ABSTRACT

STOCK MARKET PREDICTION USING INVESTOR SENTIMENT

by
Sarvesh Shukla

Stock market prediction has attracted not only business but academia as well. It is a research topic, to which many computational methods have been proposed, but desirable and reliable performance is yet to be attained. This study proposes a new method for stock market prediction, which adopts the Gated Recurrent Unit a deep neural network and incorporates investor sentiment to improve its forecasting performance. By extracting investor sentiment from news headlines using VADER sentiment, this paper makes it possible to analyze the irrational component of stock price. Our empirical study on DJIA index proves that our prediction method provides 6% better prediction compared to baseline models. Furthermore, our empirical study helps to better understand investor sentiment and stock behaviors. Finally, this work shows the potential of deep learning in forecasting a financial time series in the presence of strong noises.

STOCK MARKET PREDICTION USING INVESTOR SENTIMENT

by
Sarvesh Shukla

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

Department of Computer Science

May 2021

Blank Page

APPROVAL PAGE

STOCK MARKET PREDICTION USING INVESTOR SENTIMENT

Sarvesh Shukla

Guiling Wang, Thesis Advisor — Date
Professor, New Jersey Institute of Technology

Ioannis Koutis, Committee Member — Date
Associate Professor, New Jersey Institute of Technology

Alberto Martin-Utrera, Committee Member — Date
Assistant Professor of Finance, MT School of Management

BIOGRAPHICAL SKETCH

Author: Sarvesh Shukla

Degree: Master of Science

Date: May 2021

Undergraduate and Graduate Education:

- Master of Science in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2021
- Bachelor of Engineering in Information Technology,
  Birla Vishvakarma Mahavidyalaya, Anand, India, 2017

Major: Computer Science

I dedicate this thesis to my parents who allowed their son to pursue his dream remaining halfway across the world through what has possibly been the most challenging time in our lives so I could complete my master's degree.

Sarvesh Onkarnath Shukla

ACKNOWLEDGMENT

This thesis has really been a collaboration of the efforts of a lot of people and I owe a lot of it to them. I would firstly like to thank Professor Guiling Wang for believing in me by taking me under her guidance and introducing me to Deep Learning. I would also like to thank her for checking in regarding my well-being in these uncertain times and guiding me in the right direction during my thesis by offering invaluable advice, of which this thesis is a result. I would like to thank Professor Roshan Usman from whom I learnt a lot about Machine Learning during my first year at NJIT. Also, I would like to thank Jingyi Gu and Junyi Ye for guiding me during my thesis. Most importantly, I would like to thank my parents for providing me with the financial support that I needed during this thesis.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

In recent years, a whole industry has been formed around financial market sentiment detection [11]. More people have begun to carry out scientific and detailed research attempting to define how the stock market operates and extract features from all aspects for successful prediction of how market changes. In the beginning, research on stock market prediction was based on Efficient Market Hypothesis [10] and random walk theory. In Efficient Market Hypothesis, new information (i.e. news) has a major impact on stock prices than past stock market prices. Stock market will follow random walk pattern since news are not predictable. However, many researchers disagree with EMH [26]. Some studies are trying to measure the different effciency levels for mature and emerging markets, while other studies are trying to build effective prediction models for stock markets.

The effort begins with the work on fundamental analysis and technical analysis. Fundamental analysis evaluates the stock price based on its intrinsic value, i.e., fair value, while technical analysis only relies on the basis of charts and trends. The technical indicators from experience can be further used as handcrafted input features for machine learning and deep learning models. Afterwards, linear models are introduced as the solutions for stock market prediction, which include autoregressive integrated moving average (ARIMA)[19] and generalized autoregressive conditional heteroskedasticity (GARCH) [5]. With the development of machine learning models, they are also applied for stock market prediction, e.g., logistic regression and support vector machine [1].

