JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

STOCK PRICES PREDICTION USING SENTIMENT ANALYSIS
A Comparative study Analysing the influence of market sentiments on prediction accuracy

Arjun Bakshi **Mohammed Faisal**
Thadishetty Vamshi

NIIT UNIVERSITY, INDIA

Abstract : *Through this paper we aim to analyse the influence of market news sentiments on stock market prices prediction. We propose a hybrid model encompassing Deep Learning and Natural Language Processing.* ***Further, we go on to analyse a mechanism of feature selection and compare the effect of varying the time intervals of parameters.*** *Lastly, we analyse and compare pre-existing models and the model we have created after including different types of sentiment scores using LSTMs(Long Short-term Memory) and GRUs(Gated Recurrent Units).*

I. INTRODUCTION

There has been extensive research in the field of prediction of stock prices in the past using carefully structured and appropriately tweaked models. If it is correctly modeled, stock prices can be predicted and can give better results compared to human predictions. An approximate prediction of financial data using one or a series of models can help understand the movement of stock market and changes in the financial market at the macroscopic level and provide a good return to the investors, thereby allowing them to maximize profits. Since financial data have complex and fuzzy information it is extremely difficult to predict the movement. Stock prices can be predicted with two methods which are indirectly linked, one is Technical analysis where we use the historical data and some variables and attributes to predict the closing price of the stock using various Machine Learning and Deep Learning techniques. The other is Sentiment analysis which deals with the backend of the stock movement. Basically, the news which affects the particular stock or the whole bunch of stocks such as Nifty50, Bank Nifty, Sensex etc. Sentiment analysis is very important since the stock price of a company depends on intrinsic as well as extrinsic attributes. Some bad news can trigger the stock, it may affect a company value by decreasing its stock value. Some of the intrinsic factors could be company's net profit, liabilities, demand stability, competition in market, stakes in raw material supplier and finished product distributors etc. and some of the extrinsic attributes are crude oil price, dollar exchange rate, political stability, government policy decision etc. Note that extrinsic attributes are not in control of the company. Many researchers have tried technical analysis and neglected sentiment analysis which we think is the very essential thing to consider in the prediction. Many different statistical models were applied such as moving average (MA), Weighted moving average, ARIMA, SARIMA etc. Some used deep neural networks like CNN, RNN etc. In this paper we have proposed a hybrid model using Machine Learning, Deep Learning, and Natural Language Processing which will be predicting the closing price of the stock based on the past stock prices and the stock news affecting that company. Sentiment scores for each day will be added as a separate parameter to the dataset.Some researchers included market sentiments but fail to experiment with the lookback periods(Window Size). For example, using 2 previous day's sentiment instead of only one day.
Also, a combination of stock news and multivariate time series is taken into consideration in our paper. For example, high, low, open, close prices can all be used instead of only one in combination with sentiment analysis. Finally using a validation set during training and testing data.

II. PROBLEM STATEMENT AND OBJECTIVE

The target of our work is to collect stock prices of companies listed in NSE of India and scrap the news data related to that stock and predict the closing price of the stock. Prediction accuracy will be expected to be better than the previous research works in the field of stock market prediction by using LSTM and GRU as a robust Deep learning architecture in technical analysis. This will include web scraping and evaluating the keywords to understand the sentiments of the market. More financial data like open value, close value, high, low, volume traded, turnover is collected so that the accuracy will be increased. News will be extracted from google news or money control or screener and then scraped and downloaded. It will be done through python libraries (Beautiful Soup, Selenium and Scrapy), then we can get the sentiment scores by comparing the news with lexicons and then repeat the process for every keyword. These scores will then be added into our dataset.

III. LITERATURE REVIEW

Arjun and Shakya in [1] propose a unique machine learning model combining particle swarm optimization and Least-Square SVMs. Displays results on 13 company datasets. It shows the advantages of incorporating optimization techniques in addition to models, here free parameter combination selections.
The study showed that this combination outperformed even ANNs which are prone to overfitting and local minima problems; these problems are effectively removed by this model. Sentiment analysis was not included in this paper.

The SVM model is used by V Kranthi Sai Reddy [5], to predict the closing date of the NSE. By taking the sharp closing data as the training data to predict the future closing price.The author mainly concentrates on machine learning models for prediction, They try to find the efficiency of SVM in predicting closing price of the stock . By taking the data from different global financial markets. This study indicates that SVM high efficiency of SVM for predicting stock prices.Financial news is considered in this study.

