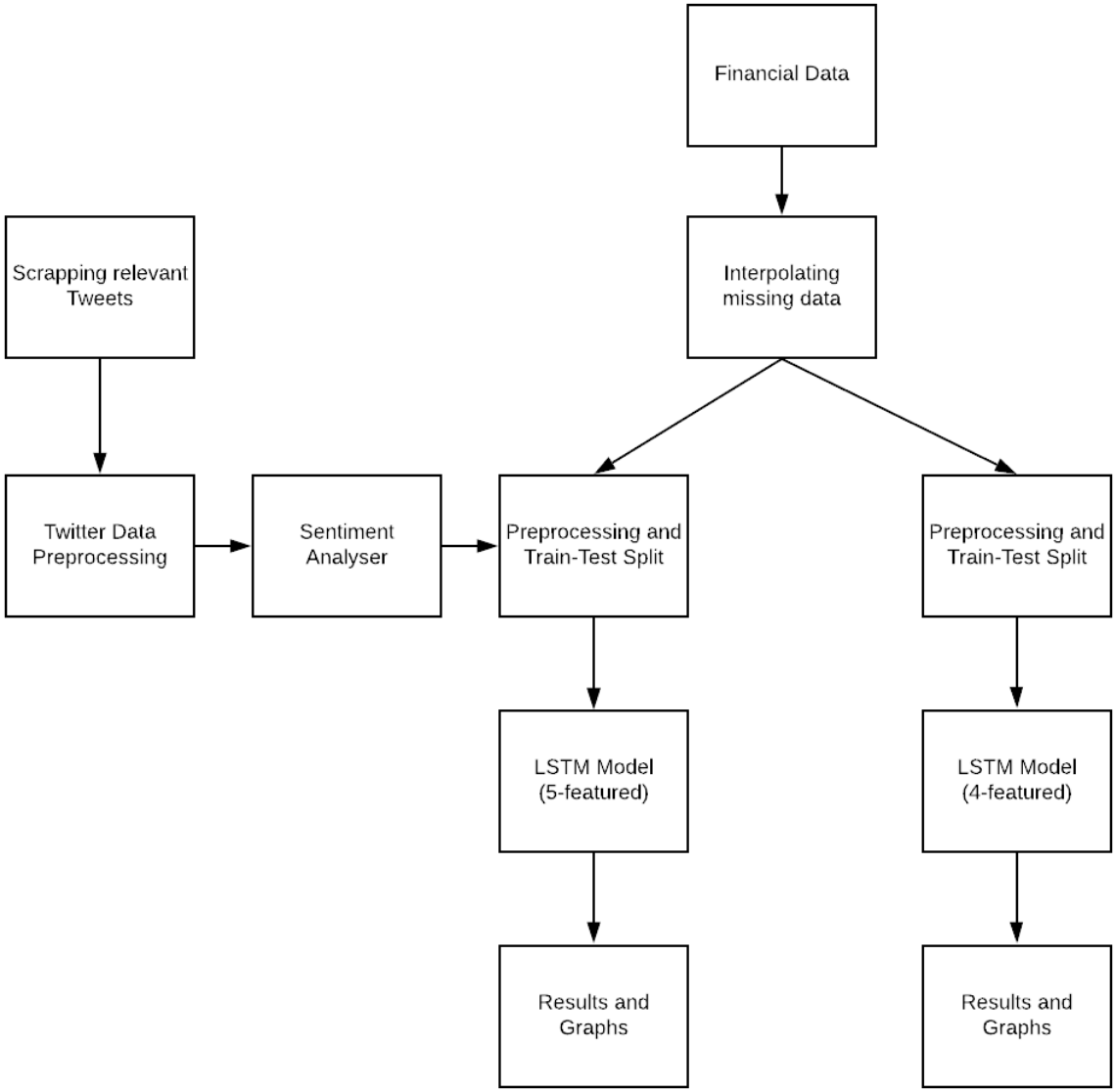


Figure 2.1. Block Diagram of Methodology

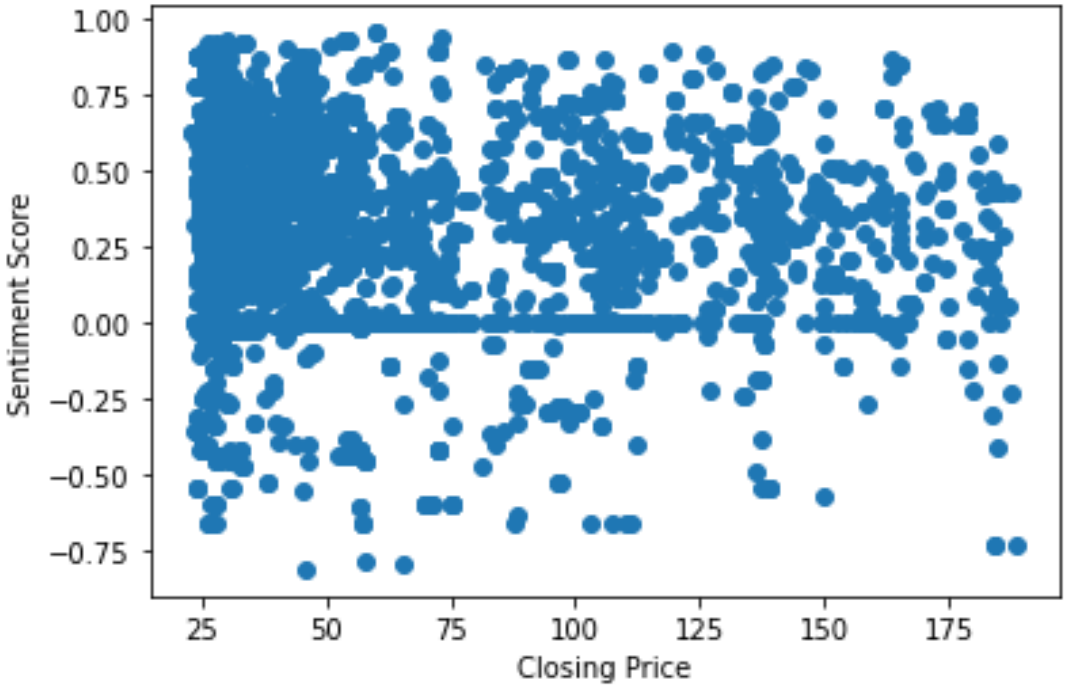


Figure 2.2. Scatter plot of sentiment score of tweets collected from Microsoft's twitter timeline v/s Closing Price of stocks

above mentioned keywords by only connsidering the tweets collected from verified twitter accounts by including a checker in our tweet-scraping algorithm. Unnecessary details were then removed by pre-processing described in section 2.2.

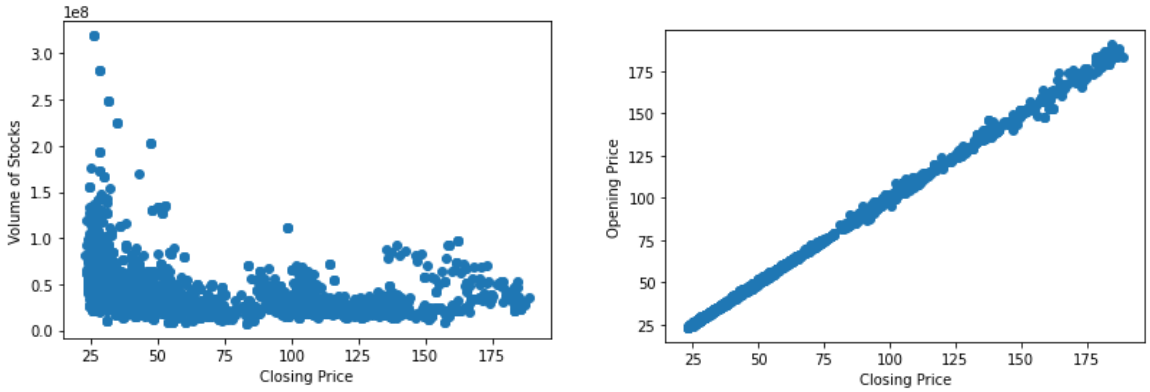
Table 2.1: Summary Of Data

Data	Features	Source	No. of Data
Microsoft Financial Data	Dail y Open, Close, High, Low Stock Price Volume and Adjusted Closing Price	Yahoo Finance	3903
Sentiment Data	Daily Tweets	Twitter	163,732

2.2 Data Pre-processing

2.2.1 Financial Data

Since, the originally collected data did not contain data for weekends, such gaps were populated using previous data(assuming the market was stable). As we want to predict only the closing prices, the volume of stocks and the adjusted closing prices were removed, since they were negatively correlated with the closing prices.