In the past few years, both the basic tools for deep learning and the new prediction models are undergoing a rapid development. With the continuous improved programming packages, it becomes easier to implement and test a novel deep learning model. With the collection of online news or twitter data provides new sources of predicting stock market. The study for stock market prediction is not limited to the academia. Attracted by the potential profit by stock trading powered by the latest deep learning models, asset management companies and investment banks are also increasing their research grant for artificial intelligence which is represented by deep learning models nowadays.

Forecasting financial time series, which is highly volatile, like stock market is a challenging task specially when there is a strong noise. We need a good non-linear function, like Artificial Neural networks, which also have non-linear dependence on inputs[17]. Applying machine learning methods using time series has been implemented in past and recurrent neural network is the most popular method for this task. Work done on recurrent neural networks to predict stock market like [21] [4] which include public mood data and Dow Jones Industrial Average to predict up and down direction in stock market, where [14] reported volatility forecasting model.

Our focus in this study would be the latest emerging deep learning, which is represented by various structures of deep neural networks. With a strong ability of dealing with learning the nonlinear relationship between input features and prediction target, deep learning models have shown a better performance than both linear and machine learning models on the tasks that include stock market prediction.

Forecasting highly volatile stock returns could be challenging especially in the presence of strong noise. Thus, we choose duration of study to be from 2008, when was the beginning of recession to 2016, when showed recovery. This time frame is selected because it considers both ups and downs in stock market values, further the implementation of deep learning algorithms with investor sentiments is to learn the complexity and show the effect of investor sentiment as a feature on predicting stock market values. Our contribution in this study are listed as follows

- We collected news headlines to calculate sentiment scores and using these scores to create investor sentiments.
- We are using investor sentiment as a feature which tells us how an investor is expressing sentiment on news, while most study before applied public mood and tweets to define sentiment ignoring the sentiment of investor..
- We applied deep learning models, LSTM and GRU to solve the complexity of prediction of stock market.

The rest of the study is organized as follows: Section 2 presents related work; Section 3 gives technical background required for this study; Section 4 describes our prediction framework ,the preparation of dataset and evaluation of baseline models and GRU and Section 6 discusses future work and conclusion.

CHAPTER 2

RELATED WORK

Predicting the stock market using an econometric model is one of the many areas of research. To evaluate conditional volatility and expected return [9] generalized autoregressive conditional heteroscedasticity (GARCH) and investor sentiment for testing noise trader risk was used by Lee et al [23]. Principal component analysis was used by Baker and Wurgler for construction of investor sentiment index and predicting that for securities whose valuations were highly subjective and difficult to arbitrage had a larger effect on securities [3]. Brown and Cliff [6] investigated relation between investor sentiment to near-term stock market returns and found that sentiment levels and changes were strongly correlated with market returns, the result showed having a little predictive power for near term stock returns.

Different technical approaches to market trend prediction have been proposed in the research literature, ranging from AutoRegressive Integrated Moving Average (ARIMA) [29, 37] to ensemble methods [30]. Huang et al.[38] in their work demonstrated the superiority of Support Vector Machines (SVM) in forecasting weekly movement directions of the NIKKEI 225 index, and Lin et al. [41] managed to achieve 70% accuracy by combining decision trees and neural networks. Recent advances in deep learning have brought a new wave of methods [20, 13] to this field. In particular, the Long-Short Term Memory (LSTM) recurrent neural network has been shown to be very effective.

Recent theoretical studies in behavioral finance revealed that emotion does effect decisions made for investment [9, 34]. Hence, to assume public mood and sentiment can drive stock market values. Li [24] and Schumaker et al. [32] findings confirmed that news articles and financial reports affect stock market values.

Therefore, to study how stock market is influenced by investor sentiment, we need to know public mood in early stage, which is reliable and scalable for stock market prediction. In past years, progress in sentiment analysis techniques which can extract indicators of public mood from blogs and forum text has been significant. Tetlock et al. [35] and Chen et al. [7] finds that views, negative in particular, from social media and news forecast impacts on firm's earning and stock returns. Antweiler and Frank [39], used Yahoo Finance and RagingBull.com downloaded text messages on large firms in 2000 and confirmed that investor sentiment from Internet posting messages is powerful in predicting volatility and trading volume.

