INTRADAY STOCK TREND PREDICTION USING LSTM, ANN AND FIN-BERT

ANAND NATARAJ

Final Thesis Report

NOVEMBER 2020

TABLE OF CONTENTS

DEDICATION

I dedicate this work especially to my father, who is an intraday trader and is trying to find a pattern in the day trading for almost a decade now, which would help him to trade better to gain more profits. Moreover, I contribute this work to my friends, colleagues, stock market enthusiasts and all Data Scientists.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my thesis mentor, Mr. Vinay Katiyar and Professor Dr Manoj for their maximum guidance in completing this thesis report. Secondly, I would like to thank my father who helped in understanding the Indian intraday stock market better. Thirdly, my humble thanks to all my batch mates who helped me in clarifying my queries when needed. Finally, I would like to express my gratitude to Liverpool John Moores University for letting me in enrolling in this program.

ABSTRACT:

In this research we are aimed at predicting the intraday stock trend (positive/negative) especially, the trend between OPEN and CLOSE prices, and how to trade a stock profitably by identifying the trend. The reason for choosing the open-close price is explained in the plan of action.

In this paper, we have discussed the utilization of a sequence-based deep learning (LSTM) model for understanding the hidden patterns in the historical time-series intraday stock data, the output of this, model along with the technical indicators as the input features to an ANN model, is the predicted next day's CLOSE price of the stock.

The state-of-the-art FIN-BERT model is used for knowing the current day's sentiment of the stock based on the news and articles published in the news channels and blogs.

Note: by current day's sentiment, we mean that the sentiment predicted for the stock before the market opens for the next day.

We finally combine the next day's predicted close price and the current day's stock sentiment to decide on the next day's stock trend and based on it, if it is positive or negative trend, we buy or sell the stock respectively on the next day when the market opens.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SBI	State Bank of India
NSE	National Stock Exchange
BERT	Bidirectional Encoder Representations from Transformers
FIN-BERT	Financial-BERT
finVocab	Financial Vocabularies
ML	Machine Learning
LSTM	Long Short-Term Memory
ANN	Artificial Neural Networks
RNN	Recurrent Neural Networks
GRU	Gated Recurrent Unit
SMA	Simple moving Average
ES	Early Stopping
MSE	Mean Squared Error

CHAPTER 1

INTRODUCTION

1.1 Background of the study

The stock market is the place where the listed companies' shares are sold or bought to make profits. The name stock indicates the shares of different companies or the same company. If a share has been bought for a very less price and sold at a very high price, it indicates a profit, similarly, if a stock has been sold at a very high price and bough at a very less price, it indicates profit too. The reverse of any of these two would result in loss. One interesting fact in the stock market is that it is not necessary to buy a stock to sell, we can sell even before buying a stock.

Stock sentiments could be **Positive, Negative or Neutral** based on the information collected for a stock from news channels and web-blogs is good, bad or neutral. Similarly, **stock trend** could be **Positive, Negative or Neutral** for a given time-period.

We can say a particular time-period is a positive or negative trend by understanding the starting and the ending prices of a time-period. If the difference of starting and ending prices in a time-period is positive, then it is a positive trend and if it is negative then it is a negative trend.

Figure 1.1 Example plot to show the positive and negative trend in real time

For example, in the graph shown above the trend at the period 1 (10:15 AM to 10:40 PM) is a negative trend and the period 2 (1:20 PM to 2:30 PM) is positive trend.

Intraday in the stock market is shorthand for the stocks that trade on the markets during regular business hours and their **price** movements. During intraday, the price of a stock at which the market opens is the **OPEN** price and the price at which the market closes is the **CLOSE** price,

and in between, there are **HIGH** and **LOW** prices which is the maximum and the minimum price reach of the stock of that particular day. Both high and low prices can sometimes be the open and close prices too.

Figure 1.2 Example plot to show Open, High, Low, Close price points

Stock prices change every day by market forces. By this, we mean that share prices change because of supply and demand. If more people want to buy a stock (demand) than sell it (supply), then the price moves up. Conversely, if more people wanted to sell a stock than buy it, then there would be greater supply than demand, and the price would fall.

Understanding supply and demand is easy. What is difficult to comprehend is what makes people like a particular stock and dislike another stock. This comes down to figuring out what news is **positive** for a company and what news is **negative**. There are many answers to this problem and just about any investor you ask will have their own ideas and strategies.

But for certain, if the news published is very bad about the performance of a company it affects its stock price negatively, and vice-versa is true. This is the reason why we are very much interested in capturing the everyday news/blog article (from the web) sentiments of the stock, which would contribute greatly to predicting the intraday trend.

1.2 Problem Statement

To predict the highly fluctuating and known to world the unpredictable stock price trend for the next day in intraday market, and by doing so we are trying to trade the stock profitably.

1.3 Scope of the study

The experiment is conducted only by focusing a particular stock (SBI-NSE) and the ML techniques used are limited to two: LSTM and FIN-BERT in finding the next day's trend, which is our primary task.

1.4 Aim of the study

The main goal is to build a model which could predict the trend (positive/Negative) between the open and close prices of the stock for the next day's intraday market - using the LSTM model for identifying the pattern in the historical time-series stock data and FIN-BERT model to understand the sentiment of the stock using the current days new/blog articles from the web. Using the trend predicted between the Open and Close price for the next day we are trying to trade the stock profitably by buying or selling the stock with respect to the trend (buy if the trend is positive and sell if the trend is negative).

1.5 Research Questions

- Is it possible to track a particular stock's trend (positive/negative) for the next day in intraday market, especially between the open and close prices by analyzing the historical price data and understanding the today's sentiment of the stock?

- Is it possible to trade profitably the stock by understanding the open-close price trend for the next day?

1.6 Significance of the study

Trading stocks profitable in intraday can introduce huge profits in a short period of less than six hours (Intraday market active time). In our stock trend prediction task using LSTM and FIN-BERT several experiments were conducted (training using the historical data and fine-tuning hyperparameters) to conclude the best model for our application.

1.7 Structure of the study

The study contains literature review and research methodologies; each topic is mentioned as chapters in this report.

Chapter-2 describes literature review of existing techniques about the stock tend/price predictions. The systematic literature review has been performed by referring various conference papers, journals, articles and books related the methodologies for time-series data training and prediction and sentiment analysis for the given finance related news/blog articles.

Chapter-3 elaborates the research methodology and further we have discussed in detail about the data pre-processing techniques which is the input to the LSTM (Historical time-series stock data) and the FIN-BERT (news/blog articles relevant to the stocks obtained from google alerts).

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The stock price prediction problem has been tried solving using the conventional machine learning algorithms earlier. Several experiments have been done in the past to explore the stock data, but till date, there is no robust model as per our knowledge which would help in trading a stock profitably. Our try is not to predict the prices exactly but to predict the trend, which would be comparatively easier than stock price prediction.

2.2 Existing approaches and the papers referenced for our work

We have explained below in detail the papers that we have referred to choose the models that we have mentioned in the plan of action for achieving our tasks. Moreover, our entire research is an improvement of ideas mentioned in this paper [18], further at the end, we have mentioned our novel contributions to keep this entire research unique. We have asked several questions below and answered the same with the proof of the research papers to keep it interesting.

2.2.1 Why we have opted for deep learning models for our task?

As rightly concluded in this paper [8] it would be optimal to choose Neural networks for stock predictions since the quantity and co-relation among the historical stock data is very high. Maybe the length (time steps) might not be very large as mentioned in the paper since we are not exceeding 3 months (90 steps) of a window in our work, but the other two aspects of the number of the data available and the co-relation among the time-series data (open, close, high, low) is very high.

2.2.2 Why we have narrowed down to LSTM for time-series model?

After deciding to work with neural networks we have chosen to work with the Recurrent Neural Networks which is one of the major classes in neural networks and are very powerful for treating the sequence and time-series data as mentioned in these papers [10, 11, 12]

In the early vanilla RNN's proposed in the '90s [9] were bound to two main problems vanishing and exploding gradients [13], which prevented from maintaining the long-term memory which was addressed by the invention of LSTM - These two papers represent the earliest invention of LSTM [4] and its variance [14].

A similar structure to LSTM was proposed lately in 2014 known as GRU [15] to address the same problem of vanishing and diminishing gradients, but these papers prove that LSTM is

strictly stronger than GRU as it can easily perform unbounded counting, while the GRU cannot. That's why the GRU fails to learn simple languages that are learnable by the LSTM [16].

Similarly, as shown by Denny Britz, Anna Goldie, Minh-Thang Luong and Quoc Le of Google Brain, LSTM cells consistently outperform GRU cells in "the first large-scale analysis of architecture variations for Neural Machine Translation."[17].

Moreover, this paper [2] demonstrates that the LSTM model outperforming the traditional model, especially in our scenarios which is stock price perdition.

So as per our analysis, we could see that LSTM is one robust RNN variant capable of handling vanishing and diminishing gradient problem and retain the long-term memory along with the short-term memory of the sequence.

2.2.3 Papers helped in training and tuning the LSTM for our task

This paper [24] helped on how to practically select the features and train an LSTM model for stock prediction and these papers [4,5,6,7] has helped in understanding the nuances of LSTM to improve the accuracy of the model.

Referred the past works on stock prediction using LSTM to understand the actual performance ability of it especially in stock prediction [1, 2, 3, 19, 20, 21, 22, 23]

2.2.4 Why NLTK-BERT is the state-of-the-art model for sentiment analysis?

We have chosen NLTK-Bert for sentiment analysis since as mentioned in this paper [25] Bert could able to outperform most of the state-of-the-art sentiment analysis models, and this paper [26] gave an idea behind understanding Bert for sentiment analysis.

Similar works done in the past are using the BERT sentiments as a feature to the deep learning models to improve the accuracy stock price prediction [27,28, 29, 30].

2.2.5 why FIN-BERT is the most needed model for our task?

But we are not going to use directly the base BERT model for our sentiment analysis purpose, instead, we are going to use FIN-BERT [31], a finance domain specific model trained on top of BASE-BERT using a large financial communication corpora of 4.9 billion tokens, including corporate reports, earnings conference call transcripts and analyst reports.

Below are the accuracy differences, as mentioned in the paper [31], while trying to classify the given sentiments are positive, negative or neural on the datasets: PhraseBank, FiQA, AnalystTone.

Dataset	BASE-BERT accuracy	FIN-BERT accuracy
PhraseBank	0.835	0.872
FiQA	0.730	0.844
AnalystTone	0.850	0.877

Table 2.1 Comparative accuracy report on the Base/Fin-BERT for various datasets

2.2.6 Similar work in the past that is resembling our research

Apart from all the literature reviews, our entire research methodology is based on the improvement of Nisha Shetty's paper on "Indian Stock Market Prediction Using Machine Learning and Sentiment Analysis" [18] where the author has combined the prediction results of the traditional ML model and the output of the preliminary sentiments analysis model for deciding on buy/sell trend.

2.2.7 Our novel ideas to keep the research unique and more efficient

The improvement in our model is in using the state-of-the-art deep learning model (LSTM) with technical indicators as the additional features to the ANN. Moreover, the usage of the state-of-the-art FIN-BERT for sentiment analysis has made our work unique. In addition, for sentiment analysis we are not dependent on one blog articles, as mentioned in the existing paper above [18], rather, we are going to collect the information from all the relevant blogs and news channels using google alerts for sentiment analysis.

2.3 Summary

In this chapter, we have discussed in detail on the ML methodologies that has been used in the past to predict the stock prices. Secondly, by analyzing the latest works in this field, we understood that LSTM and Fin-Bert are the state-of-the-art ML models to predict the time-series and sentiments respectively, and none of the papers that we have encountered so far has the combination of both these models in predicting the stock prices, and further more to ensure the uniqueness of our research we are using the ANN to combine the output of LSTM model with the technical indicators to improve the stock price prediction accuracy.

CHAPTER 3

PLAN OF ACTION

3.1 Introduction

We could categories the overall plan into three modules: First module involves building LSTM and ANN model for predicting the next days close price. Second module involves building a FIN-BERT model to predict the stock sentiment based on the current days new/blog articles from internet using google alerts. Third module involves combining the outputs of the first and the second module via fuzzy logic and trading it profitably.

3.2 First Module

We are going to build a robust LSTM model which could predict the pattern for the CLOSE price historical time-series stock data.

The reason for choosing Close price prediction as mentioned above is as follows:

There are 6 periods in intraday where we could trade,
1. Between Open and High (positive trend)
2. Between Open and Low (negative trend)
3. Between Open and Close (can be a positive or negative trend depending on which price is greater)
4. Between Low/High and High/Low (can be a positive or negative trend depending on which price hits first)
5. Between High and Close (negative trend)
6. Between Low and Close (positive trend)

For us to trade between (4, 5, 6) regions we have to accurately predict the price of both the starting and the ending points, but in the first three regions, it is enough to predict the price of the ending points because the starting point is the open price which is known at the time of market opening (we can trade only after the market has been opened, so predicting the open price before the market has opened has no significance).