(a) Volume v/s Closing price (high negative correlation)

(b) Opening price v/s Closing price (high positive correlation)

Figure 2.3. Correlation of Volume & Closing Price with Closing Price

To avoid different features getting irregular weight assignment and pseudo-prioritization during training, we normalized all the data in the inclusive range [0,1], using the equation 2.1. The plot of normalized daily closing price of stocks against number of days can be seen in figure 2.4.

$$x_{\text{normalized}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{2.1}$$

For the purpose of training and testing, we had subdivided the data into the ratio of 60:20:20, i.e. first 60% of the data for training, next 20% for validation and the remaining 20% for testing our model.

2.2.2 Twitter Data

The tweets scraped from from twitter had lots of irregular expressions in them, so for the sentiment analyzer to work properly, we processed the tweets in the following ways:

- Removed links and URLs using regular expressions by removing matching sequences that RegEx detected.
- Tweets contained lot of hashtags(#) which were explicitly removed from the beginning of the words, using RegEx.
- Usernames(starting with '@') were removed from the tweets.
- Next using google translate APIs we identified and converted non-english native tweets into english.
- Irregular symbols, emoticons etc was removed leaving sentences which gave precise sentiment score.
- Further the same date tweets were clubbed together giving us day-wise time-series of relevant tweets.

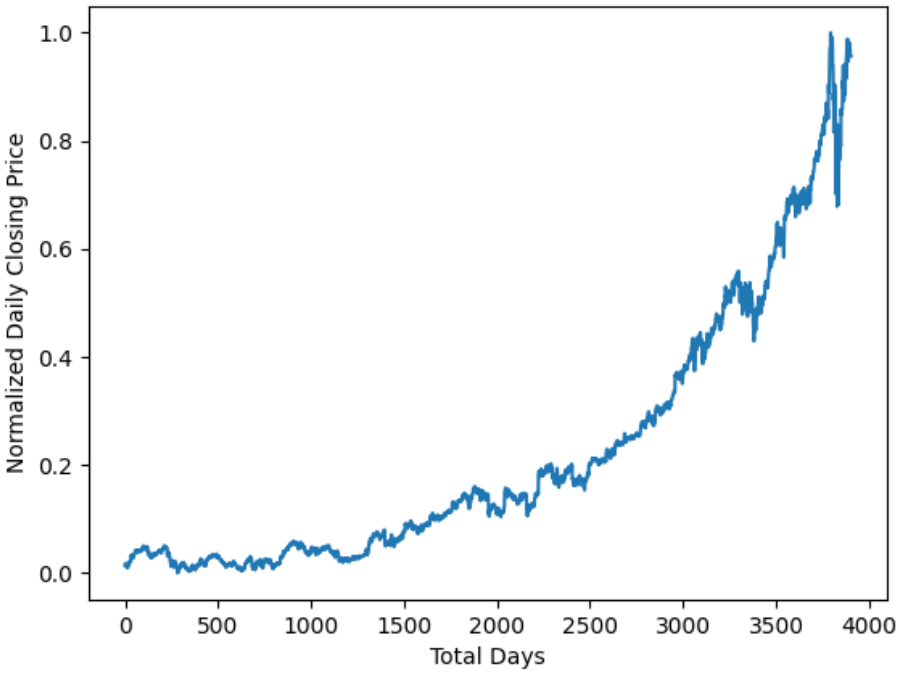


Figure 2.4. Normalized Daily Closing Price v/s Total Days

For the purpose of obtaining a sentiment score of each day, we used the pre-trained sentiment analysis library, VADER (Valence Aware Dictionary and Sentiment Reasoner). It assigned a compound score to tweets of each day by summing up the valence score of word/lexicons and normalizing it between -1(most negative) and 1(most positive). This daily compound score effectively quantified the general emotion of the tweet and showed a correlation of 14.9% with the daily stock closing prices.