Using natural language processing and capturing media influence on stocks to bridge some connection is another area of research. Early work is done by [21, 4] to predict stock market value using public mood data to classify if the value will move upwards or downwards. News with proper nouns was most effective compared to other textual representative as experiment by Schumaker and Chen[32]. Google Domestic trends as indicators of public mood and S& P 500 volatilities as inputs to Long Short-Term Memory outperforming other neural network models by 31 % by Xiong et al [43].

Li & Ma [25] gave a survey on the application of artificial neural networks in forecasting financial market prices, including the forecast of stock prices, option pricing, exchange rates, banking and financial crisis. Nikfarjam et al. [27] surveyed some primary studies which implement text mining techniques to extract qualitative information about companies and use this information to predict the future behavior of stock prices based on how good or bad the news evaluated these companies.

Verner [36] provided a systematic overview of neural network applications in business between 1994 and 2015 and revealed that most of the research aimed at financial distress and bankruptcy problems, stock price forecasting, and decision support, with special attention to classification tasks. More recently, Xing et al.(2018)

[42] reviewed the application of cutting-edge NLP techniques for financial forecasting, which would be concerned when text including the financial news or twitters is used as input for stock market prediction.

Rundo et al. [31] covered a wider topic in the machine learning techniques, which include deep learning, and the field of quantitative finance from HFT trading systems to financial portfolio allocation and optimization systems. Nti et al. [28] focused on the fundamental and technical analysis, and found that support vector machine and artificial neural network are the most used machine learning techniques for stock market prediction. Sezer et al. [33], focused on deep learning for financial time series forecasting and a much longer time period (from 2005 to 2019 exactly), we focus on the recent progress in the past three years (2017-2019) and a narrower scope of stock price and market index prediction.

Numerous studies have been carried out to understand the intricate relationship between sentiment and price on the financial market by Wang et al. [12]. In most research based on deep learning are only using stock history data and market trading indicators as input variables. In this work, we integrate related market data and investor sentiment into a deep learning model to forecast future stock price.

In our work, different types of recurrent neural networks (RNN) [40] were used like long short-term memory (LSTM) [16], Gated recurrent unit (GRU) [8]. Different network has different mechanism like LSTM has dynamic gating mechanism, GRU is like LSTM but lacks output gate and shown to perform better on smaller and less frequent datasets.

CHAPTER 3

TECHNICAL BACKGROUND

3.1 VADER Sentiment Analysis

VADER (Valence Aware Dictionary and sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media [18]. The VADER sentiment lexicon is sensitive to both the polarity and the intensity of sentiments expressed in social media contexts and is also generally applicable to sentiment analysis in other domains. Over 9,000 token features were rated on a scale from (–4) Extremely Negative to (4) Extremely Positive, with allowance for (0) Neutral (or Neither, N/A). We kept every lexical feature that had a non-zero mean rating, and whose standard deviation was less than 2.5 as determined by the aggregate of those ten independent raters. This left us with just over 7,500 lexical features with validated valence scores that indicated both the sentiment polarity (positive/negative), and the sentiment intensity on a scale from –4 to +4. For example, the word "okay" has a positive valence of 0.9, "good" is 1.9, and "great" is 3.1, whereas "horrible" is –2.5, the frowning emoticon :(is –2.2, and "sucks" and it's slang derivative "sux" are both –1.5. VADER performs as good as individual human raters at matching ground truth. Further inspecting the F1 scores (classification accuracy), we see that VADER (0.96) outperforms individual human raters (0.84) at correctly labelling the sentiment of tweets into positive, neutral, or negative classes.