By trading between the first three regions (open-high, open-low, open-close) the risk of the wrong prediction has just reduced to half since we are not going to predict the starting point (Open price). And among the first three regions, open-close is the most optimal period to trade in because we do not know the time at which the high or low price hits, and it would be tedious to predict the time and the high/low price together, whereas, in the close price we could say that it hits every day at 3 PM in the intraday market (the time when the NSE market closes) therefore, it is easier to automate our algorithm to buy or sell stock at the closing time (which is known) instead the time (which unknown) at which the price hits the high or low price. This is the reason why we have chosen Close price prediction in the first place.

Dataset for training LSTM: historical data of the stock from Yahoo finance website.

The input to LSTM for prediction could either be the past days close price or the combination of any prices, for example, to predict the next day's CLOSE price in LSTM we could either input the historical prices of the previous day's (CLOSE price) or the previous days (CLOSE & LOW) or (HIGH & LOW) or (HIGH, CLOSE & LOW) or in any combination of all four prices, but based on the test results we could see that the combination of all four prices (HIGH, LOW, OPEN & CLOSE) as the input feature had produced better results.

The output of the LSTM model along with the technical indicators** is the input features for the ANN model which would output the nearest next day's stock CLOSE price.

** Technical indicators are heuristic or pattern-based signals produced by the price, volume, and/or open interest of a security or contract used by traders who follow technical analysis. By analyzing historical data, technical analysts use indicators to predict future price movements [32]. The technical indicator that we have used is SME*** (Simple Moving Average), which is one of the simplest, most successful and widely used technical indicator.

***A simple moving average (SMA) calculates the average of a selected range of prices, usually closing prices, by the number of periods in that range. The period or the time-period taken in our case is 5 (1 week including holidays) which is the mostly used standard.

3.3 Second Module

The present day's stock sentiment is analyzed by using the Fin-BERT model. The inputs for this model are the news and blog articles captured today (by today we mean before the next day's market open) for the stock via google alerts. The output of the Fin-BERT model is the input for the sentiment fuzzy logic module to understand the stock sentiment.

We are not going to train the FIN-BERT model since we are going to use the pre-trained Bert model that can predict the sentiment of web-blog and news articles.

For Sentiment Analysis of stock on daily basis, we are planning to use 2 main sources and they are: News channels and web-Blogs

Since all news channels now have websites, and the web-blogs are websites itself we do not need any other sources other than Google alerts to capture the information about the stocks.

Once the data has been captured on daily basis the data has been processed and sent to the Fin - BERT model for sentiment analysis, here we are going to capture three types of sentiments, namely: positive, negative and neutral.

The sentiments are analyzed for each article captured from blogs and news sites. Based on the number of sentiments and their percentage the overall stock sentiment is judged for the day in the sentiment fuzzy logic module. For example, if the number of articles captured by google alerts for the stock is 5 and the sentiments analyzed for each article is: positive, positive, positive, negative, and neutral (positive = 60%, negative = 20%, neutral = 20%) then it would be categorized as positive sentiment overall. A complete sentiment fuzzy logic rules on deciding present day's stock sentiment is on below:

Sentiment Fuzzy module Logic:

Rule 1:

If (% of Positive sentiments collected) > (% of Negative and Neutral Sentiments collected):
Then the present days stock sentiment = POSITIVE

Rule 2:

If (% of Negative sentiments collected) > (% of Positive and Neutral Sentiments collected):
Then the present days stock sentiment = NEGATIVE

Rule 3:

If (% of Neutral sentiments collected) > (% of Positive and Negative Sentiments collected):
Then the present days stock sentiment = NEUTRAL

Rule 4:

If (% of Neutral sentiments collected == % of Positive sentiments == Negative Sentiments collected):
Then the present days stock sentiment = NEUTRAL

Rule 5:

If (% of Negative sentiments collected == % of Positive sentiments not= Negative Sentiments collected):
Then the present days stock sentiment = NEUTRAL

Rule 6:

If (0 % of Positive sentiments collected and (% of Negative sentiments == Neutral Sentiments collected)):
Then the present days stock sentiment = NEGATIVE

Rule 7:

If (0 % of Negative sentiments collected and (% of positive sentiments == Neutral Sentiments collected)):
Then the present days stock sentiment = POSITIVE

Rule 8:

If (0 % of Neutral sentiments collected and (% of positive sentiments == Negative Sentiments collected)):
Then the present days stock sentiment = NEUTRAL

Rule 9:

If (0 % of Negative sentiments collected == % of Positive sentiments == Negative Sentiments collected):
Then the present days stock sentiment = NEUTRAL

An example scenario describing all the fuzzy rules above:

Positive Sentiment	Negative Sentiment	Neutral Sentiment	Present Day's Stock Sentiment
60%	20%	20%	Positive
20%	60%	20%	Negative
20%	20%	60%	Neutral
33%	33%	33%	Neutral
40%	40%	20%	Neutral
0%	50%	50%	Negative
50%	0%	50%	Positive
50%	50%	0%	Neutral
0%	0%	0%	Neutral (no info on the stock)

Table 3.1 Example chart on how decision made for present day's stock sentiment

3.4 Third Module

we are going to input the output of the ANN and Fin -BERT models to an overall fuzzy logic module to judge the next day's overall stock trend and based on this trend the option of buy and sell the stock is decided.

Overall Fuzzy module Logic:

Rule1:

If (next day's (OPEN < CLOSE)) and (Present Day's (Stock Sentiment) = Positive):
Then, next day's predicted stock trend = POSITIVE

Rule2:

If (next day's (OPEN > CLOSE)) and (Present Day's (Stock Sentiment) = Negative):
Then, next day's predicted stock trend = NEGATIVE

Rule3:

If (next day's (OPEN < CLOSE)) and (Present Day's (Stock Sentiment) = Neutral):
Then, next day's predicted stock trend = POSITIVE

Rule4:

If (next day's (OPEN > CLOSE)) and (Present Day's (Stock Sentiment) = Neutral):
Then, next day's predicted stock trend = NEGATIVE

In fuzzy logic module we are not going to consider any other scenarios other than the above four because in any other scenarios the sentiment and open-close trends contradict with each other, for example:

If (next day's (OPEN > CLOSE)) and (Present Day's (Stock Sentiment) = Positive):
Then, next day's predicted stock trend = OTHER. Here, (OPEN > CLOSE) trend is negative whereas the sentiment trend is positive, so in the other scenarios like this we are not going to trade on that particular day.

Based on the fuzzy module output we are going to place the buy/sell option in two places, one is at the market open and another one at the market close:

1. If it is a **POSITIVE** trend, then buy the stock at market open and sell at market close.
2. If it is a **NEGATIVE** trend, then sell the stock at market open and buy at market close.
3. If it is **OTHER** than the above two trends, then there is no trading action taken for the day.

3.5 Summary

In module one, we are going to train an LSTM model efficiently using the historical data to predict the next days close price. Further, the predicted next days close price has been combined with the technical indicators and sent as an input to the ANN model, that has been trained on LSTM outputs and technical indicators for the historical data, to improve the accuracy of the predicted next days close price.

In module two, we are going to use the per-trained FinBert model to predict the sentiment about the stock, before the next day's market opening, based on the news articles published

In module three, we are combining the predicted close price and the sentiment to take action on either to buy or sell a stock when the market open for the next day.

3.6 Logical flow of the entire system

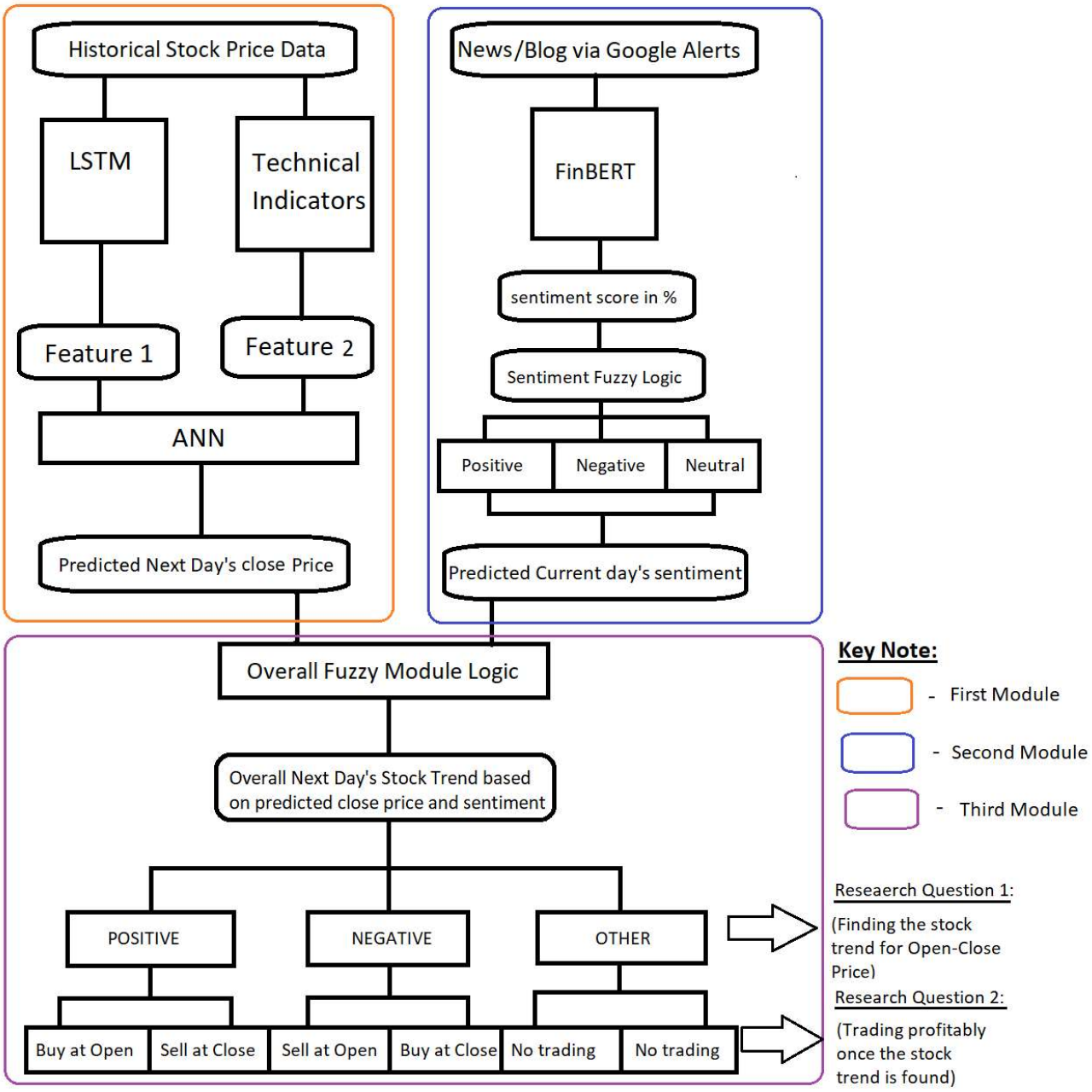


Figure 3.1 Logical flow of the complete plan

CHAPTER 4

RESEARCH METHODOLOGY

4.1 Introduction

In this session, we introduced the methodology for building a model to achieve our task. Detail description of the architecture of the models trained, the datasets over which we are going to evaluate, data pre-processing and the evaluation metrics used to check the quality of results are mentioned here.

4.2 Research Methodology

4.2.1 Data Selection

As mentioned in the plan of action earlier we have used three different models in achieving our tasks and the respective data for these three models and its sources are mentioned in the table below.

Models	Data	Source
LSTM	Historical Time-series stock data	Yahoo Finance website
ANN	To get better accuracy for LSTM output by combining technical indicators (SMA)	SMA values are calculated manually using close price
FIN-BERT	Parsed news/blog articles	Google-alerts

Table 4.1 Respective model datasets and its sources

4.2.1.1 Getting the historical time-series SBI-NSE stock data for LSTM modelling

The historical time-series stock data is very straight forward to obtain which can be downloaded easily from 'yahoo.finance' for any given time interval. In our case we have obtained all the available SBI-NSE data, the maximum data which we could obtained is for the past 20 years.

Below is the SBI-NSE dataset used for training and evaluating the LSTM and ANN models.

File Name: sbi_dataset.csv

Cloud Path:
https://drive.google.com/file/d/1CSpm1W_NCcYej1iBMNZ9FdL73Zg8M5EX/view?usp=sharing

4.2.1.2 Getting the relevant article data from news/blog sites for FIN-BERT

Since we are not going to train a Fin-Bert model we have collected only the test dataset, manually for the days from 21-09-2020 to 06-11-2020, total 34 days.