This paper mainly concentrates on comparing machine learning models and deep learning for predicting prices. Hiransha Ma, Gopalakrishnan E.Ab , Vijay Krishna Menonab, Soman K.P They have considered[6] two types of datasets one form NSE and NYSE , In NSE they have taken one company into three sectors Automobile,Banking and IT sectors. In the NYSE they have taken the top two active stocks Bank of America (BAC) and Chesapeake Energy (CHK). For training they used Tata Motors Data and predicted closing prices for each company using ARIMA, MLP,RNN,LSTM,CNN . This study concludes that Deep learning algorithms were more dynamic for time series data and Machine learning algorithms were not dynamic and DL algorithms were outperformed. . Financial news is considered in this study and hybrid networks are not explored.

Arjun Singh and Subarna Shakya [2] performed analysis of the parameter of lookback with variants of RNN for the stock price prediction performance of the two commercial banks listed on Nepal Stock Exchange (NEPSE). It concludes with GRUs getting the highest performance (takes into account 7 look back variants). The paper does not provide a variety in comparison of pre-processing, error analysis. Sentiment influence was also not considered.

Sidra Mehtab and Jaydip Sen [7] took the Nifty 50 dataset. The paper mainly concentrates on comparing different machine learning models and LSTM models. Four different LSTM models were used, Univariant with one, two weeks previous opening values as input variables and Multivariate with one, two weeks previous open, close, high, low and volume as input variables. They have many optimization algorithms and error analysis. This study concludes that Deep learning algorithms are more accurate with time series (stock data). They also reveal the fact that multivariate inputs are less accurate than univariant inputs. Financial news is considered in this study and GRU are not explored.

Can Yang and Zhai [14] proposed a framework combining a Convolutional neural network for feature extraction and a LSTM network for prediction. They used a three-dimensional CNN for data input in the framework, including the information on time series, technical indicators, and the correlation between stock indices. And in the three-dimensional input tensor, the technical indicators were converted into deterministic trend signals and the stock indices were ranked by Pearson product-moment correlation coefficient (PPMCC). The study includes a fully connected network which was used to drive the CNN to learn a feature vector, which acted as the input of concatenated LSTM. After both the CNN and the LSTM were trained well, they were finally used for prediction in the testing set. This study indicated that the deterministic trend signals and the ranked stock indices in the three-dimensional input tensor play a significant role in improving the prediction performance.

Xuesong Yan [11] treated the financial product price data as a one-dimensional series generated by the projection of a chaotic system composed of multiple factors into the time dimension, and the price series was reconstructed using the time series phase-space reconstruction (PSR) method. In this paper, the PSR method for time series analysis was combined with a DNN-based LSTM network model which was then used to predict stock prices. They selected an optimum activation function and optimization method to optimize DNN model. This paper compared the results with different numbers of hidden layers and number of nodes in the hidden layers of the model and different activation functions(tanh and ReLU). This paper also compared ARIMA, SVR, deep MLP model, deep LSTM model with no PSR process and LSTM combined with PSR.
In [15] Dutta, Kumar and Basu propose a method where sets of exogenous and endogenous parameters are focused on. GRUs train faster and have not been used rarely for cryptocurrency predictions.

Model is compared with an LSTM network. Time series data is obtained from bitcoincharts.com. Features considered are price, daily lag, returns, volatility, transaction volume, fees, hash rate, money supply, block size. Variance Inflation factor is considered for feature selection to address multicollinearity. TanH was used for learning and ReLU for activation as it gave best results. Look back periods of 15, 30, 45 and 60 were calculated to achieve best results at 30. Aftermarket trading (weekend fluctuations) and sentiment analysis are not considered.

Ghosh and Bose [10] proposed a framework using the LSTM Model to analyze which is the best time span to predict the future share price of a company from a particular sector. Firstly, they collected the data from BSE official website. Then they preprocessed the data mainly data discretization, Data transformation, Data cleaning and Data integration. They predicted the future closing price of 5 different companies with the help of LSTM. Future prediction was done for 3 month, 6 month, 1 year and 3 years. LSTM model consists of a sequential input layer followed by 3 LSTM layers and then a dense layer with activation. They generated output using RNN and compared with target values and calculated the error difference. In this study the prediction was visualized using Keras. This study indicated that companies from a certain sector have the same dependencies as well as the same growth rate.

However, all the above-mentioned studies did not take sentiment analysis into consideration.