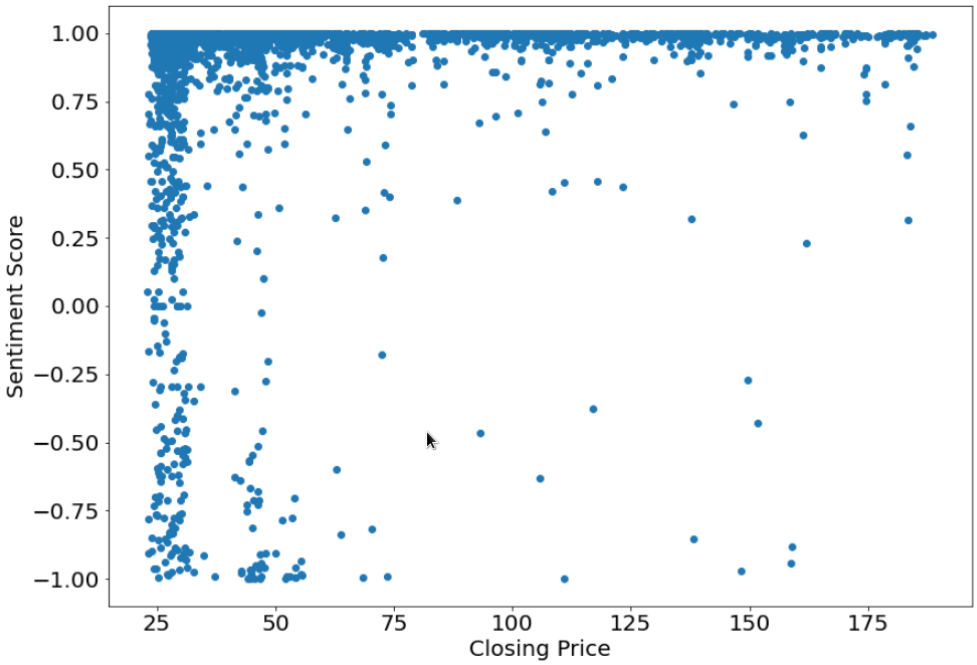


Figure 2.5. Scatter plot between Sentiment Score and Closing Price

These obtained values are further normalized in the range [0,1], using the equation 2.1