The active market days in the above-mentioned date range are as below, total 34 days, the days when the market is on leave due to weekend or a holiday is not included below:

9/21/2020	9/22/2020	9/23/2020	9/24/2020	9/25/2020	9/28/2020
9/29/2020	9/30/2020	10/1/2020	10/5/2020	10/6/2020	10/7/2020
10/8/2020	10/9/2020	10/12/2020	10/13/2020	10/14/2020	10/15/2020
10/16/2020	10/19/2020	10/20/2020	10/21/2020	10/22/2020	10/23/2020
10/26/2020	10/27/2020	10/28/2020	10/29/2020	10/30/2020	11/2/2020
11/3/2020	11/4/2020	11/5/2020	11/6/2020		

In the list above we have highlighted 8 days in **Red** and these are the days we have missed collecting the google alerts data, So below is the list of days for which we are going to predict the sentiments, total 26 test data:

9/21/2020	9/22/2020	9/23/2020	9/25/2020	9/28/2020	9/29/2020
10/5/2020	10/6/2020	10/7/2020	10/8/2020	10/9/2020	10/12/2020
10/13/2020	10/14/2020	10/16/2020	10/20/2020	10/22/2020	10/23/2020
10/26/2020	10/27/2020	10/28/2020	10/29/2020	10/30/2020	11/3/2020
11/4/2020	11/6/2020				

Below are the steps involved in obtaining the data for sentiment analysis:

To obtain the news/Blog article data for a particular stock we have to set a google alert for that company and the stock. Once the google alert is set, we will be provided with that particular google alerts feed URL, the feed URL for the alert set for “State bank of India” and “SBI-NSE” in our case is below:

https://www.google.com/alerts/feeds/15354403877403381725/10792550034617801620

With the help of ‘feedparser’ module in python we can parse the web-request returns for this feed URL and get the list of all the latest blog and news articles present in the web for “State bank of India” and “SBI-NSE” alerts

Below is the exact output (at once we will get 20 outputs when a web-request for the google alert feed URL is made, but for sample we have displayed only 1 below, the number of alerts to return at a time when the request is made can be customized in google alerts) of what we would get after feed-parsing the google alert’s feed URL:

*{'**published**': time.struct_time(tm_year=2020, tm_mon=8, tm_mday=31, tm_hour=8, tm_min=26, tm_sec=15, tm_wday=0, tm_yday=244, tm_isdst=0),*

*'**summary**': 'Among PSU banks, SBI remains the best play on the gradual recovery in the Indian economy, with a healthy PCR, robust capitalization, a strong liability ...',*
'title': 'Buy State Bank of India, target price Rs 250: Axis Securities',
***'website-url'**:*
https://www.google.com/url?rct=j&sa=t&url=https://economictimes.indiatimes.com/markets/stocks/recos/buy-state-bank-of-india-target-price-rs-255-icici-securities/articleshow/77848364.cms&ct=ga&cd=CAIyGmY2OTE4ODVmMjNlYjY3NGE6Y29tOmVuOlVT&usg=AFQjCNFTg5ZhWde6cS8iwCDfU9G917AFNA'},

We get dictionaries as outputs for each alert returned while requesting the google alert feed URL. The keys and values are published date of the article, summary of each article (containing the header and few lines from the paragraphs of an article), and finally the web-URL which takes to the site where the article is posted.

The website-url from the above dictionary after feed-parsing is a google redirect link and not the actual website link in which the article is present. We directly cannot parse the article content by using this re-direct link, so we are requesting these re-direct links using the python 'request' module and from its response (along with the original sites URL it will contain lot of metadata) we can obtain the original sites URL by using regular expression for URL matching.

The original website URL obtained by parsing the above mentioned re-direct link in the dictionary (website-url) is mentioned below.

https://economictimes.indiatimes.com/markets/stocks/recos/buy-state-bank-of-india-target-price-rs-255-icici-securities/articleshow/77848364.cms

Now we have to parse (using 'beautifulsoup' python module) the response obtained after web-requesting the actual site URL using the python 'request' model to get the full content of the article, by doing so we now have the raw data of an article for sentiment analysis.

Note: While sending a web-request using the python 'request' module either to the re-direct URL or to the actual URL there are chances where the websites can block our request as a security measure or if we are requesting it for several times it may be considered as security threat. If we have gotten bad response then it is not possible to scrape the article data from that website, so in these cases we will be using the feed summary as mentioned in the dictionary (obtained after feed-parsing the google-alerts feed URL) above for sentiment analysis.

4.2.2 Data pre-processing

4.2.2.1 Data Pre-processing for historical-time series stock data

The only thing that has to be taken care off in pre-processing the historical stock data is to remove NULL values in the data since during the weekends and government holidays the market will remain closed and the prices listed for those days will be NULL. So, we have to make sure that we are removing the rows in which the prices are not listed for the day, which could be achieved easily by utilizing the filter option in Excel, just filtering out the NULL values rows and removing it.

Below is the dataset after removing the NULL rows.

File Name: sbi_dataset_preprocessed.csv

Cloud Path:

https://drive.google.com/file/d/1uwWIyP8gnuuqcaszEgbW57XI7QyTAEZr/view?usp=sharing

If the digits mentioned in any of the columns in the dataset above are more than four numbers, there is a chance that the comma is introduced in between, example: 1,234. So, we are performing a simple check using pandas in identifying the comma character (,) in the entire dataset and replacing it with a blank character (‘’).

4.2.2.2 Data Pre-processing for articles scrapped from web

The raw data of the articles obtained from websites/feeds will contain lot of HTML tags even after parsing since the webpages are HTMLs itself. So, we are using ‘htmltotext’ python module to convert all the article data into human readable text. Below is the pickled pre-processed data for the articles captured using google-alerts for the dates mentioned in 4.2.1.2

Cloud Path:
https://drive.google.com/drive/folders/1_8p38eu6NLRDBbXsND6WEVDajFQ1rjsG?usp=sharing

4.2.3 Data transformation

4.2.3.1 Historical stock price data transformation

For historical stock price data, we have done min-max scaling as per the below formula so that the data ranges between 0 to 1.

$$X_{sc} = (X - X_{min}) / (X_{max} - X_{min})$$

Further, we are grouping the prices into 60-time steps, so that it can be inputted as the discrete time-series data for both training and testing the LSTM model.

A Time series is a collection of data points indexed, listed or graphed in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus, it is a sequence of discrete-time data.

Based on the dimension of the Independent data we can either perform univariate time series forecasting or multivariate time series forecasting.

Univariate time series data: Only one independent variable is considered, which is varying over time. For example, taking **either one** of the high, low, close or open price in predicting the next days close price.

Date	Close Price	Grouping in 1-time steps	Grouping in 2-time steps	Grouping in 3-time steps
4/28/2020	10	[[10], [20]]	[[10, 20], [30]]	[[10, 20, 30], [40]]
4/29/2020	20	[[20], [30]]	[[20, 30], [40]]	[[20, 30, 40], [50]]
4/30/2020	30	[[30], [40]]	[[30, 40], [50]]	
5/4/2020	40	[[40], [50]]		
5/5/2020	50			

Table 4.2 Example of single-variate time series data grouped for given time-steps

Multi-variate time series data: Multi-independent variables are considered, which are varying over time. For example: taking open, high, close, low prices **combined** in any order (high & low; low & close; high, close, low & open; …etc) to predict the dependent variable which in our case is close price.

Date	Close Price	Open Price	Grouping in 1-time steps	Grouping in 2-time steps	Grouping in 3-time steps
4/28/2020	10	1	[[[10], [1]], [20]]	[[[10, 20], [1, 2]], [30]]	[[10, 20, 30], [1, 2, 3]], [40]]
4/29/2020	20	2	[[[20], [2]], [30]]	[[[20, 30], [2, 3]], [40]]	[[20, 30, 40], [2, 3, 4]], [50]]
4/30/2020	30	3	[[[30], [3]], [40]]	[[[30, 40], [3, 4]], [50]]	
5/4/2020	40	4	[[[40], [4]], [50]]		
5/5/2020	50	5			

Table 4.3 Example of multi-variate time series data grouped for given time-steps

Note: **Red** digit denotes independent data and **Green** digits denote dependent data in above tables

As per our experiment with several hyperparameter tuning we have obtained the best model for next days close price prediction with 60 time-steps and all four prices (high, low, close, open) as independent variables together (multi-variate). We have mentioned in detail on the list of hyper-parameters tested in the below session (4.2.4.2)

4.2.3.2 News/Blog articles data transformation

Once the articles are collected from news/blogs from websites for sentiment analysis we have to de-structure the data so that the maximum sequence length of each paragraph is not exceeding 512 characters (512 is the maximum sequence length which the pre-trained FIN-BERT model can handle).

If a single paragraph is exceeding max-seq-length of 512 then we will split the sentences in that paragraph using NLTK-punkt (sentence tokenizer) and combining it one by one, while doing this we will ensure that the final para obtained by combining the sentences is not exceeding 512 characters, if any of the last sentences by combining it crosses the max-seq-length then we will add it in the next paragraph. By performing this activity, we will have a list of paragraphs for each article not exceeding max-seq-length of 512 characters.

Finally, these lists of paragraphs collected for each article is tokenized using the BertTokenizer' so that the input data is now optimal for our FIN-BERT model to predict the sentiments for each paragraph in an article.

4.2.4 Modelling and Evaluation

4.2.4.1 LSTM modelling

We have taken the last 34 days data as testing set (this include the date range from 21-9-2020 to 06-11-2020 as mentioned in session 4.2.1.2) and the last before 22 days data as validation set.

Note: we have used 22 days of testing and validation data which is equivalent to a month's data including holidays.

Period	Dataset type	Count
1/1/1996 – 11/06/2020	Training Set	6141 days
20/08/2020 – 18/09/2020	Validation Set	22 days
09/21/2020 – 11/06/2020	Testing Set	34 days

Table 4.4 SBI Dataset split up for Training, Validation and Testing.

We have used the keras framework for implementing the LSTM algorithm. LSTM model is created by training the LSTM algorithm on the training set using the open, close, low, high prices combined as the independent variables and the next day's close price as the dependent variable.

4.2.4.2 LSTM model evaluation

Below is the table listed with the various parameter sets with which the model has been trained on and we could see that the best minimum test loss has been obtained with the parameter set 25.

CLOSE PRICE	P1	P2	P3	P4	P5	P6	P7	P8	P9	Early Stopping	Test Loss
parameter_set_1	16	50	5	5	3	0.2	30	adam	output_clm	64 epochs	71.4887
parameter_set_2	32	50	5	5	3	0.2	30	adam	output_clm	103 epochs	61.4053
parameter_set_3	64	50	5	5	3	0.2	30	adam	output_clm	19 epochs	322.226
parameter_set_4	32	100	5	5	3	0.2	30	adam	output_clm	61 epochs	62.8386
parameter_set_5	32	150	5	5	3	0.2	30	adam	output_clm	63 epochs	60.896
parameter_set_6	32	200	5	5	3	0.2	30	adam	output_clm	112 epochs	73.18662
parameter_set_7	32	150	10	5	3	0.2	30	adam	output_clm	11 epochs	339.119
parameter_set_8	32	150	3	5	3	0.2	30	adam	output_clm	111 epochs	56.2679
parameter_set_9	32	150	2	5	3	0.2	30	adam	output_clm	46 epochs	64.7087
parameter_set_10	32	150	3	10	3	0.2	30	adam	output_clm	82 epochs	59.9803
parameter_set_11	32	150	3	3	3	0.2	30	adam	output_clm	80 epochs	63.9117
parameter_set_12	32	150	3	5	6	0.2	30	adam	output_clm	36 epochs	66.9614
parameter_set_13	32	150	3	5	2	0.2	30	adam	output_clm	72 epochs	55.734
parameter_set_14	32	150	3	5	1	0.2	30	adam	output_clm	102 epochs	56.273
parameter_set_15	32	150	3	5	2	0.3	30	adam	output_clm	103 epochs	56.7295
parameter_set_16	32	150	3	5	2	0.4	30	adam	output_clm	54 epochs	81.8922
parameter_set_17	32	150	3	5	2	0.1	30	adam	output_clm	28 epochs	72.0069
parameter_set_18	32	150	3	5	2	0.2	60	adam	output_clm	89 epochs	56.4987
parameter_set_19	32	150	3	5	2	0.2	90	adam	output_clm	40 epochs	77.9071
parameter_set_20	32	150	3	5	2	0.2	20	adam	output_clm	79 epochs	66.7829
parameter_set_21	32	150	3	5	2	0.2	30	rmsprop	output_clm	11 epochs	92.3169
parameter_set_22	32	150	3	5	2	0.2	30	sgd	output_clm	0 epochs	1434.4744
parameter_set_23	32	150	3	5	2	0.2	30	adam	Open,High,Low,Close,Volume	246 epochs	53.829
parameter_set_24	32	150	3	5	2	0.2	30	adam	High, Volume	55 epochs	73.8788
parameter_set_25	32	150	3	5	2	0.2	30	adam	Open,High,Low,Close	183 epochs	26.1711
Best Hyper Params set is 25	32	150	3	5	2	0.2	30	adam	Open,High,Low,Close	183 epochs	26.1711

Table 4.5 List of hyper-parameters and respective test loss for LSTM model

Keynote		Variables used in Coding
Batch Size	P1	batch_size
Number of nodes in a LSTM hidden layers	P2	neural_units_LSTM
Number of LSTM hidden layers	P3	hidden_layers_LSTM
Number of nodes in the dense hidden layers	P4	neuron_units_dense
Number of dense hidden layers	P5	hidden_layers_dense
Dropout percentage for regularisation	P6	dropout
Time steps	P7	tim_steps_params
Optimizer	P8	optimizer
Open, Close, High, or Low prices considered	P9	clms_cnsdrd_params

Table 4.6 Keynote to understand table 4.5

Fixed Parameters		Variables used in Coding
Number of epochs	1000	Epochs
Early stopping patience	50	ES_patience
testing data count	22	testing_data_count
validation data count	22	validation_data_count
Metric used for loss calculation	Mean squared Error	loss

Table 4.7 Fixed hyper-parameters while LSTM training

Apart from few hyper-parameters that we have tuned we have also fixed few hyper parameters that is listed in the above table. We have set 1000 epochs along with early stopping of 50 patience, that is, is the validation loss did not converge after 50 epochs the training will stop automatically, this has a huge significance in avoiding overfitting and saving time.