Xu Jiawei and Tomohiro [8] used LSTM as their base model and had taken the Chinese stock market. Author preprossed the data like feature selection and dimensionality reduction and took different inputs in LSTM like stock data, technical index and macro index. Here, stock news sentiment is used as another input. This study mainly concludes that stock news emotions can have an impact on the stock market. Author has not considered comparing many models to predict the stock data.

Sidra and Jaydip [12] proposed a hybrid approach for stock price movement prediction using ML, DL, NLP. In this study NIFTY50 daily data was collected for a period of 4 years which consists of Date, Open, High value and Low value of the Index, Close, Volume of the stock traded on a given date. Nine variables were derived and used in their forecasting models. In this study 8 approaches for classification and 8 approaches of regression were implemented. Then they used the Twitter sentiment analysis of NIFTY50 related tweets during the training and test period. And then these two inputs were fed into the fuzzy neural network based SOFNN algorithm. This study proved that public sentiments in social media serve a very significant input in predictive model building. The prediction results were the best among all the papers. However, this study did not consider Neural networks which might have given better results compared to used Machine Learning techniques.

Satish and Girivarman [13] used a system combining the LSTM for technical analysis and sentiment analysis for fundamental analysis for stock price prediction. Firstly, they collected the NSE historical data then removed the null values then removed the ‘,’ in close value. After that the dataset was split into train and test and implemented the LSTM model to predict the close price for next ‘n’ days. In the sentiment analysis they extracted Google news data using feed parser and RSS feed [Basically NLP] then used the ‘newspaper’ package to access articles. Later they got the sentiment value by comparing with lexicons and then repeated this process for every keyword. Finally the output from the technical and the sentiment analysis was compared and processed the individual output and then displayed the result. This paper showed that the proposed system will be useful for the user who is unaware of the stock market.

Tej Bahadur Shahi, Ashish Shrestha , Arjun Neupane and William Guo [8] have used LSTM and GRU to predict the stock price. Data set was taken from Nepal Stock Exchange. Stock news and stock data was scraped from the sharesansar webpage. The Authors had tried to integrate the stock news sentiment with both LSTM and GRU. By this study the author concluded that Sentiment analysis is recommended for stock prediction and can get accurate results.

V. PROPOSED METHODOLOGY

A.Workflow:

● Collection of Data (Technical Analysis):

○ Historical data of the stock is collected from Yahoo Finance Website which consists of Open, High, Low, Volume Traded, Close price of the stock.

● Collection of News Data:

○ News data is collected from Money Control and TimesofIndia website for that day of the targeting stock.

○ Data is scraped from money control using a beautiful soup library in python.

● Preparation of Sentiment Scores (Sentiment Analysis):

○ Financial Sentiment Analysis is a challenging task due to specialized language and lack of labeled data in that domain. General purpose models (Vader, TextBlob by NLTK) are not effective enough.

○ We used FinBERT, a language model based on BERT (Bidirectional Encoder Representations from Transformers) which is a state of art Machine Learning model used for NLP tasks.

○ Using the FinBERT model we achieved very good results for the correlation factor in the sentiment analysis.

● Preprocessing of the Dataset:

○ Standardization of the dataset is done. Since we have columns with different scales such as Open/High/Low will be in the range of 50 to 20000 but Volume traded might be in lakhs or crores so standardization is a must.

○ Sklearn’s preprocessing StandardScaler is used which standardizes features by removing the mean and scaling to unit variance.

● Splitting the dataset into Train,Validation and Test Data:

○ Data set is split into Training set, the model sees and learns from this data

○ The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters is known as Validation set. Hence the model occasionally sees this data, but never does it “Learn” from this. We incorporated this set also to increase the accuracy of the results.

○ While creating the train and test data we are taking the window size of 14 and 7. It is like we are seeing the previous 14 days and predicting the 1 in this way the train x any train y for prediction is prepared. This same applies for 7-day window size.

○ The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset is known as the Test set.

○ The data is now ready to be fed into the Deep Learning Algorithm which is The LSTM model.

● Training the model using LSTM:

○ Long short-term memory (LSTM) is a type of recurrent neural network that allows long-term dependencies in a sequence to persist in the network by using "forget" and "update" gates. It is one of the primary architectures for modeling any sequential data generation process, from stock prices to natural language.

○ A sequential model which is a linear stack of layers is used in our case. The first layer is an LSTM layer with 120 memory units. Since we have only one layer so no need of returning sequences (return_sequences =True is done to ensure that the next LSTM layer receives sequences and not just randomly scattered data).

○ A dropout layer is applied after the LSTM layer to avoid overfitting of the model. Finally, we have the last layer(Dense) as a fully connected layer with a 'relu' activation and neurons equal to the number of unique characters.