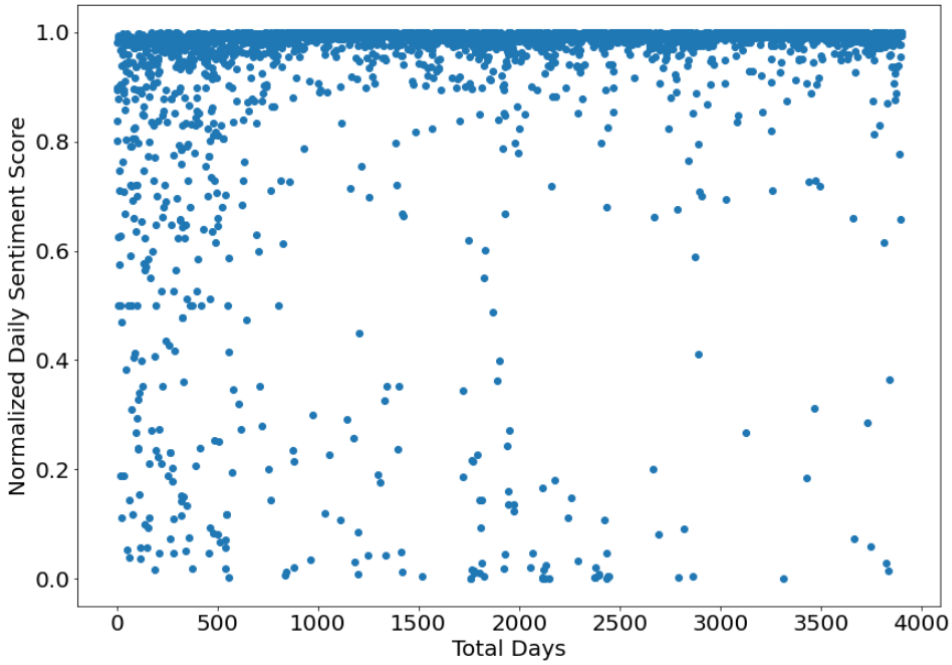


Figure 2.6. Time series scatter of Daily Sentiment of tweets

2.3 MODELS

2.3.1 LONG SHORT TERM MEMORY(LSTM) NETWORKS

Recurrent Neural Networks are a class of Neural Networks arranged in a multi-perceptron, layered structure. The connections between nodes in RNNs form a directed graph which allows it to exhibit dynamic behaviour. A finite recurrent neural network forms a directed acyclic graph internally and can be represented into its unrolled feed forward network form i.e. connected to its different versions in time as follows:

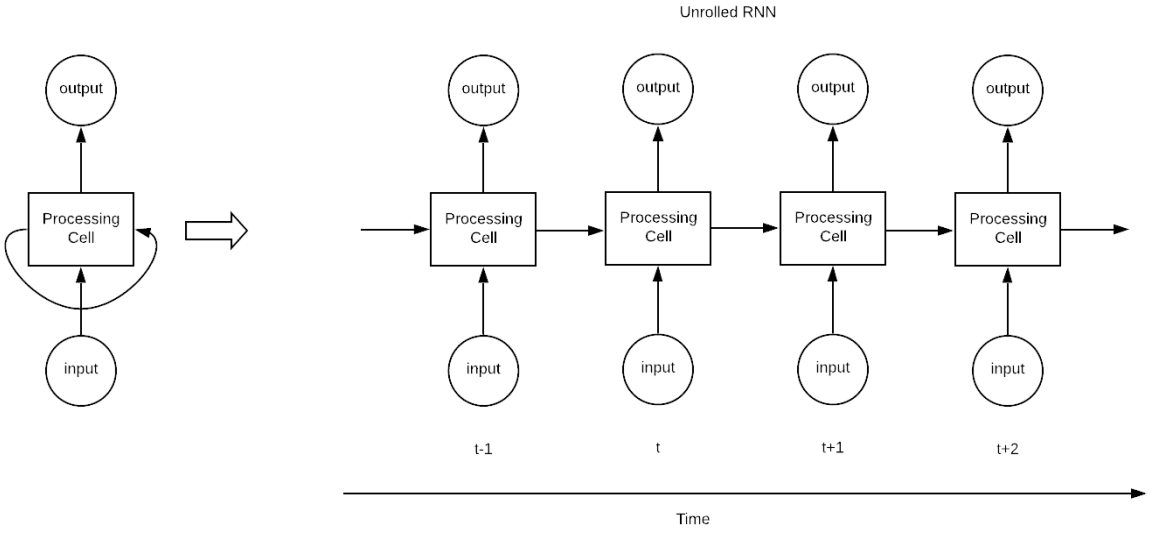


Figure 2.7. Recurrent Neural Network (Unrolled form)

Apart from the input and output cell, the most important and essential component of an RNN is the processing unit shown in figure 2.7. The processing unit in any generic RNN is usually a sigmoid or tanh layer. This restricts its prediction powers for long term trends and introduces the vanishing and exploding gradient problems. The major change in LSTMs is the internal processing unit, which is multilayered in contrast to RNN's processing unit.