Below is the weights obtained while training the model with the hyper-parameter set 25

File Name: best_weights.h5

Cloud Path:

https://drive.google.com/file/d/1-6iUVKURbFNHrF0c2U6VLHZA6cMVaBWU/view?usp=sharing

4.2.4.2.1 LSTM model training result with the best hyper parameters

Below is the graph plotted for training and validation losses with respect to epochs while training the model with the best hyper-parameter set

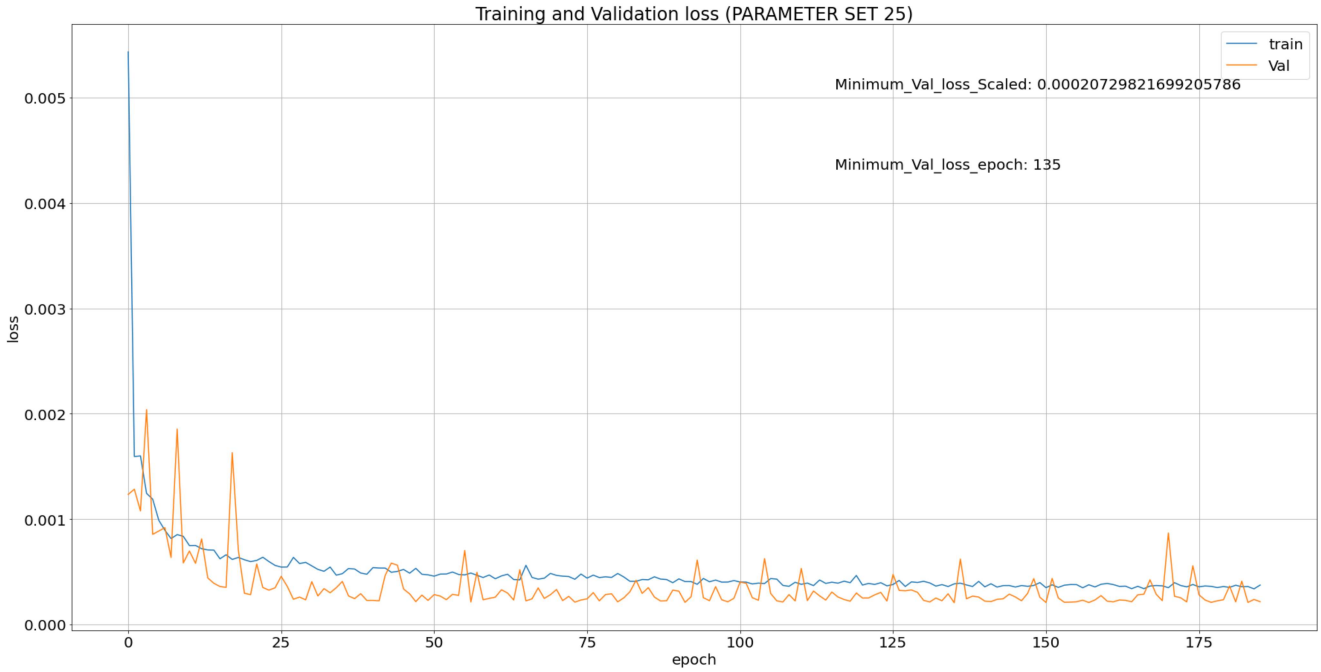


Figure 4.1 LSTM Training and validation loss convergence with respect to epochs

Note: we obtained the minimum validation loss at epoch 135 and the training stopped at 185th epoch since we have set the early stopping patience to 50, and since there is no convergence in validation loss from 135th epoch to 185th epoch the training has been stopped.

Note: The loss mentioned in the Graph is the scaled value.

Below is the graph plotted for the predictions in testing data using the model trained with the best hyper-parameters (we have also plotted the ground truth for reference).

4.2.4.2.2 Best hyper parameter LSTM model testing results

Below is the graph obtained while training the LSTM model on the historical dataset for the best hyper parameter set of 25. The Test loss for the Mean Squared Error Metric is 26.1711 as mentioned in the Graph below and in the table: 4.5

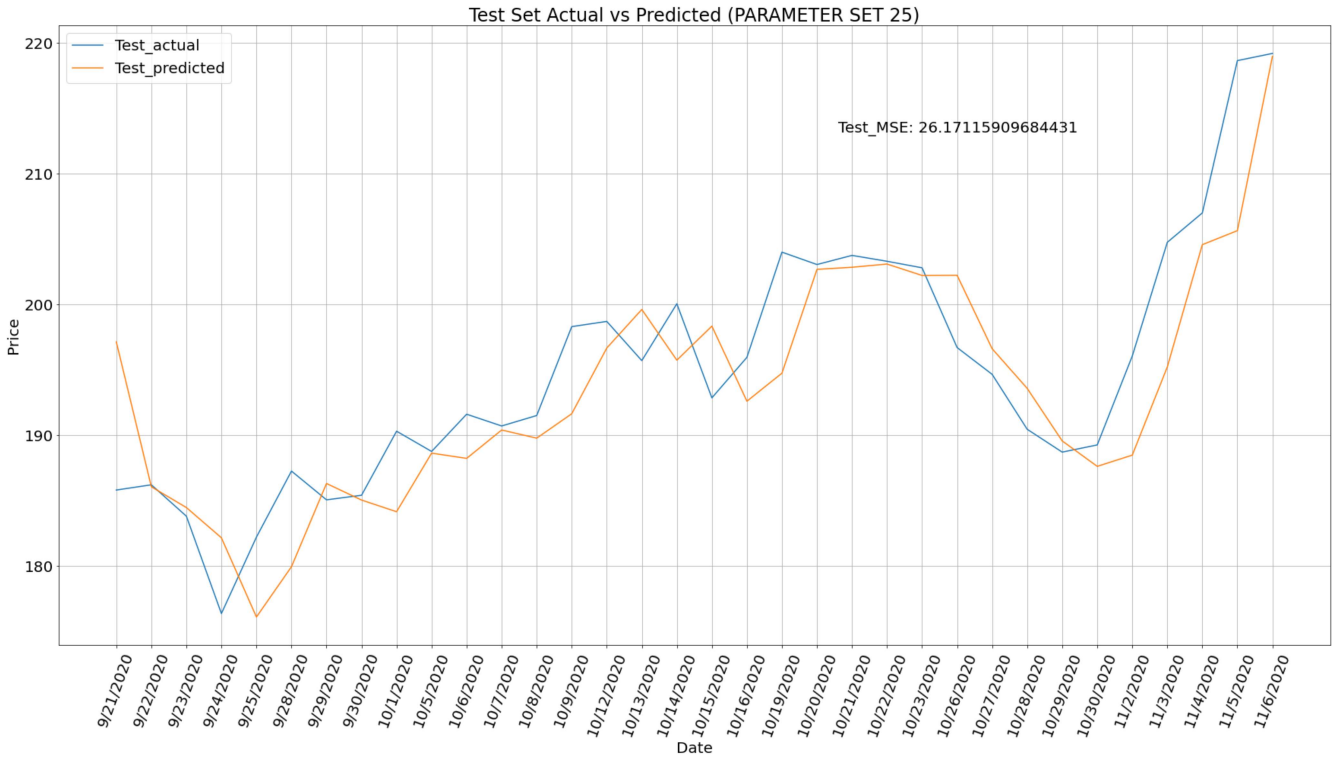


Figure 4.2 Testing data predictions with LSTM model compared with ground truth

4.2.4.3 ANN Modelling for consuming Technical Indicators

As we have mentioned in the plan of action - first module, the main reason why we are involving an ANN is to input the technical indicators.

ANN model takes the output of the LSTM model and the technical indicator (SMA) of closing price as input to predict the next days close price more accurately than the LSTM model.

In order to prepare the training dataset for ANN we have made the LSTM model to predict the next days close price using open, low, close and high prices from SBI-NSE training set and the output of it and the SMA calculated for the closing prices are the independent variables and the next day's closing price is the dependent data for training the ANN model.

Below is the Exact dataset that we have used in training the ANN model, which is arranged in time-series

Step1: predicted the training and testing data for next days close price using LSTM and logged the results in column 5, we could see that the first 30 rows has been dropped after pre-processing because we have dropped the top rows with null columns after grouping for 30 time-steps.

File Name: Training_set_LSTM_predictions.csv

Cloud Path:

https://drive.google.com/file/d/1Z2c5bbaPt8ucIQiFoA4k4LWmiRL2AIPZ/view?usp=sharing

Step2: calculated the SMA (with 5 time-steps) for Actual close price and logged the results

File Name: Training_set_LSTM_predictions_SMA_calculation.csv

Cloud Path:

https://drive.google.com/file/d/1-5VdVhIaS-7ZJtfeIpQOtfsz5ax4-wBc/view?usp=sharing

Steps3: dropped the row with null after time-stamp grouping (pre-processed), Final dataset for training ANN is below.

File Name: Training_set_LSTM_predictions_SMA_calculation_preprocessed.csv

Cloud Path:

https://drive.google.com/file/d/1-HDWkh4UfYm3DdaQL8jgh4yxQM7AbH3a/view?usp=sharing

Similarly, for testing, we have taken the last 34 rows, from 21-9-2020 to 06-11-2020, from the complete dataset and kept it as testing data in the same way we did it for LSTM model evaluation, and prepared the testing data in the same way we prepared it for training data. (First use the open, close, low, high prices to predict the next days close price and log it in under LSTM_predictions, secondly use the close price to calculate SMA for the 5 time-steps).

Below is the Exact dataset that we have used in testing the ANN model, which is arranged in time-series. Note: we have added first 4 rows manually from training data for Actual closing price, so that the we will not have any null values in creating the SMA column starting from the testing date of 21/9/2020 as we have time-steps of five.

Step1: preparing Prediction dataset with LSTM predictions

File Name: Testing_set_LSTM_predictions.csv

Cloud Path:

https://drive.google.com/file/d/1-M-IAZgrABh_3UEJ4da1C_RGPYx9dzwB/view?usp=sharing

Step2: Calculating SMA for the Actual Closing Price

File Name: Testing_set_LSTM_predictions_SMA_calculation.csv

Cloud Path:

https://drive.google.com/file/d/1-RdIe27HiEQ4qJb5HOHS6m-cHvhJhYws/view?usp=sharing

Step3: Preprocessing the above by removing null values after grouping into time-steps for technical indicator

File Name: Testing_set_LSTM_predictions_SMA_calculation_preprocessed.csv

Cloud Path:

https://drive.google.com/file/d/1-SgaxnlCw7yZy1-xS1qbqtFLNkVTV4HV/view?usp=sharing

4.2.4.4 ANN model Evaluation for varied hyper parameters

We have split the training set into 80% training data and 20% validation data and Logged the below validation losses for the range of hyper-parameters tuned.

Hidden Layers	Units per layer	Batch size	Val loss
6	2	64	16.876
6	2	32	16.476
6	2	16	17.127
6	2	8	17.547
5	2	32	16.883
10	2	32	16.369
10	3	32	15.103
10	4	32	16.973

Table 4.8 List of hyper-parameters and respective test losses for ANN

The hyper parameters that we have tuned are number of hidden layers in an ANN, number of units per hidden layer, Batch size. All other hyper-parameters are fixed throughout the training. We have used Early stopping criteria with 50 patience while training the above models. The best hyper parameter which gave the minimum validation loss is highlighted in green.

Highlighted green row is the best hyper-parameters for which we achieved the minimum validation loss and below is the model and the loss logged while training.

The weights saved for the best hyper-parameter training is below:

File Name: ANN_best_weights.h5

Cloud Path:
https://drive.google.com/file/d/1UBU_N_AiSx1JeW_4cEmZZHJBEBuS1Nvf/view?usp=sharing

Below is the log file that captured the validation loss for each epoch while training the ANN model with the best hyper-parameters. As mentioned in the log the training ended at 233rd epoch as per the early stopping criteria because for the next 50 epochs from 183rd epoch there is no loss reduction from 17.8047.

File Name: log.csv

Cloud Path:
https://drive.google.com/file/d/1GFDwL5C30cns0ex3Y7eQX0ItmfipLbQ7/view?usp=sharing

4.2.4.4.1 ANN Model Training results for the best hyper parameters

Below is the graph plotted for training and validation loss obtained while training with the best hyper-parameters.