○ Now the model is used to predict the samples in the test data.

B.Technology:

● Python is used as a programming language in the complete project.

● As a development environment we used the Anaconda Distribution and Jupyter Notebook. We used Matplotlib for data visualization, Numpy for various array operations and Pandas for data analysis.

● Deep Learning tools/libraries for predicting the stock prices:

○ Tensorflow:

■ TensorFlow is an open-source library developed by Google primarily for deep learning applications. TensorFlow was originally developed for large numerical computations without keeping deep learning in mind. However, it proved to be very useful for deep learning development as well, and therefore Google open-sourced it.

■ TensorFlow accepts data in the form of multi-dimensional arrays of higher dimensions called tensors. Multi-dimensional arrays are very handy in handling large amounts of data.

■ TensorFlow works on the basis of data flow graphs that have nodes and edges. As the execution mechanism is in the form of graphs, it is much easier to execute TensorFlow code in a distributed manner across a cluster of computers while using GPUs.

○ Keras:

■ While TensorFlow is an infrastructure layer for differentiable programming, dealing with tensors, variables, and gradients, Keras is a user interface for deep learning, dealing with layers, models, optimizers, loss functions, metrics, and more.

■ Keras serves as the high-level API for TensorFlow: Keras is what makes TensorFlow simple and productive.

■ Building a model in keras:

● **Define a network:** In this step, you define the different layers in our model and the connections between them. Keras has two main types of models: Sequential and Functional models. You choose which type of model you want and then define the dataflow between them.

● **Compile a network:** To compile code means to convert it in a form suitable for the machine to understand. In Keras, the model.compile() method performs this function. To compile the model, we define the loss function which calculates the losses in our model, the optimizer which reduces the loss, and the metrics which are used to find the accuracy of our model.

● **Fit the network:** Using this, we fit our model to our data after compiling. This is used to train the model on our data.

● **Evaluate the network:** After fitting our model, we need to evaluate the error in our model.

● **Make Predictions:** We use model.predict() to make predictions using our model on new data

○ Sequential:

■ The core idea of Sequential API is simply arranging the Keras layers in a sequential order and so, it is called Sequential API. Most of the ANN also has layers in sequential order and the data flows from one layer to another layer in the given order until the data finally reaches the output layer.

■ Here, we have created one input layer, one hidden layer and one output layer.

■ Train and predict the model:

● Model provides a function for the training, evaluation and prediction process. They are as follows −

○ compile − Configure the learning process of the model

○ fit − Train the model using the training data

○ evaluate − Evaluate the model using the test data

○ predict − Predict the results for new input.

○ LSTM:

■ Long short-term memory (LSTM) is a type of recurrent neural network that allows long-term dependencies in a sequence to persist in the network by using "forget" and "update" gates. It is one of the primary architectures for modeling any sequential data generation process, from stock prices to natural language.

■ An LSTM module has a cell state and three gates which provides them with the power to selectively learn, unlearn or retain information from each of the units.

■ The cell state helps the information to flow through the units without being altered by allowing only a few linear interactions.

■ The forget gate decides which information from the previous cell state should be forgotten for which it uses a sigmoid function.

■ The input gate controls the information flow to the current cell state using a pointwise multiplication operation of 'sigmoid' and 'tanh' respectively.

■ Finally, the output gate decides which information should be passed on to the next hidden state.

○ FinBERT:

■ BERT was released recently (October 2018 by Google). FinBERT is BERT fine-tuned specifically for financial sentiment.

■ This model was used by us primarily due to the failure of VADER to output scores that had correlation with the open prices.

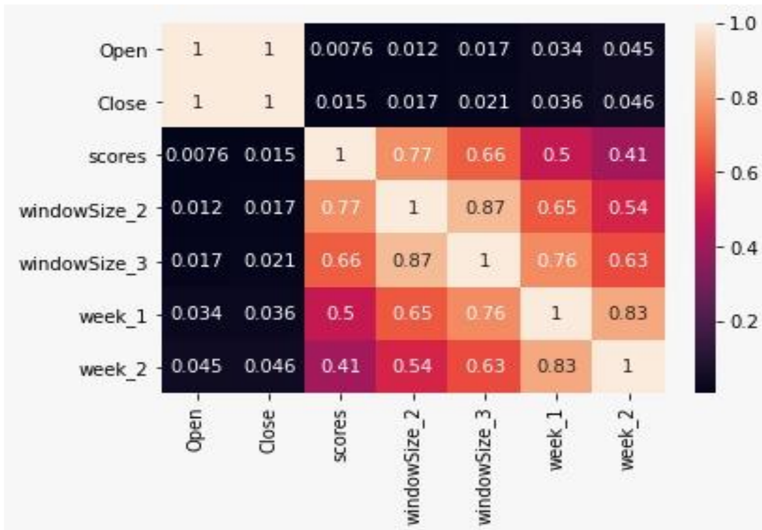
■ BERT overcomes the challenge posed by RNNs where words later in a sentence can be understood with reference to words earlier in the sentence but the reverse is not true. Even Bi-directional RNNs go left to right and right to left but these two are only concatenated and hence some amount of meaning is inevitably lost.