3.2 Investor Sentiment

After calculating the sentiment scores of each news headline, we are constructing investor sentiment using Antweiler and Frank [39], we measure investor sentiment based on explicitly revealed sentiment. Explicit expressions of a sentiment are subjective utterances of a positive or negative opinion about a certain topic or object. First revealed sentiment measure is defined as

$$B_t = (M_t^{pos} - M_t^{neg})/(M_t^{pos} + M_t^{neg}) \tag{3.1}$$

where, $M_t^{pos} = \sum_{(i \epsilon D(t))} w_i x_i^{pos}$ denotes the weighted sum of post of type positive in the time interval D(t) where w_i the weight of the post i.e. the sentiment score of a news headline and x_i^{pos} is an indicator variable that is 1 when post is positive and 0 otherwise, same is for M_t^{neg}.

To calculate M_t^{pos} for one day's new headlines, first we will classify the indicator variable x_i, as 1 for positive and 0 for negative, where positive sentiment score is greater than negative sentient score. Then we calculate w_i summing up all the sentiment scores where indicator variable is 1 for M_t^{pos}. Similarly we calculate M_t^{neg} using indicator variable as 0.

The second revealed sentiment measures number of traders expressing a particular sentiment.

$$B_t^* = ln[(1 + M_t^{pos})/(1 + M_t^{neg}) \tag{3.2}$$

There are two types of investor sentiment, while sentiment 1 expresses how an investor will explicitly reveals his sentiment towards a news headlines. Sentiment 2, expresses how many investors are revealing the same sentiment towards a news headline.

3.3 Long Short-Term Memory

This is a popular Recurrent Neural Network architecture, which was introduced by Sepp Hochreiter and Juergen Schmidhuber [16] as a solution to vanishing gradient problem. They work to address the problem of long-term dependencies. That is, if the previous state that is influencing the current prediction is not in the recent past, the RNN model may not be able to accurately predict the current state. As an example, let's say we wanted to predict the italicized words in following,"Alice is allergic to nuts. She can't eat peanut butter." The context of a nut allergy can help us anticipate that the food that cannot be eaten contains nuts. However, if that context was a few sentences prior, then it would make it difficult, or even impossible, for the RNN to connect the information. To remedy this, LSTMs have "cells" in the hidden layers of the neural network, which have three gates–an input gate, an output gate, and a forget gate. These gates control the flow of information which is needed to predict the output in the network. For example, if gender pronouns, such as "she", was repeated multiple times in prior sentences, you may exclude that from the cell state.

Long short-term memory has dynamic gating mechanism and solves the long-term dependency and vanishing gradient problem of recurrent neural network. Here, I_i, which we interpret as the information flow of market sensitivity. I_i has a memory of past time information [16] and learns to forget through equation,

$$I_i = f_i \odot I_{i-1} + c_i \odot I_i \tag{3.3}$$

Here, f_i is the fraction of past-time information passed over to the present, I_i measures the information flowing in at the current time and c_i is the weight of how important this current information is, Equation (5) answers the fundamental question of memory in time series forecasting which is equivalent to evaluating autocorrelation and partial autocorrelation functions to determine the p and q maximum lags in autoregressive moving average model(ARMA(p,q)) [22].

LSTM's are very powerful in sequence prediction problems because they're able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price. The main advantage of an LSTM is its

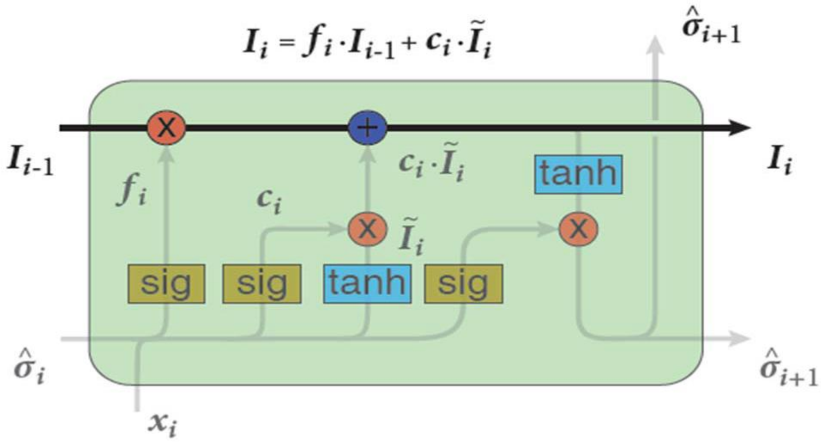

Figure 3.1 Long short term memory (LSTM) cell with inside mechanism and equation to calculate each cell state.