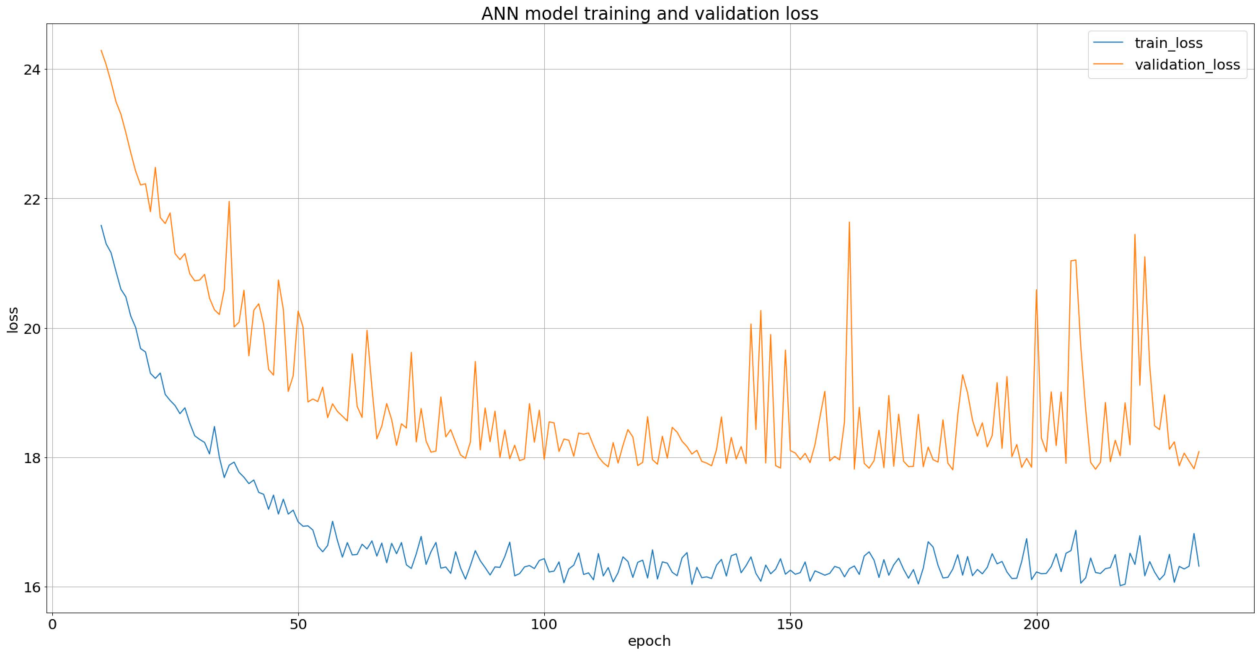


Figure 4.3 ANN Training and validation loss convergence with respect to epochs for the best hyper-parameters mentioned in table 4.8

4.2.4.4.2 Best hyper parameter ANN Model Testing results

Below is the graph plotting the predictions for the testing dataset mentioned above.

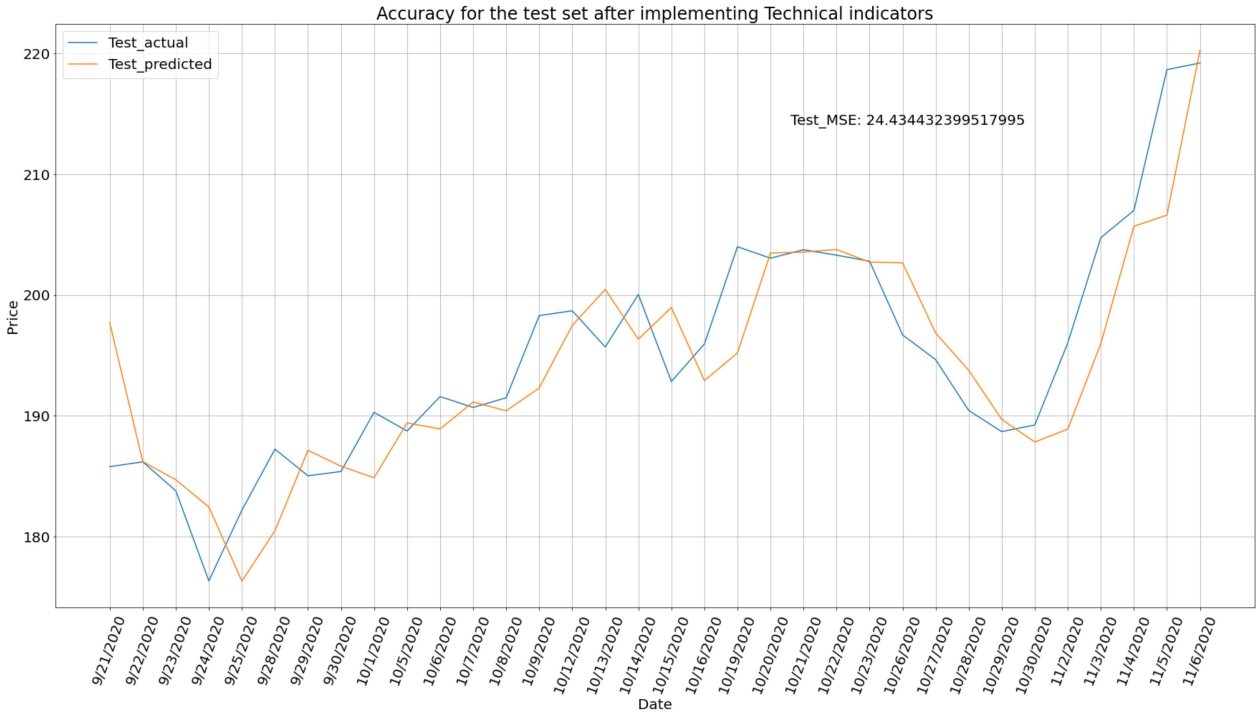


Figure 4.4 Testing data predictions with ANN model compared with ground truth

Below are the results tabulated for the above graph figure 4.4, representing the predicted next days' close price.

Date	Predicted Close Price	Date	Predicted Close Price
9/21/2020	197.4439	10/15/2020	198.84413
9/22/2020	189.41508	10/16/2020	196.0476
9/23/2020	184.83174	10/19/2020	195.27335
9/24/2020	183.59212	10/20/2020	202.04526
9/25/2020	178.1102	10/21/2020	204.71701
9/28/2020	179.31923	10/22/2020	204.73505
9/29/2020	185.6796	10/23/2020	203.96976
9/30/2020	186.47104	10/26/2020	203.74403
10/1/2020	185.11932	10/27/2020	199.14493
10/5/2020	188.60406	10/28/2020	194.71582
10/6/2020	189.72284	10/29/2020	190.97389
10/7/2020	190.71046	10/30/2020	188.2137
10/8/2020	190.9859	11/2/2020	188.61711
10/9/2020	192.45793	11/3/2020	194.05838
10/12/2020	196.59882	11/4/2020	203.08037
10/13/2020	201.03629	11/5/2020	206.05124
10/14/2020	198.44289	11/6/2020	217.45999

Table 4.10 ANN model prediction results for the next days close price on the testing data

4.2.4.5 Achievement in implementing Technical indicators

The main reason to construct the ANN model is to use the power of technical indicators and by doing so we could see a considerable decrease in the test loss (MSE) by 1.73 in predicting the next days close price as mentioned in the table below.

Testing data (34 days)	Models	MSE Test Loss
9/21/2020 – 11/6/2020	LSTM	26.1711
9/21/2020 – 11/6/2020	ANN	24.4344

Table 4.9 Loss decreased in implementing ANN with technical indicators

4.2.4.6 FIN-BERT Modelling

We are not going to train a model for sentiment prediction instead we are going to use the pre-trained weights of the FIN-BERT model mentioned in the paper [31]

Within the pre-trained FIN-BERT weights, we have below four categories:

'finBert-baseVocab-uncased'

'finBert-baseVocab-cased'

'finBert-finVocab-cased'

'finBert-finVocab-uncased'

We have also provided an option to test the base-BERT models (provided by google) too in the submitted Jupiter notebook:

'baseBert-baseVocab-cased'

'baseBert-baseVocab-uncased'

4.2.4.7 FIN-BERT Evaluation report

Below is the evaluation result from the paper [31] on different varieties of FIN-BERT along with the base-BERT. Apart from these four pre-trained (FIN-BERT) models we have chosen to use the 'finBert-finVocab-uncased' model which is proven to dominate all 3 of the other FIN-BERT models as mentioned in the paper [31], below table for reference.

Datasets	BERT		FinBERT-BaseVocab		FinBERT-FinVocab	
	cased	uncased	cased	uncased	cased	uncased
PhraseBank	0.755	0.835	0.856	0.870	0.864	**0.872**
FiQA	0.653	0.730	0.767	0.796	0.814	**0.844**
AnalystTone	0.840	0.850	0.872	0.880	0.876	**0.887**

Table 4.11 comparison of accuracies for different FIN-BERT and BASE-BERT versions

4.2.4.8 Fine-Tuning 'finBert-baseVocab-uncased' model

In order to make the pre-trained 'finBert-finVocab-uncased' model fit our task of sentiment classification (positive, neutral and negative) we used the 'finetuned-finBert-finVocab-uncased' model that has been trained on top of pre-trained 'finBert-finVocab-uncased' model using 10,000 sentences as mentioned in the paper [31], which resembles the same that we are going to predict in our case.

4.2.4.9 Manual evaluation of 'finetuned-finBert-finVocab-uncased' model

We have proof as shown in the table 4.11 that the 'finBert-finVocab-uncased' is the state of the art model to predict sentiments for financial data, but still we were eager to manually check the final 'finetuned- finBert-finVocab-uncased' models sentiment prediction ability for finance related statements and the results are as mentioned below:

Manually composed sentences	Predicted Sentiment
The company is in good shape	Positive
there is a shortage of capital, and we need extra financing	Negative
the companies stock value is growing positively	Positive
growth is strong and we have plenty of liquidity	Positive
the company is in bad shape	Negative
there are doubts about our finances	Negative
there are doubts about our finances	Neutral
profits are flat	Neutral

Table 4.12 Manual evaluation of finetuned-finBert-finVocab-uncased model

We could see from the above table that the sentiments predicted by the 'finetuned-finBert-finVoca-uncased' model for the manually written sentences are accurate. So, we now confidently can use this model in our case to predict the sentiments for the everyday news articles gathered manually as mentioned in session 4.2.1.2

4.2.4.10 finetuned-finBert-finVocab-uncased model sentiment prediction results

Date	Predicted Sentiment	Date	Predicted Sentiment
9/21/2020	Positive	10/14/2020	Positive
9/22/2020	Positive	10/16/2020	Positive
9/23/2020	Positive	10/20/2020	Positive
9/25/2020	Positive	10/22/2020	Positive
9/28/2020	Positive	10/23/2020	Positive
9/29/2020	Positive	10/26/2020	Positive
10/5/2020	Positive	10/27/2020	Positive
10/6/2020	Positive	10/28/2020	Positive
10/7/2020	Positive	10/29/2020	Positive
10/8/2020	Positive	10/30/2020	Positive
10/9/2020	Positive	11/3/2020	Positive
10/12/2020	Positive	11/4/2020	Positive
10/13/2020	Positive	11/6/2020	Neutral

Table 4.13 finetuned-finBert-finVocab-uncased model sentiment prediction results

The above table shows the predicted sentiments for the test dataset collected manually as mentioned in session 4.2.1.2. For this limited test dataset collected manually we predicted sentiments for positive and neutral, but not the negative sentiment. This shows that the company's performance (State Bank of India) was great from the late September to early November.

Per day's sentiments are calculated for based on the sentiment fuzzy logic mentioned in chapter 3.3 the sentiments are first predicted for each paragraph of an article, and the articles sentiment is the cumulative result of all the sentiments of paragraphs present in that particular article. Similarly, the sentiment for a particular day is the cumulative result of sentiments calculated for each article for that particular day. The cumulative result is the maximum repeating sentiment.

For example, let assume that for a particular day the number of articles collected for the stock is 5 and the number of paragraphs for each article is 5 and the sentiments predicted for each of the paragraphs is as mentioned below in the below table.

Articles	Paragraphs	predicted sentiments	cumulative article sentiments	cumulative per-days sentiment
Atricle1	paragraph 1	Positive	positive	neutral
	paragraph 2	negative		
	paragraph 3	positive		
	paragraph 4	neutral		
	paragraph 5	Positive		
Atricle2	paragraph 1	neutral	neutral	
	paragraph 2	negative		
	paragraph 3	positive		
	paragraph 4	neutral		
	paragraph 5	negative		
Atricle3	paragraph 1	negative	negative	
	paragraph 2	positive		
	paragraph 3	negative		
	paragraph 4	neutral		
	paragraph 5	negative		
Atricle4	paragraph 1	neutral	neutral	
	paragraph 2	positive		
	paragraph 3	neutral		
	paragraph 4	positive		
	paragraph 5	neutral		
Atricle5	paragraph 1	positive	neutral	
	paragraph 2	positive		
	paragraph 3	neutral		
	paragraph 4	negative		
	paragraph 5	negative		
	paragraph 6	neutral		

Table 4.14 Example to showcase how sentiment per days is calculated in table 4.13

The table mentioned above gives an overall view on how the sentiments are calculated for a day, as explained in detail on the below steps:

Step1: Predicting the sentiment for each paragraph in an article using the finetune-pretrained-FinBert model, column in three in the above tables shows the same.