■ It is available pre-trained with a vocab of 30873 words, 12 hidden layers with 768 cells with an hidden activation of 'relu' and hidden dropout probability of 0.1. These are pre-determined for financial sentiments; we have not experimented with parameters or changed the model.

V. RESULTS AND OBSERVATIONS

The Data Set that has been used for training and testing is developed by downloading the data from yahoo finance and appending to it columns of various types of scores. The scores were calculated from running models on data scraped from money control. These scores are of two types. The first is the output using the nltk VADER library and the second is using the FINbert sentiment analysis model. Through the VADER model we created 4 columns. These were the lagged average scores of two days, three days, one week and 2 weeks. The average of the lagged window was taken for a particular date with equal weights for all days indicating that the scores of each date in a window were given equal importance. By analysing the correlation matrix of the columns of the VADER model we observed extremely poor correlation between the opening price that we wish to predict and the four new columns we created eg 0.0076 which suggests that this column does not provide much information about open prices.

(VADER **correlation matrix Image-1)**



We hence shifted to the FINBERT model in hope for better correlation between the columns we newly create and the opening prices. The FINBERT model is specifically developed for financial news and the correlation matrix of the four new columns we created gave a much better correlation with the opening prices than those with the VADER model. For e.g. the correlation between open price and two week window size was 0.045. Since the results were promising we further experimented with the data to create four more columns, this time these were weighted averages of the window size. The idea here was to provide greater precedence to the news that was collected more recently. For eg in the 2 day window size we can make a split of 0.666 and 0.333 for the current day and the day previous respectively. We have used a linearly decreasing function for choosing the weights. From the correlation matrix below we can see that the weighted averages share a much greater correlation with the opening price than the columns without the unequal weight distribution for e.g., the 2-week weighted window size has a correlation of 0.54 with the open price.

(**FinBERT correlation matrix Image - 2**)

A. Predictions:

1. LSTM(with and without sentiment)

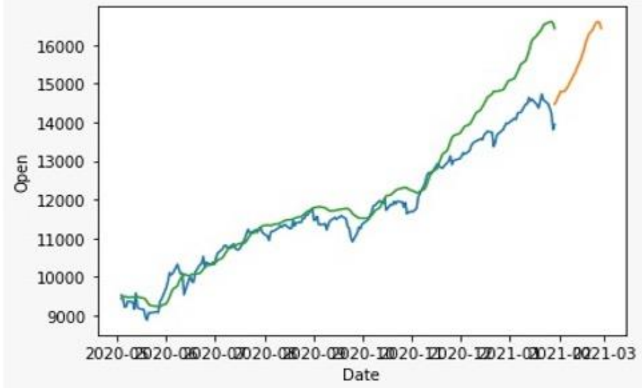
Predicted Window - 21 days
Lookback size - 7 days

Type	MSE	MAE	MAPE
Without Sentiment	647977.2137757116	630.7099513809524	0.042435809556255706
With sentiment(2week)	756663.0971496409	678.6332712857144	0.04548751787897253
With Sentiment(1week)	467037.7502019805	569.1553612857145	0.03800686726217849
With Sentiment(3days)	254628.97642213182	430.3315279523806	0.028995393415229064

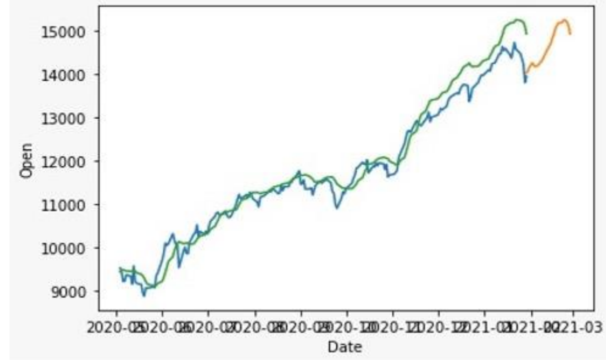
Lookback Size - 14 days

Type of Data	MSE	MAE	MAPE
Without Sentiment	864577.3038522463	689.5893031904761	0.04643926711431763
With Sentiment(2week)	473333.5418729142	583.5811730000003	0.03889286277814990
With Sentiment(1week)	349845.01349752303	514.1932422380951	0.034349073970992086
With sentiment(3days)	373853.7378489153	548.2033560476193	0.03662869951104582