Source: Xiong, Ruoxuan, Eric P. Nichols, and Yuan Shen. "Deep Learning Stock Volatility with Google Domestic Trends." arXiv preprint arXiv:1512.04916 (2015).

ability to learn context-specific temporal dependence. Each LSTM unit remembers information for either a long or a short period of time without explicitly using an activation function within the recurrent components. For stock market prediction, this is very important as very considering t-5 days of data to predict t, so if the model is able to remember information for the previous 5 days to predict the next day, hence it is very much efficient for prediction of stock prices.

3.4 Gated Recurrent Unit

The Gated Recurrent Unit (GRU) cell contains only two gates: The Update gate and the Reset gate. These gates are essentially vectors containing values between 0 and 1 which will be multiplied with input data and hidden state. 0 value indicates that vector corresponding the data is unimportant while a value 1, in the gate vector means that corresponding data is important and will be used.

Reset gate is derived and calculated using both the hidden state from the previous timestep and the input data at the current time step. This is achieved

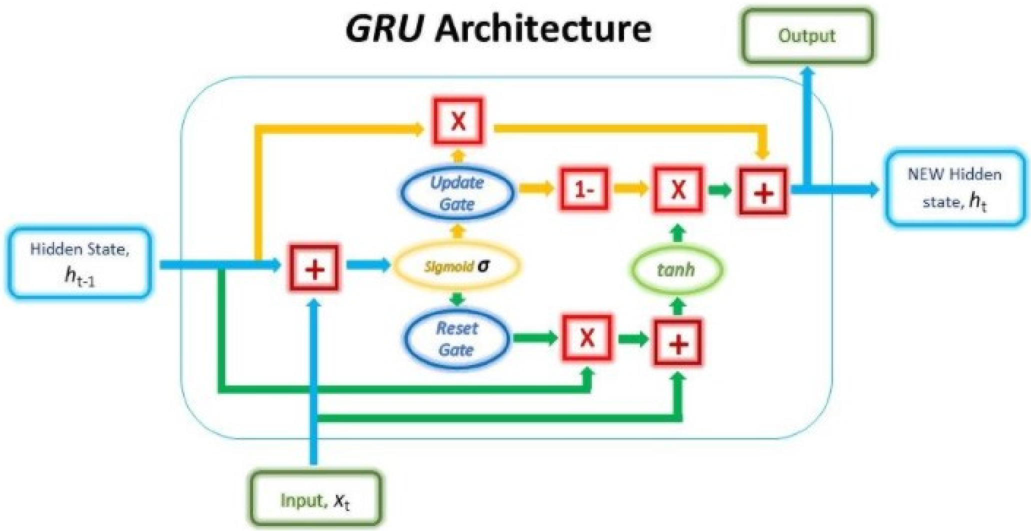

Figure 3.2 Gated recurrent unit cell mechanism is shown here, illustrating update and reset gate and how information flows in a cell.

Source: https://blog.floydhub.com/gru-with-pytorch/

by multiplying the previous hidden state by a trainable weight and an element-wise multiplication with reset vector. Only the important information will be decided and kept being used with new inputs. Current input will also be multiplied by a trainable weight before being summed with product of reset vector.

$$r = \tanh(gate_{reset} \odot (W_{h1} \cdot h_{t-1}) + W_x \cdot x_t \tag{3.4}$$

Update Gate is computed using the previous hidden state and current input data. Weights are multiplied with input and hidden states are unique meaning final vectors to each state are different.

$$gate_{update} = \sigma(W_{input_{update}}) \cdot x_t + W_{hidden_{update}} \cdot h_{t-1}) \tag{3.5}$$

Then performing element-wise inverse version of the same update vector (1 – update gate) and doing an element-wise multiplication with our output from the reset gate, r. The result from these operations will be summed with our output from the update gate in the previous step, u. The new updated hidden state will be our output for that time step,

$$h_t = r \odot (1 - gate_{update}) + u \tag{3.6}$$

CHAPTER 4

OUR PREDICTION

4.1 Deep Neural Network Architecture

Our Prediction framework is illustrated in Figure 4.1. In our prediction framework, we started with collecting new headlines from Reddit for the duration of 11^{th} August, 2008 to 21^{st} July 2016. For each day, we collected top 25 news headlines. This duration for predicting stock market values is selected because of the recession in 2008 and how the market recovers from the recession is all within this time frame.