Step2: Finding the maximum occurrence (cumulative result) of a sentiment predicted for each para in an article, and that is the sentiment for that particular article. Rows 1, 3, and 4 in column three in the above table represents the maximum occurrence of sentiments for each para in an article, but the rows two and five are bit tricky as they do not have a single maximum occurring

sentiment. In placed where we could not identify a single maximum occurrence of sentiments will be assigned neutral.

Step3: Finding the sentiment for a day. Similar to step2 the sentiment for a day is the cumulative result of each sentiment calculated for each article of that particular day.

CHAPTER 5

RESULTS AND DISCUSSIONS

In this chapter we are aimed at discussing the consolidated results from the previous chapters.

5.1 Combined results from the predicted next days close price and the current day's sentiment for the stock

Date	Predicted Sentiment	Actual Close Price	Predicted close price	Actual open price	Predicted Difference (Actual Open - Predicted Close)	Actual Difference (Actual Open - Actual Close)	Predicted Trend	Actual Trend
9/21/2020	Positive	185.8	197.698	193.05	4.6476 (positive)	-7.25 (negative)	buy	sell
9/22/2020	Positive	186.2	186.218	186.15	0.0682 (positive)	0.0500 (positive)	buy	buy
9/23/2020	Positive	183.8	184.718	188.15	-3.432 (negative)	-4.349 (negative)	Other	sell
9/25/2020	Positive	182.2	176.332	179	-2.668 (negative)	3.1999 (positive)	Other	buy
9/28/2020	Positive	187.25	180.493	184	-3.507 (negative)	3.25 (positive)	Other	buy
9/29/2020	Positive	185.05	187.156	188.5	-1.344 (negative)	-3.45 (negative)	Other	sell
10/5/2020	Positive	188.75	189.422	192	-2.577 (negative)	-3.25 (negative)	Other	sell
10/6/2020	Positive	191.6	188.921	191.1	-2.179 (negative)	0.5 (positive)	Other	buy
10/7/2020	Positive	190.7	191.148	192.55	-1.401 (negative)	-1.850 (negative)	Other	sell
10/8/2020	Positive	191.5	190.424	191.35	-0.925 (negative)	0.1499 (positive)	Other	buy
10/9/2020	Positive	198.3	192.311	192	0.3113 (positive)	6.3000 (positive)	buy	buy
10/12/2020	Positive	198.7	197.49	199.7	-2.209 (negative)	-1 (negative)	Other	sell
10/13/2020	Positive	195.7	200.467	198.65	1.8173 (positive)	-2.95 (negative)	buy	sell
10/14/2020	Positive	200.05	196.35	194.05	2.2997 (positive)	6 (positive)	buy	buy
10/16/2020	Positive	195.95	192.926	194	-1.073 (negative)	1.9499 (positive)	Other	buy
10/20/2020	Positive	203.05	203.475	201	2.4748 (positive)	2.0500 (positive)	buy	buy
10/22/2020	Positive	203.3	203.768	201.9	1.8684 (positive)	1.4000 (positive)	buy	buy
10/23/2020	Positive	202.8	202.727	204	-1.273 (negative)	-1.2 (negative)	Other	sell
10/26/2020	Positive	196.7	202.661	202.7	-0.039 (negative)	-6 (negative)	Other	sell
10/27/2020	Positive	194.65	196.851	197.25	-0.398 (negative)	-2.600 (negative)	Other	sell
10/28/2020	Positive	190.45	193.772	195	-1.227 (negative)	-4.55 (negative)	Other	sell
10/29/2020	Positive	188.7	189.71	189.35	0.3601 (positive)	-0.650 (negative)	buy	sell
10/30/2020	Positive	189.25	187.844	189.35	-1.506 (negative)	-0.100 (negative)	Other	sell
11/3/2020	Positive	204.75	195.992	198	-2.008 (negative)	6.75 (positive)	Other	buy
11/4/2020	Positive	207	205.689	203.5	2.1887 (positive)	3.5 (positive)	buy	buy
11/6/2020	Neutral	219.2	220.242	219	1.2423 (positive)	0.1999 (positive)	buy	buy

Table 5.1 Consolidated results with Overall market trend predicted for the next day

The above table is constructed as follows:

- Columns 2 and 4 is the predicted current day's (before the market opens for the next day) sentiment and the predicted next day's close price collected from the tables 4.13 and 4.10 respectively.
- Columns 3 and 5 is collected directly from the historical data for that particular day.
- Column 6 is the difference in between the 'Actual open price' (column 5) and the 'predicted close price' (column 4)
- Similarly, column 7 is the difference between the 'Actual open price' (column 5) and the 'actual close price' (column 3)
- Column 8 is based on the overall fuzzy logic model explained in chapter 3.4 applied between the columns 'predicted sentiment' (column 2) and the 'predicted difference' (column 6). If column 2 and 6 are not matching, then we are going to mark column 8 as 'other'. If it is matching with positive, we are going to mark as buy. And If it is matching with negative, we are going to mark as sell.
- Column 9 is just the reflection of column 7, where if the difference is positive it is buy trend, and if the difference is negative it is sell trend.

5.2 Discussion on the results

Predicted trend (column 8) in table 5.1 is the final result obtained from our overall research work. We are not going to trade for days whose predicted trend is 'Other' and the remaining days are trading days as estimated by the model.

Total trading days from the above table 5.1 are ten (highlighted in red and green). If both the 'predicted trend' (column 8) and the 'actual trend' (column 9) are the same, then we have traded profitably for that particular day (highlighted as green) if not we have traded for loss (highlighted as red).

Total days traded = 10 days

Days traded in profits = 7 days

Days traded in loss = 3 days

Accuracy of our model in trading profitable for the trading days is as below:

= [(days traded in profits)/(Total days traded)]*100

= (7/10)*100

= 70% accuracy

Accuracy of our model in trading profitable for the days estimated as trading days = 70%

CHAPTER 6

CONCLUSIONS AND FUTURE PLANS

6.1 Conclusion

In this thesis, we have used ANN and LSTM models for predicting the next days close price, and the pre-trained and fine-tuned Fin-Bert model for predicting the current day's (before the next day's market open) sentiment for the news articles collected for the stock using google-alerts. By using both the predicted current day's (before the net days market opening) sentiments and the next day's close price we have estimated the market trend for the stock for the next day. The final accuracy of our model in trading profitable for the days estimated as trading days is 70%.

6.2 Our future plans on improving our work further

- To verify our work with different stocks on different stock markets.
- Using Social media posts and user comments for sentiment analysis.
- Trying to predict the trend and trade between the other prices too, and not restricting to Open-Close prices only.
- Building an ensemble model trained on difference stock data to capture novel variations.

REFERENCES:

1. Moghar, A. and Hamiche, M., (2020) Stock Market Prediction Using LSTM Recurrent Neural Network. In: *Procedia Computer Science*. Elsevier B.V., pp.1168–1173.
2. Siami-Namini, S., Tavakoli, N. and Siami Namin, A., (2019) A Comparison of ARIMA and LSTM in Forecasting Time Series. In: *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*. Institute of Electrical and Electronics Engineers Inc., pp.1394–1401.
3. Ghosh, P., Neufeld, A. and Sahoo, J.K., (2020) Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. [online] Available at: http://arxiv.org/abs/2004.10178 [Accessed 6 Jul. 2020].
4. Hochreiter, S. and Schmidhuber, J., (1997) Long Short-Term Memory. *Neural Computation*, 98, pp.1735–1780.
5. Anon (2020) *Understanding LSTM Networks -- colah's blog*. [online] Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ [Accessed 6 Jul. 2020].
6. Shi Yan (2020) *Understanding LSTM and its diagrams - ML Review - Medium*. [online] Available at: https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714 [Accessed 6 Jul. 2020].
7. Sherstinsky, A., (2018) Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, [online] 404. Available at: http://arxiv.org/abs/1808.03314 [Accessed 6 Jul. 2020].
8. Laptev, N., Yosinski, J., Erran Li, L. and Smyl, S., (n.d.) *Time-series Extreme Event Forecasting with Neural Networks at Uber*.
9. Bailer-Jones, C., MacKay, D. and Withers, P., 1998. A recurrent neural network for modelling dynamical systems. *Network: Computation in Neural Systems*, 9(4), pp.531-547.
10. Hewamalage, H., Bergmeir, C. and Bandara, K., (2019) Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions. [online] Available at: http://arxiv.org/abs/1909.00590 [Accessed 6 Jul. 2020].
11. Lipton, Z.C., Berkowitz, J. and Elkan, C., (2015) A Critical Review of Recurrent Neural Networks for Sequence Learning. [online] Available at: http://arxiv.org/abs/1506.00019 [Accessed 6 Jul. 2020].
12. Petneházi, G., (2018) Recurrent Neural Networks for Time Series Forecasting. [online] Available at: http://arxiv.org/abs/1901.00069 [Accessed 6 Jul. 2020].
13. Pascanu, R., Mikolov, T. and Bengio, Y., (2013) *On the difficulty of training recurrent neural networks*. [online] Available at: http://proceedings.mlr.press/v28/pascanu13.html [Accessed 6 Jul. 2020].
14. Gers, F.A., Schmidhuber, J. and Cummins, F., (1999) Learning to forget: Continual prediction with LSTM. In: *IEEE Conference Publication*. IEEE, pp.850–855.
15. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *EMNLP 2014 Conference on Empirical*

Methods in Natural Language Processing, Proceedings of the Conference. [online] Association for Computational Linguistics (ACL), pp.1724–1734. Available at: https://arxiv.org/abs/1406.1078v3 [Accessed 6 Jul. 2020].
16. Weiss, G., Goldberg, Y. and Yahav, E., (2018) On the Practical Computational Power of Finite Precision RNNs for Language Recognition. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, [online] 2, pp.740–745. Available at: http://arxiv.org/abs/1805.04908 [Accessed 6 Jul. 2020].
17. Britz, D., Goldie, A., Luong, M.T. and Le, Q. V., (2017) Massive exploration of neural machine translation architectures. In: *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. [online] Association for Computational Linguistics (ACL), pp.1442–1451. Available at: https://github.com/moses- [Accessed 6 Jul. 2020].
18. Pathak, A. and Shetty, N.P., (2019) Indian Stock Market Prediction Using Machine Learning and Sentiment Analysis. In: *Advances in Intelligent Systems and Computing*. Springer Verlag, pp.595–603.
19. Li, H., Shen, Y. and Zhu, Y., (2018) *Stock Price Prediction Using Attention-based Multi-Input LSTM*. [online] *Proceedings of Machine Learning Research*, Available at: http://proceedings.mlr.press/v95/li18c.html [Accessed 6 Jul. 2020].
20. Roondiwala, M., Patel, H. and Varma, S., (2015) *Predicting Stock Prices Using LSTM*. [online] *International Journal of Science and Research*, Available at: https://pdfs.semanticscholar.org/3f5a/cb5ce4ad79f08024979149767da6d35992ba.pdf [Accessed 6 Jul. 2020].
21. Nelson, D.M.Q., Pereira, A.C.M. and De Oliveira, R.A., (2017) Stock market's price movement prediction with LSTM neural networks. In: *Proceedings of the International Joint Conference on Neural Networks*. Institute of Electrical and Electronics Engineers Inc., pp.1419–1426.
22. Chen, K., Zhou, Y. and Dai, F., (2015) A LSTM-based method for stock returns prediction: A case study of China stock market. In: *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*. Institute of Electrical and Electronics Engineers Inc., pp.2823–2824.
23. Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P., (2017) Stock price prediction using LSTM, RNN and CNN-sliding window model. In: *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*. Institute of Electrical and Electronics Engineers Inc., pp.1643–1647.
24. Jia, H., (2016) Investigation Into The Effectiveness Of Long Short Term Memory Networks For Stock Price Prediction. [online] Available at: http://arxiv.org/abs/1603.07893 [Accessed 6 Jul. 2020].
25. Hoang, M., Alija Bihorac, O. and Rouces, J., (2020) *Article | Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa), September 30 - October 2, Turku, Finland | Aspect-Based Sentiment Analysis using BERT*. [online]

26. Available at: https://ep.liu.se/ecp/article.asp?issue=167&article=020&volume= [Accessed 6 Jul. 2020].
27. Li, X., Bing, L., Zhang, W. and Lam, W., (2019) Exploiting BERT for End-to-End Aspect-based Sentiment Analysis. [online] pp.34–41. Available at: http://arxiv.org/abs/1910.00883 [Accessed 6 Jul. 2020].
28. Hiew, J.Z.G., Huang, X., Mou, H., Li, D., Wu, Q. and Xu, Y., (2019) BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability. [online] Available at: http://arxiv.org/abs/1906.09024 [Accessed 6 Jul. 2020].
29. Yang, L., Dong, R., Ng, T.L.J. and Xu, Y., (2019) *Leveraging BERT to Improve the FEARS Index for Stock Forecasting*. [online] Available at: https://www.aclweb.org/anthology/W19-5509/ [Accessed 6 Jul. 2020].
30. Anon (2020) *Technical Indicator Definition*. [online] Available at: https://www.investopedia.com/terms/t/technicalindicator.asp [Accessed 6 Jul. 2020].
31. Yang, Y., UY, M.C.S. and Huang, A., (2020) FinBERT: A Pretrained Language Model for Financial Communications. [online] Available at: http://arxiv.org/abs/2006.08097.