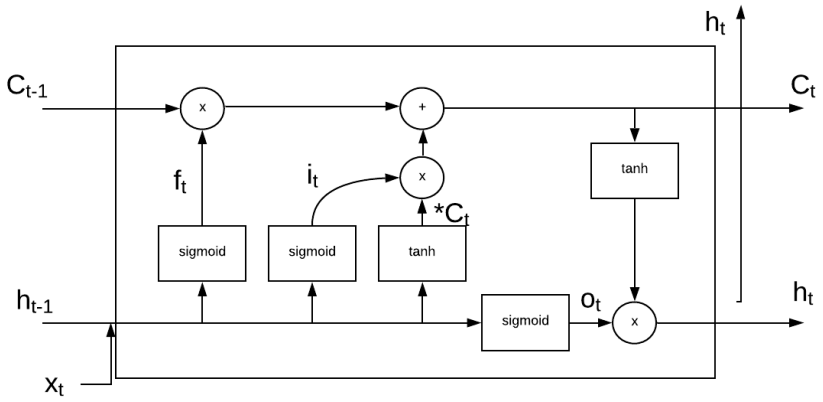


Figure 2.8. LSTM Cell

In figure 2.8, h_t : Output of the LSTM Cell, x_t : Input to the LSTM cell, o_t : Apparent Output of the LSTM cell, $*c_t$: Apparent Cell State value of the LSTM cell, c_t : Cell State value of the LSTM cell, f_t : Forget gate output of the LSTM cell

At the start of time 't', the input from x_t is concatenated to the output of previous output h_{t-1} which is passed through a sigmoid layer (forget gate), that decides information is to be forgotten and what is to be preserved. The output from forget gate (f_t) is squashed with the previous cell state value (c_{t-1}).i_t, the information to be updated along with $_t$ are multiplied and added to get the cell state value (c_t). But c_t is not the actual output of the cell. Rather the output is a filtered version of c_t, means not all the information from the current state is outputted but it is passed to the next cell.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{2.2}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2.3}$$

$$\tilde{c}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{2.4}$$

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{2.5}$$

$$c_t = \tilde{c}_t.i_t + f_t.c_{t-1} \tag{2.6}$$

$$h_t = o_t.\tanh(c_t) \tag{2.7}$$

W_f, W_i, W_o : Weight Matrices
b_f, b_i, b_o : Bias Matrices

We have implemented our model with an input layer consisting of 600 LSTM cells, One hidden layer consisting of 700 LSTM cells, a dropout layer of 15% to avoid overfitting, and finally an output Dense cell. The data (open,close,high,low) was divided into windows of first 60 for input and the closing price of 61st data as the output, for the purpose of training and testing. The model was trained for 100 epochs with early stopping.

2.3.2 Long Short Term Memory(LSTM) Networks with Sentiment

For incorporating sentiment into our model, keeping the hyperparameters same as in section 2.3.1, only the input was concatenated with a column of sentiment score hence consisted of 5 features instead of 4.

3 Experimentation & Results

3.1 Comparison Parameters

For comparing the accuracy of the prediction, we use the R-squared score[20] also known as the coefficient of determination(equation 3.1). It accounts for the scatter between the predicted value of the LSTM based regressor and the actual closing prices.

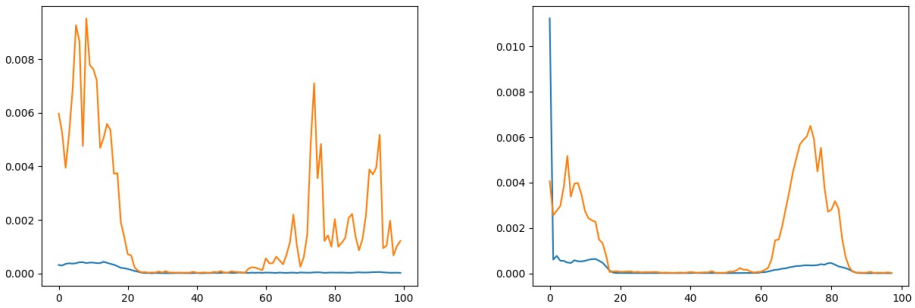
$$R^2 = 1 - \frac{\sum(y_{\text{actual}} - y_{\text{predicted}})^2}{\sum(y_{\text{actual}} - y_{\text{mean}})^2} \tag{3.1}$$

$$MSE = \frac{1}{n}\sum(y_{\text{actual}} - y_{\text{predicted}})^2 \tag{3.2}$$

The values of R2 score are in the range [0,1] , 0 being the worst fit and 1 being the best fit. Apart from that we have calculated the Mean Squared Error equation 3.2 for both the cases. The R2 Score and Mean Squared Error (MSE) for the predictions of both the models are compared in 3.1.