Common legend - *Blue - Actual*
Green - Model Training
Yellow – Forecasts



LSTM without sentiment



LSTM window size 14days and 2week lag scores

2. ARIMA

Model	MSE	MAE	MAPE
ARIMA	2238320.1017061346	1444.2212430000004	0.09622921569431611

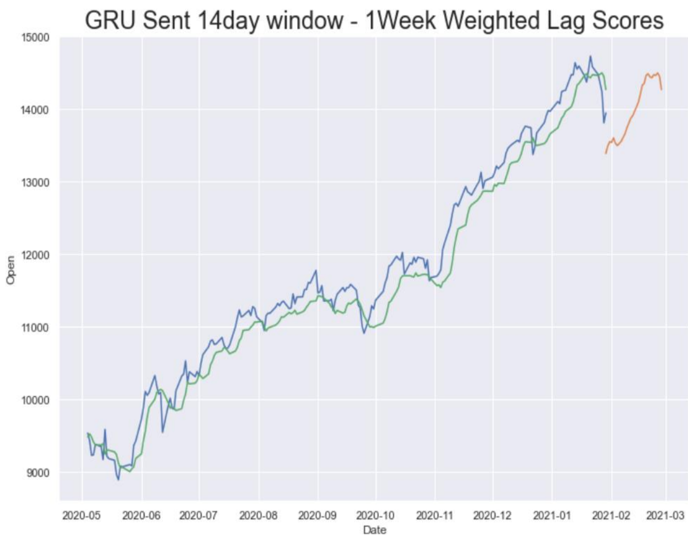
3. GRU(with and without sentiment)

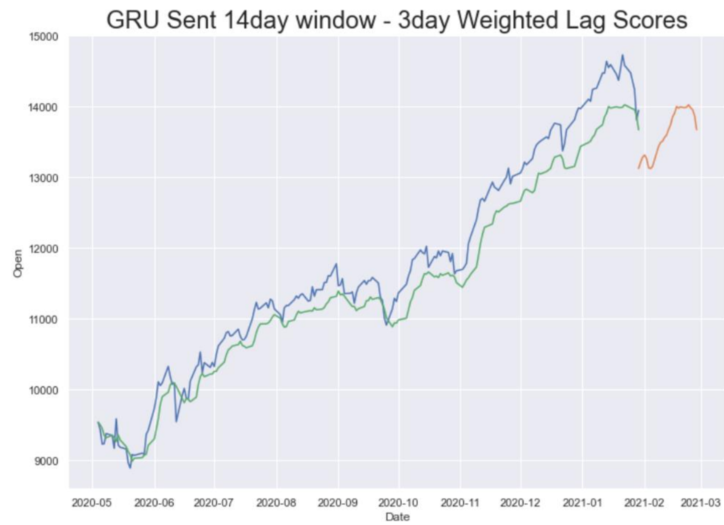
Predicted Window - 21 days
Lookback Size - 7 days

Type	MSE	MAE	MAPE
Without Sentiment	1132471.8471608334	894.4148373809522	0.060190939471106084
With Sentiment(2week)	297595.21434753574	447.38433966666656	0.029905192841074863
With sentiment(1week)	823062.1190218959	814.5386491904762	0.054311121275039564
With sentiment (3day)	654733.847610565	697.7538872857147	0.046502510106130115