APPENDIX: RESEARCH PROPOSAL

INTRADAY STOCK TREND PRICE PREDICTION USING LSTM AND BERT

ANAND NATARAJ MANOHARAN

Liverpool john Moores University, UK

MSc Data Science (DS) Program

A.N.Manoharan@2020.ljmu.ac.uk

ABSTRACT:

In this research we are aimed at predicting the intraday stock trend (positive/negative) especially the trend between OPEN and CLOSE prices, and how to trade the stock profitably once the trend is found. The reason for choosing the open-close price is explained in the objectives.

In this paper we have discussed the utilization of a sequence-based deep learning (LSTM) model for understanding the hidden patterns in the historical time-series stock data, the output of this model along with the technical indicators are the input features to an ANN model to predict the next day's CLOSE price of the stock.

The state-of-the-art NLTK-BERT model is used for knowing the current day's sentiment of the stock based on the news and articles published in the news channels and blogs.

We finally combine the next day's predicted close price and the current day's stock sentiment to decide on the next day's stock trend and based on it, if it is positive or negative trend, we buy or sell the stock respectively on the next day, when the market opens.

Please note: We have considered SBI stock from NSE market for our research.

Research Questions:

1. Is it possible to track a particular stock's (SBI-NSE in our case) trend (positive/negative) for the next day in intraday market, especially between the open and close prices, by analyzing the historical price data and understanding the today's sentiment of the stock?

2. Is it possible to trade profitably the stock by understanding the open-close price trend for the next day?

INTRODUCTION:

The stock market is the place where the listed companies' shares are sold or bought to make profits. The name stock indicates the shares of different companies or the same company. If a share has been bought for a very less price and sold at a very high price indicates a profit, similarly, if a stock has been sold at a very high price and bough at a very less price indicates profit too. The reverse of any of these two would result in loss.

Note: One interesting fact in the stock market is that it is not necessary to buy a stock to sell, we can sell even before buying a stock.

Stock sentiments could be **Positive, Negative or Neutral** based on the information collected about the stock from news channels and web-blogs is good, bad or neutral.

Similarly, **stock trend** could be **Positive, Negative or Neutral** for a given time-period.

We can say a particular time-period is a positive or negative trend by understanding the starting and the ending prices of a time-period.

If the difference of starting and ending prices in a time-period is positive, then it is a positive trend and if it is negative then it is a negative trend.

For example, in the graph shown above the trend at the period 1 (10:15 AM to 10:40 PM) is negative trend period 2 (1:20 PM to 2:30 PM) is positive trend.

Intraday in the stock market is shorthand for the stocks that trade on the markets during regular business hours and their **price** movements.

During intraday, the price of a stock at which the market opens is the **OPEN** price and the price at which the market closes is the **CLOSE** price, and in between, there are **HIGH** and **LOW** prices which is the maximum and the minimum price reach of the stock of that particular day. Note: both high and low prices can sometimes be the open and close prices too.

AIM OF THE PROJECT:

Our work is to predict the trend between the open and close prices of the SBI-NSE stock for intraday using LSTM and BERT model, and by using the trend predicted to trade the stock profitably for the day.

OBJECTIVES:

We could categories the overall objective into **three steps:**

First Objective:

We are going to build a robust **LSTM** model which could predict the pattern for the CLOSE price historical time-series stock data.

The reason for choosing Close price prediction as mentioned above is as follows:

There are 6 periods in intraday where we could trade,
1. Between Open and High (positive trend)
2. Between Open and Low (negative trend)
3. Between Open and Close (can be a positive or negative trend depending on which price is greater)
4. Between Low/High and High/Low (can be a positive or negative trend depending on which price hits first)
5. Between High and Close (negative trend)
6. Between Low and Close (positive trend)

For us to trade between (4, 5, 6) regions we have to accurately predict the price of both the starting and the ending points, but in the first three regions, it is enough to predict the price of the ending points because the starting point is the open price which is known at the time of market opening (we can trade only after the market has been opened, so predicting the open price before the market has opened has no significance).

By trading between the first three regions (open-high, open-low, open-close) the risk of the wrong prediction has just reduced to half since we are not going to predict the starting point (Open price). And among the first three regions, open-close is the most optimal period to trade in because we do not know the time at which the high or low price hits, and it would be tedious to predict the time and the high/low price together, whereas, in the close price we could say that it hits every day at 3 PM in the intraday market (the time when the NSE market closes) therefore, it is easier to automate our algorithm to buy or sell stock at the closing time (which is known) instead the time (which unknown) at which the price hits the high or low price. This is the reason why we have chosen Close price prediction in the first place.

Data set for training LSTM: historical data of the stock from Yahoo finance website.

The input to LSTM for prediction could either be the past days close price or the combination of any prices, for example, to predict the next day's CLOSE price in LSTM we could either input the historical prices of the previous day's (CLOSE price) or the previous days (CLOSE & LOW) or (HIGH & LOW) or (HIGH, CLOSE & LOW) or in any combination of all four prices, but based on the test results we could see that the combination of all four prices (HIGH, LOW, OPEN & CLOSE) as the input feature had produced better results.

The output of the LSTM model along with the technical indicators is the input features for the ANN model which would output the nearest next day's stock CLOSE price.

Technical indicators for the stock can be calculated based on the historical data of the stock and by using the formulas of the technical indicators listed in this blog [30]

Second Objective:

The present day's stock sentiment is analyzed by using the NLTK-BERT model. The inputs for this model are the news and blog articles captured today for the stock via google alerts. The output of the NLTK-BERT model is the input for the sentiment fuzzy logic module to understand the stock sentiment.

We are not going to train the NLTK model since we are going to use the pre-trained Bert model that can predict the sentiment of web-blog and news articles.

For Sentiment Analysis of stock on daily basis**,** we are planning to use 2 main sources and they are: News channels and web-Blogs

Since all news channels now have websites, and the web-blogs are websites itself we do not need any other sources other than Google alerts to capture the information about the stocks.

Once the data has been captured on daily basis the data has been processed and sent to the NLTK-BERT model for sentiment analysis, here we are going to capture three types of sentiments, namely: positive, negative and neutral.

The sentiments are analyzed for each article captured from blogs and news sites. Based on the number of sentiments and their percentage the overall stock sentiment is judged for the day in the sentiment fuzzy logic module. For example, if the number of articles captured by google alerts for the stock is 5 and the sentiments analyzed for each article is: positive, positive, positive, negative, and neutral (positive = 60%, negative = 20%, neutral = 20%) then it would be categorized as positive sentiment overall. A complete sentiment fuzzy logic rules on deciding present day’s stock sentiment is on below:

Sentiment Fuzzy module Logic:

Rule 1:

If (% of Positive sentiments collected) > (% of Negative and Neutral Sentiments collected):
Then the present days stock sentiment = POSITIVE

Rule 2:

If (% of Negative sentiments collected) > (% of Positive and Neutral Sentiments collected):
Then the present days stock sentiment = NEGATIVE

Rule 3:

If (% of Neutral sentiments collected) > (% of Positive and Negative Sentiments collected):
Then the present days stock sentiment = NEUTRAL

Rule 4:

If (% of Neutral sentiments collected == % of Positive sentiments == Negative Sentiments collected):
Then the present days stock sentiment = NEUTRAL

Rule 5:

If (% of Negative sentiments collected == % of Positive sentiments not= Negative Sentiments collected):
Then the present days stock sentiment = NEUTRAL

Rule 6:

If (0 % of Positive sentiments collected and (% of Negative sentiments == Neutral Sentiments collected)):
Then the present days stock sentiment = NEGATIVE

Rule 7:

If (0 % of Negative sentiments collected and (% of positive sentiments == Neutral Sentiments collected)):
Then the present days stock sentiment = POSITIVE

Rule 8:

If (0 % of Neutral sentiments collected and (% of positive sentiments == Negative Sentiments collected)):
Then the present days stock sentiment = NEUTRAL

Rule 9:

If (0 % of Negative sentiments collected == % of Positive sentiments == Negative Sentiments collected):
Then the present days stock sentiment = NEUTRAL

An example scenario describing all the fuzzy rules above:

Positive Sentiment	Negative Sentiment	Neutral Sentiment	Present Day's Stock Sentiment
60%	20%	20%	Positive
20%	60%	20%	Negative
20%	20%	60%	Neutral
33%	33%	33%	Neutral
40%	40%	20%	Neutral
0%	50%	50%	Negative
50%	0%	50%	Positive
50%	50%	0%	Neutral
0%	0%	0%	Neutral (no info on the stock)

Third Objective:

we are going to input the output of the ANN and NLTK-BERT models to an overall fuzzy logic module to judge the next day's overall stock trend and based on this trend the option of buy and sell the stock is decided.

Overall Fuzzy module Logic:

Rule1:

If (next day's (OPEN < CLOSE)) and (Present Day's (Stock Sentiment) = Positive):
Then, next day's predicted stock trend = POSITIVE

Rule2:

If (next day's (OPEN > CLOSE)) and (Present Day's (Stock Sentiment) = Negative):
Then, next day's predicted stock trend = NEGATIVE

Rule3:

If (next day's (OPEN < CLOSE)) and (Present Day's (Stock Sentiment) = Neutral):
Then, next day's predicted stock trend = POSITIVE

Rule4:

If (next day's (OPEN > CLOSE)) and (Present Day's (Stock Sentiment) = Neutral):
Then, next day's predicted stock trend = NEGATIVE

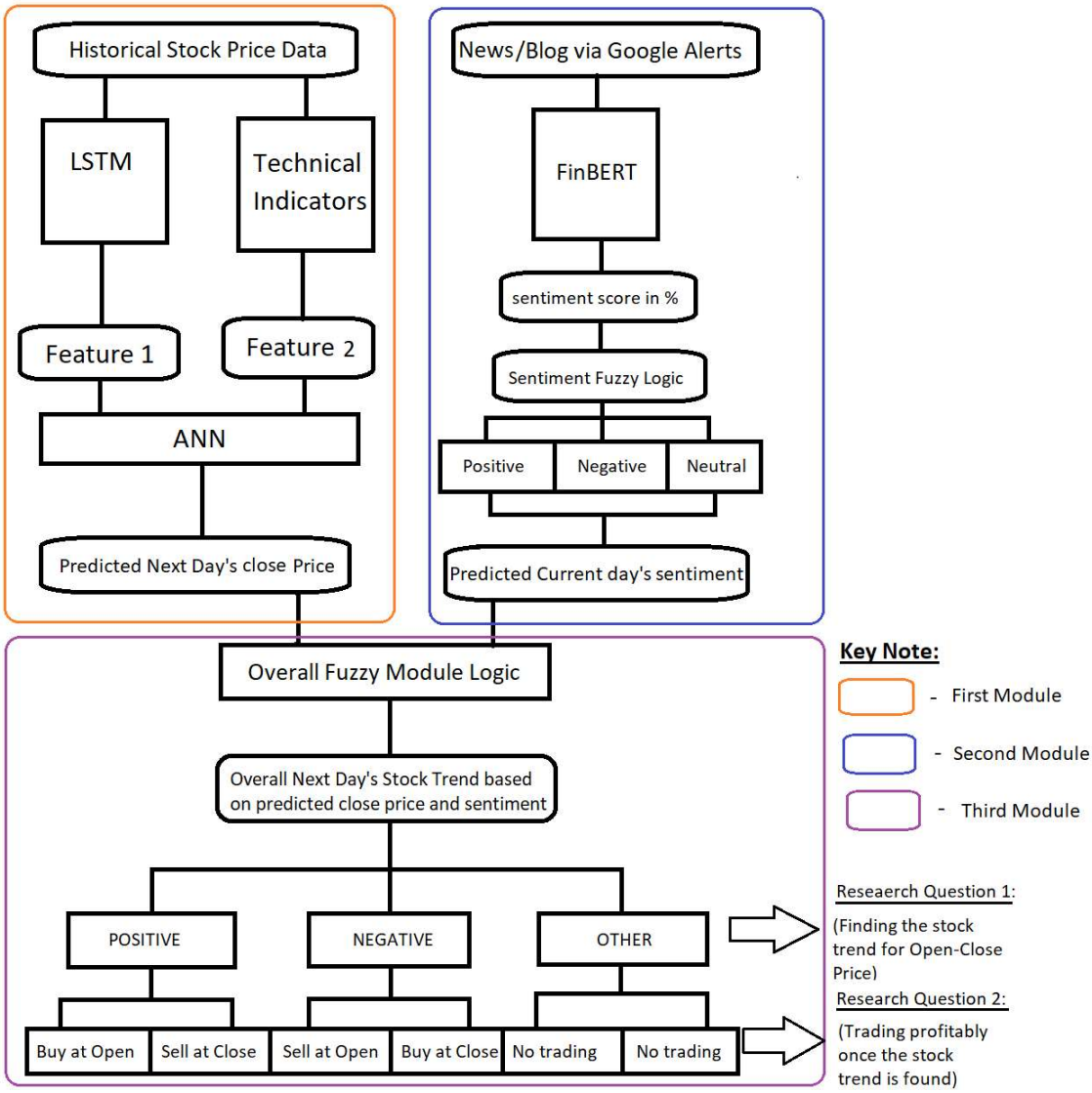
In fuzzy logic module we are not going to consider any other scenarios other than the above four because in any other scenarios the sentiment and open-close trends contradict with each other, for example:

If (next day's (OPEN > CLOSE)) and (Present Day's (Stock Sentiment) = Positive):
Then, next day's predicted stock trend = OTHERS. Here, (OPEN > CLOSE) trend is negative whereas the sentiment trend is positive, so in the other scenarios like this we are not going to trade on that particular day.