3.2 Discussion

The sentiment score obtained in section 2.2.2 showed a 14.9% positive correlation with the closing price and hence is a considerable measure for predicting purpose. A total of 720 days (almost two years) of closing price was predicted with the help of 2281 days of 4-featured train set (in case of LSTM) and 5-featured train set (in case of LSTM + Sentiment), with 722 days' data for validation purposes of the model.



(a) Without Sentiment **(b)** With Sentiment

Figure 3.1. Validation Loss(Blue), Training Loss(Orange) v/s Epochs

Figure 3.1a shows the training and validation loss of the LSTM model after a total training time of 1521.44 seconds and figure 3.1b depicts the same with sentiment incorporated and a total training time of 1464.33 seconds. In the training loss of both the cases, similar trends can be observed, in form of slight spikes towards the 60 epochs. The spike has more magnitude in case of only LSTM case as compared to the LSTM + Sentiment case denoting that the training loss is comparatively less in the latter case.

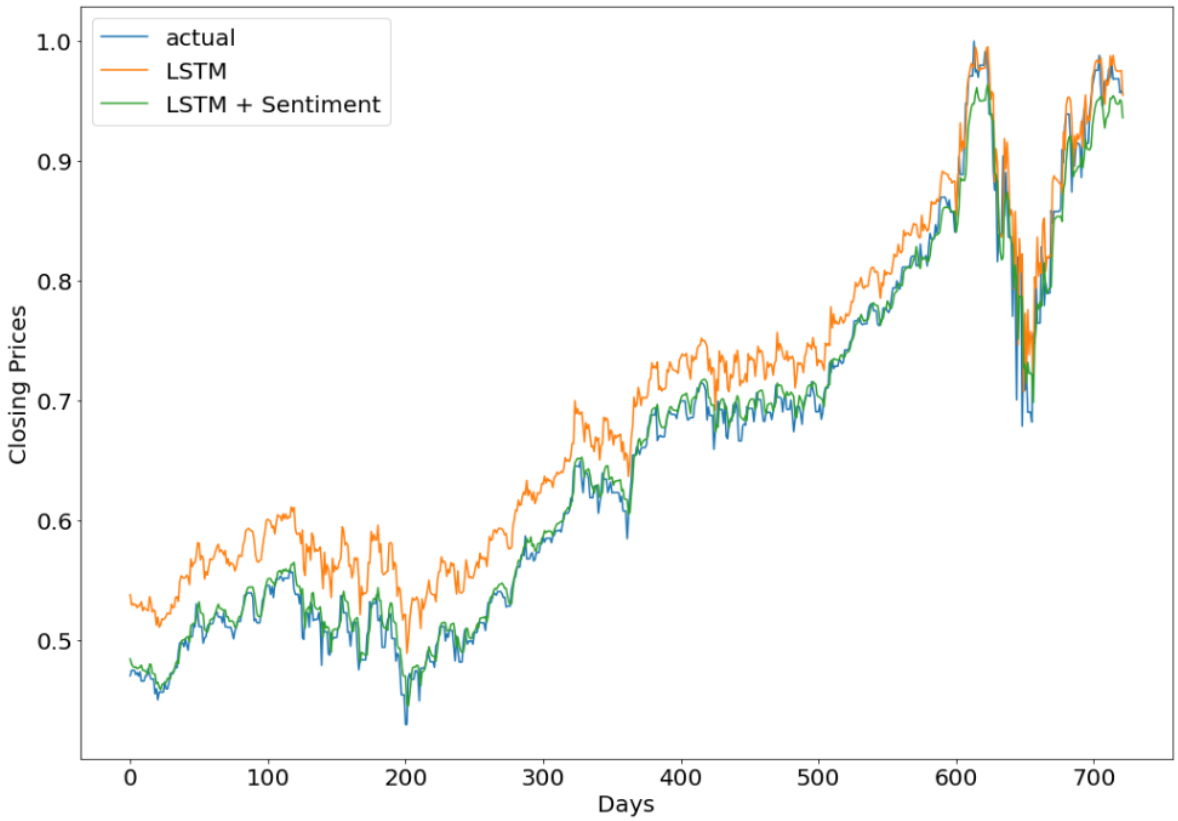


Figure 3.2. Analytical Comparison of Actual Price and Predicted Prices both the cases

Figure 3.2 shows the actual data from the test set and the data predicted using only the LSTM model and using the LSTM model with sentiment analysis. Blue colored curve represents the normalized test data, while the orange and the green represent the predicted data in cases 2.3.1 and 2.3.2 respectively. It is evident from the plots that the model trained with the additional feature of external sentiment fitted better.