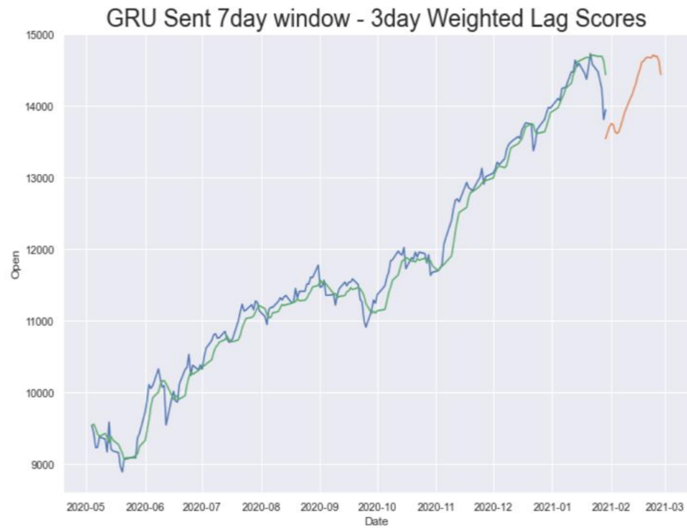
Lookback Size - 14 days

Type	MSE	MAE	MAPE
Without Sentiments	891751.0804768018	855.023553952381	0.05706105735121074
With Sentiments(2week)	700745.3886842157	741.2998292857145	0.04940032592062250
With sentiment(1 week)	738020.2481876051	756.2227206190479	0.05036917941496421
With Sentiment(3 day)	465670.8737423749	577.3670539523812	0.03847765215564279

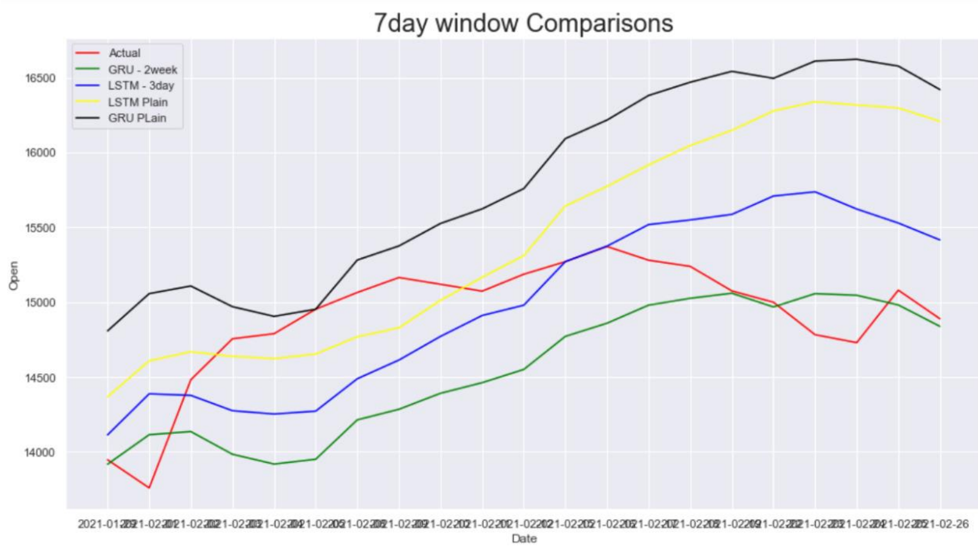


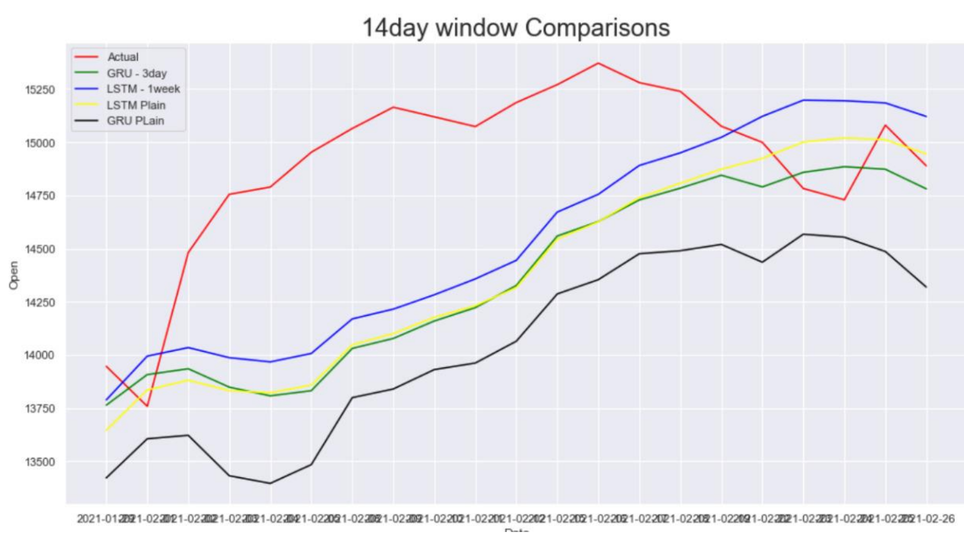


B.