Based on the fuzzy module output we are going to place the buy/sell option in two places, one is at the market open and another one at the market close:

1. If it is a **POSITIVE** trend, then buy the stock at market open and sell at market close.
2. If it is a **NEGATIVE** trend, then sell the stock at market open and buy at market close.
3. If it is **OTHER** than the above two trends, then there is no trading action taken for the day.

WORK-BREAKDOWN-STRUCTURE



PROJECT TIMELINE:

OBJECTIVES	Time period
First Objective:	1 month
Literature Review on the Lstm model setup for stock price prediction	1 week
Literature Review on the pretrained bert model for sentiment analysis	1 week
Literature Review on the Fuzzy Rule set-up for sentiment and stock trend	1 week
Literature Review on trading the stock to profitability based on stock trend	1 week
Second Objective:	1 month
Historical Stock Price Collection and Cleaning Data	1 week
LSTM Model Building	1 week
Technical indicators calculation	1 week
ANN model building to predict next days Close price	1 week
Third Objective:	1 month
Setting Google Alert to collect Stock information	1 week
Setting the pretrained NLTK Bert mode	1 week
Calcuating the sentiment percentage based on bert output	1 week
Sentiment fuzzy logic setup to predict current days stock sentiment	1 week
Fourth Objective:	2 weeks
Overall Fuzzy logic set up to predict the Stock trend	1 week
Veryfying whether the trade action taken based on stock trend is correct	1 week

Gann Chart: (Attached with this submission)

Cloud Path:
https://drive.google.com/file/d/1TLuGat73W30c7Jth_RFMS3iozjugdW2L/view?usp=sharing

LITERATURE REVIEW:

As rightly concluded in this paper [8] it would be optimal to choose Neural networks for stock predictions since the quantity and co-relation among the historical stock data is very high. Maybe the length (time steps) might not be very large since we are not exceeding 3 months (90 steps) of a window in our work, but the other two aspects of the number of the data available and the co-relation among the time-series data (open, close, high, low) is very high.

After deciding to work with neural networks we have chosen to work with the Recurrent Neural Networks which is one of the major classes in neural networks and are very powerful for treating the sequence and time-series data as mentioned in these papers [10, 11, 12]

In the early vanilla RNN's proposed in the '90s [9] were bound to two main problems vanishing and exploding gradients [13], which prevented from maintaining the long-term memory which was addressed by the invention of LSTM - These two papers represent the earliest invention of LSTM [4] and its variance [14].

A similar structure to LSTM was proposed lately in 2014 known as GRU [15] to address the same problem of vanishing and diminishing gradients, but these papers prove that LSTM is strictly stronger than GRU as it can easily perform unbounded counting, while the GRU cannot. That's why the GRU fails to learn simple languages that are learnable by the LSTM [16].

Similarly, as shown by Denny Britz, Anna Goldie, Minh-Thang Luong and Quoc Le of Google Brain, LSTM cells consistently outperform GRU cells in "the first large-scale analysis of architecture variations for Neural Machine Translation."[17].

Moreover, this paper [2] demonstrates that the LSTM model outperforming the traditional model, especially in our scenarios which is stock price perdition.

So as per our analysis, we could see that LSTM is one robust RNN variant capable of handling vanishing and diminishing gradient problem and retain the long-term memory along with the short-term memory of the sequence.

This paper [24] helped on how to practically select the features and train an LSTM model for stock prediction and these papers [4,5,6,7] has helped in understanding the nuances of LSTM to improve the accuracy of the model.

Referred the past works on stock prediction using LSTM to understand the actual performance ability of it especially in stock prediction [1, 2, 3, 19, 20, 21, 22, 23]

We have chosen NLTK-Bert for sentiment analysis since as mentioned in this paper [25] Bert could able to outperform most of the state-of-the-art sentiment analysis models, and this paper [26] gave an idea behind understanding Bert for sentiment analysis.

Similar works done in the past are using the BERT sentiments as a feature to the deep learning models to improve the accuracy stock price prediction [27,28, 29].

Apart from all the literature reviews, our entire research methodology is based on the improvement of Nisha Shetty's paper on "Indian Stock Market Prediction Using Machine Learning and Sentiment Analysis" [18] where the author has combined the prediction results of the traditional ML model and the output of the preliminary sentiments analysis model for deciding on buy/sell trend.

The improvement in our model is in using the state-of-the-art deep learning model (LSTM) with technical indicators as the additional features to the ANN. Moreover, the usage of the state-of-the-art NLTK model BERT for sentiment analysis has made our work unique. In addition, we are not dependent on one blog articles, as mentioned in the existing paper above, rather, we are going to collect the information from all the relevant blogs and news channels using google alerts for sentiment analysis.

OUTCOME:

A model that is capable of trading the stock in an intraday market to make profit.

FUTURE WORK:

- To verify our work with different stocks on different stock markets.
- Using Social media posts and user comments for sentiment analysis.
- Trying to predict the trend and trade between the other prices too, and not restricting to Open-Close prices only.

REFERENCES:

1. Moghar, A. and Hamiche, M., (2020) Stock Market Prediction Using LSTM Recurrent Neural Network. In: *Procedia Computer Science*. Elsevier B.V., pp.1168–1173.
2. Siami-Namini, S., Tavakoli, N. and Siami Namin, A., (2019) A Comparison of ARIMA and LSTM in Forecasting Time Series. In: *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*. Institute of Electrical and Electronics Engineers Inc., pp.1394–1401.
3. Ghosh, P., Neufeld, A. and Sahoo, J.K., (2020) Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. [online] Available at: http://arxiv.org/abs/2004.10178 [Accessed 6 Jul. 2020].
4. Hochreiter, S. and Schmidhuber, J., (1997) Long Short-Term Memory. *Neural Computation*, 98, pp.1735–1780.
5. Anon (2020) *Understanding LSTM Networks -- colah's blog*. [online] Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ [Accessed 6 Jul. 2020].
6. Shi Yan (2020) *Understanding LSTM and its diagrams - ML Review - Medium*. [online] Available at: https://medium.com/mlreview/understanding-lstm-and-its-diagrams- 37e2f46f1714 [Accessed 6 Jul. 2020].
7. Sherstinsky, A., (2018) Fundamentals of Recurrent Neural Network (RNN) and Long Short- Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, [online] 404. Available at: http://arxiv.org/abs/1808.03314 [Accessed 6 Jul. 2020].
8. Laptev, N., Yosinski, J., Erran Li, L. and Smyl, S., (n.d.) *Time-series Extreme Event Forecasting with Neural Networks at Uber*.
9. Bailer-Jones, C., MacKay, D. and Withers, P., 1998. A recurrent neural network for modelling dynamical systems. *Network: Computation in Neural Systems*, 9(4), pp.531-547.
10. Hewamalage, H., Bergmeir, C. and Bandara, K., (2019) Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions. [online] Available at: http://arxiv.org/abs/1909.00590 [Accessed 6 Jul. 2020].
11. Lipton, Z.C., Berkowitz, J. and Elkan, C., (2015) A Critical Review of Recurrent Neural Networks for Sequence Learning. [online] Available at: http://arxiv.org/abs/1506.00019 [Accessed 6 Jul. 2020].
12. Petneházi, G., (2018) Recurrent Neural Networks for Time Series Forecasting. [online] Available at: http://arxiv.org/abs/1901.00069 [Accessed 6 Jul. 2020].
13. Pascanu, R., Mikolov, T. and Bengio, Y., (2013) *On the difficulty of training recurrent neural networks*. [online] Available at: http://proceedings.mlr.press/v28/pascanu13.html [Accessed 6 Jul. 2020].
14. Gers, F.A., Schmidhuber, J. and Cummins, F., (1999) Learning to forget: Continual prediction with LSTM. In: *IEEE Conference Publication*. IEEE, pp.850–855.
15. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *EMNLP 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. [online] Association for Computational Linguistics (ACL), pp.1724–1734. Available at: https://arxiv.org/abs/1406.1078v3 [Accessed 6 Jul. 2020].

16. Weiss, G., Goldberg, Y. and Yahav, E., (2018) On the Practical Computational Power of Finite Precision RNNs for Language Recognition. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, [online] 2, pp.740–745. Available at: http://arxiv.org/abs/1805.04908 [Accessed 6 Jul. 2020].
17. Britz, D., Goldie, A., Luong, M.T. and Le, Q. V., (2017) Massive exploration of neural machine translation architectures. In: *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. [online] Association for Computational Linguistics (ACL), pp.1442–1451. Available at: https://github.com/moses- [Accessed 6 Jul. 2020].
18. Pathak, A. and Shetty, N.P., (2019) Indian Stock Market Prediction Using Machine Learning and Sentiment Analysis. In: *Advances in Intelligent Systems and Computing*. Springer Verlag, pp.595–603.
19. Li, H., Shen, Y. and Zhu, Y., (2018) *Stock Price Prediction Using Attention-based Multi-Input LSTM*. [online] *Proceedings of Machine Learning Research*, Available at: http://proceedings.mlr.press/v95/li18c.html [Accessed 6 Jul. 2020].
20. Roondiwala, M., Patel, H. and Varma, S., (2015) *Predicting Stock Prices Using LSTM*. [online] *International Journal of Science and Research*, Available at: https://pdfs.semanticscholar.org/3f5a/cb5ce4ad79f08024979149767da6d35992ba.pdf [Accessed 6 Jul. 2020].
21. Nelson, D.M.Q., Pereira, A.C.M. and De Oliveira, R.A., (2017) Stock market's price movement prediction with LSTM neural networks. In: *Proceedings of the International Joint Conference on Neural Networks*. Institute of Electrical and Electronics Engineers Inc., pp.1419–1426.
22. Chen, K., Zhou, Y. and Dai, F., (2015) A LSTM-based method for stock returns prediction: A case study of China stock market. In: *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*. Institute of Electrical and Electronics Engineers Inc., pp.2823– 2824.
23. Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P., (2017) Stock price prediction using LSTM, RNN and CNN-sliding window model. In: *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*. Institute of Electrical and Electronics Engineers Inc., pp.1643–1647.
24. Jia, H., (2016) Investigation Into The Effectiveness Of Long Short Term Memory Networks For Stock Price Prediction. [online] Available at: http://arxiv.org/abs/1603.07893 [Accessed 6 Jul. 2020].
25. Hoang, M., Alija Bihorac, O. and Rouces, J., (2020) *Article | Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa), September 30 - October 2, Turku, Finland | Aspect-Based Sentiment Analysis using BERT*. [online] Available at: https://ep.liu.se/ecp/article.asp?issue=167&article=020&volume= [Accessed 6 Jul. 2020].
26. Li, X., Bing, L., Zhang, W. and Lam, W., (2019) Exploiting BERT for End-to-End Aspect-based Sentiment Analysis. [online] pp.34–41. Available at: http://arxiv.org/abs/1910.00883 [Accessed 6 Jul. 2020].
27. Hiew, J.Z.G., Huang, X., Mou, H., Li, D., Wu, Q. and Xu, Y., (2019) BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability. [online] Available at: http://arxiv.org/abs/1906.09024 [Accessed 6 Jul. 2020].

28. Yang, L., Dong, R., Ng, T.L.J. and Xu, Y., (2019) *Leveraging BERT to Improve the FEARS Index for Stock Forecasting*. [online] Available at: https://www.aclweb.org/anthology/W19-5509/ [Accessed 6 Jul. 2020].
29. Sousa, M.G., Sakiyama, K., Rodrigues, L.D.S., Moraes, P.H., Fernandes, E.R. and Matsubara, E.T., (2019) BERT for stock market sentiment analysis. In: *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*. IEEE Computer Society, pp.1597–1601.
30. Anon (2020) *Technical Indicator Definition*. [online] Available at: https://www.investopedia.com/terms/t/technicalindicator.asp [Accessed 6 Jul. 2020].