Table 3.1: Metric Measure

Metric	**LSTM**	**LSTM + Sentiment**
R2 Score	0.883	0.987
MSE	0.0021304	0.0002669

4 Conclusion

In summary, the report establishes that adding an extra feature that accounts for the external sentiment increases the prediction accuracy by 10.4%. It is quite convincing to see how including relevant news sentiment as a parameter predicted even the uncertain results, which could not be solely predicted using LSTMs. Also, the data collected from twitter showed a positive correlation, thereby proving to be a good external factor for stock market analysis.

5 Future Prospects

It would be a great point of interest to see how the above results may improve by using more advanced sentiment analysis tools with refined news from broker firms or tuning different kinds of layers within the neural network architecture. Also, predicting stock indices (SP500, Nifty, etc.) would be beneficial to enhance some new findings as a wider range of external news is available for such indices (New York Times, Economic Time, etc.) instead of mere Twitter Data.

References

[1] D.C. Wunsch E.W. Saad D.V. Prokhorov. *Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks*. 1998.

[2] Dr. A. K. Mittra J. G. Agrawal Dr. V. S. Chourasia. *Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks*. 2013.

[3] Sarika Bobde Vivek Rajput. *STOCK MARKET FORECASTING TECHNIQUES: LITERATURE SURVEY*. 2016.

[4] Paul Whiteley. *Time series analysis*. 1980.

[5] Richard Palmer BlakeLeBaron W.Brian Arthur. *Time series properties of an artificial stock market*. 1999.

[6] Bayu Distiawan Trisedya Yahya Eru Cakra. *Stock price prediction using linear regression based on sentiment analysis*. 2015.

[7] Tim Bollerslev; Hans Ole Mikkelsen. *Modeling and pricing long memory in stock market volatility*. 2017.

[8] M Hakan Erdinç Altay Satman. *Regression techniques for the prediction of stock price trend*. 2005.

[9] Gurvinder Singh Manik Sharma Samriti Sharma. *Performance Analysis of Statistical and Supervised Learning Techniques in Stock Data Mining*. 2018.

[10] Tongda Zhang Shunrong Shen Haomiao Jiang. *Stock Market Forecasting Using Machine Learning Algorithms*. 2012.

[11] Mehak Usmani ; Syed Hasan Adil ; Kamran Raza ; Syed Saad Azhar Ali. *Stock market prediction using machine learning techniques*. 2016.

[12] Hiroshi Yajima Hirotaka Mizuno Michitaka Kosaka. *Application Of Neural Network To Technical Analysis Of Stock Market Prediction*. 1998.

[13] Jialin Liu; Fei Chao; Yu-Chen Lin; Chih-Min Lin. *Stock Prices Prediction using Deep Learning Models*. 2015.

[14] Eunsuk Chonga ;Chulwoo Han; Frank C.Park. *Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies*. 2017.

[15] Charles Elkan Zachary C. Lipton John Berkowitz. *A Critical Review of Recurrent Neural Networks for Sequence Learning*. 2015.

[16] Jurgen Schmidhuber Sepp Hochreiter. *LONG SHORT-TERM MEMORY*. 1997.

[17] Jeffrey Wurgler Baker Malcolm. *Investor Sentiment in the Stock Market*. 2007.

[18] Xiaojun Zengb Johan Bollena Huina Maoa. *Twitter mood predicts the stock market*. 2011.

[19] Eric Gilbert C.J. Hutto. *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. 2015.

[20] Maria-Pia Victoria-Feser Olivier Renaud. *A Robust Coefficient of Determination for Regression*. 2010.

[21] *Yahoo Finance:* `https://in.finance.yahoo.com/`.

[22] *Scrappy:* `https://github.com/scrapy/scrapy`.

SIGNATURES

Yash Chaubey

Anmol Gupta

Dr. Suman Kundu (Supervisor)