Comparisons between models





VI. ANALYSIS

From the above table of LSTMs and GRUs of 7 days and 14 days window sizes, it is observed that when we insert variations of sentiment scores, we observe the error is substantially reduced in most cases. The only exception being the LSTM 7 day window size where a 2 week lag period of sentiment scores show a negative effect on the error(increases the error). The 7 day and 14 day comparison graphs have the least error predictions of GRUs and LSTMs with sentiment scores along with the actual prices and the predictions when we use plain LSTM and GRU without sentiment. The above graph is evidence that the predictions with the sentiment scores(varied lags for each model) displays lines on the graph(predictions) that are closer to the actual predictions when compared with the plain models.

VII. CONCLUSION AND FUTURE SCOPE

In this paper, we have presented a Deep Learning approach to stock price prediction. We built, fine-tuned, and then tested these models using daily historical data of NIFTY 50 during August 7, 2012 till April 16, 2021. The raw data is suitably pre-processed and suitable variables are identified for building predictive models. Recurrent Neural Network with LSTM (Long Short-term Memory) proved to handle financial time series data better than traditional time series prediction method. On the other hand market sentiment is also considered, market sentiment was caught by sentiment analysis and it proved to be a very important factor influencing stock market and thus improving prediction accuracy. Our predictions were better compared to other research work in this field since the sentiment analysis by Fin-BERT gave very good results and thus increased the prediction accuracy. Although Intra-day trading was not considered in our approach which we will be extending as a future work in this field.

VIII. REFERENCES

[1]Hegazy, Osman & Soliman, Omar S. & Abdul Salam, Mustafa. (2013). A Machine Learning Model for Stock Market Prediction. International Journal of Computer Science and Telecommunications. 4. 17-23.

[2] Saud, Arjun & Shakya, Subarna. (2020). Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE. Procedia Computer Science. 167. 788-798. 10.1016/j.procs.2020.03.419.

[3] Zen, Heiga & Agiomyrgiannakis, Yannis & Egberts, Niels & Henderson, Fergus & Szczepaniak, Przemysław. (2016). Fast, Compact, and High Quality LSTM-RNN Based Statistical Parametric Speech Synthesizers for Mobile Devices. 2273-2277. 10.21437/Interspeech.2016-522.

[4]Kumar, Jitendra & Goomer, Rimsha & Singh, Ashutosh. (2018). Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters. Procedia Computer Science. 125. 676-682. 10.1016/j.procs.2017.12.087.

[5] Vanukuru, Kranthi. (2018). Stock Market Prediction Using Machine Learning. 10.13140/RG.2.2.12300.77448.

[6] M, Hiransha & Gopalakrishnan, E. A & Menon, Vijay & Kp, Soman. (2018). NSE Stock Market Prediction Using Deep-Learning Models. Procedia Computer Science. 132. 1351-1362. 10.1016/j.procs.2018.05.050.

[7]Mehtab, Sidra & Sen, Jaydip & Dutta, Abhishek. (2020). Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models. 10.13140/RG.2.2.23846.34880.

[8]Jiawei, Xu. (2019). IMECS2019 Paper XUJIAWEI.

[9]Shahi, T.B.; Shrestha, A.; Neupane, A.; Guo, W. Stock Price Forecasting with Deep Learning: A Comparative Study. Mathematics 2020, 8, 1441. https://doi.org/10.3390/math8091441

[10]Ghosh, Achyut & Bose, Soumik & Maji, Giridhar & Debnath, Narayan & Sen, Soumya. (2019). Stock Price Prediction Using LSTM on Indian Share Market. 10.29007/qgcz.

[11]Stock price prediction based on deep neural networks Pengfei Yu Xuesong Yan

[12]Mehtab, Sidra & Sen, Jaydip. (2019). A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing.

[13]"STOCK PRICE PREDICTION USING DEEP LEARNING AND SENTIMENTAL ANALYSIS", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.7, Issue 5, page no.346-354, May-2020, Available :http://www.jetir.org/papers/JETIR2005051.pdf

[14]Can Yang, Junjie Zhai, Guihua Tao, "Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory", Mathematical Problems in Engineering, vol. 2020, Article ID 2746845, 13 pages, 2020. https://doi.org/10.1155/2020/2746845

[15] Dutta, A.; Kumar, S.; Basu, M. A Gated Recurrent Unit Approach to Bitcoin Price Prediction. J. Risk Financial Manag. 2020, 13, 23. https://doi.org/10.3390/jrfm13020023