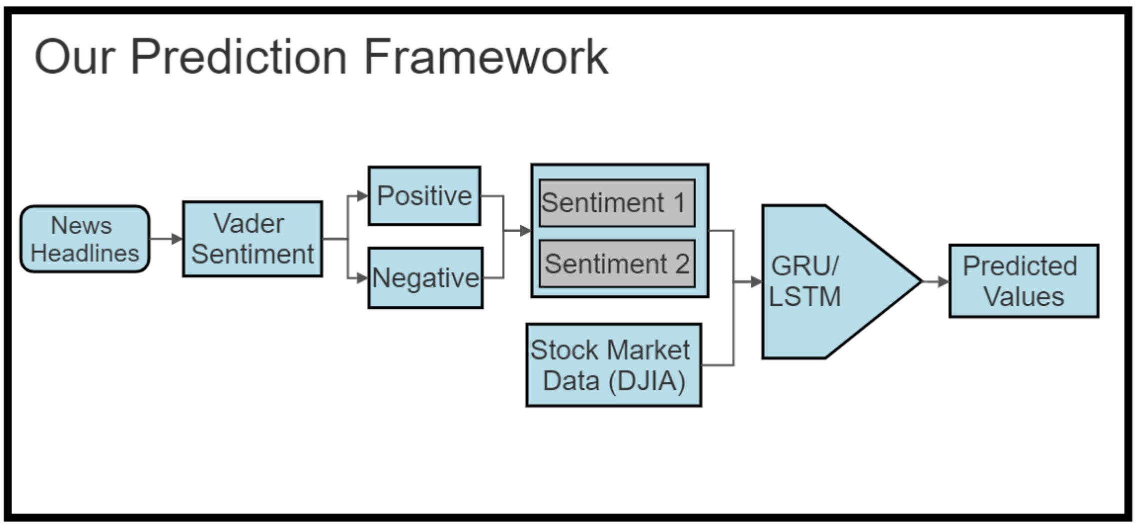

Figure 4.1 Our prediction framework indicating flow diagram of how features were created and applied to GRU for predicting values.

Then, we applied vader sentiment for each news headline and calculating positive, negative scores for each news headlines. Using these sentiment scores, we calculated two types of investor sentiment sentiment 1 and sentiment 2. We extracted stock market data of Dow Jones Industrial Average from yahoo finance for the duration as same as news headlines. Using both investor sentiments as features, sentiment 1 and sentiment 2 individually, along with stock market values we trained the model using Long short-term memory (LSTM) and Gated Recurrent Unit(GRU) for predicting stock market values.

4.2 Dataset Preparation

The process of calculating investor sentiments and extracting Dow Jones Industrial Average (DJIA) stock market values is as follows:

- Top 25 news headlines from 11th August 2008 to 21 June 2016 containing 2872 days was collected from Reddit.
- For each headline, Vader sentiment was applied, and positive, negative, and neutral scores was calculated.
- Now there are averaged and non-averaged sentiment values with two types of investor sentiment.
- In averaged sentiment, news sentiment scores from weekends and holidays along with working days were taken into consideration. The news sentiment of weekends and holidays were averaged to the next opening day of the market.
- In the non-averaged news sentiment, only the news sentiment of market opening days were considered as non-average sentiment
- Stock market values of Dow Jones Industrial Average was extracted from yahoo finance historical data API “yfinance”.
- Features from stock market values are Open, Low, High, Close, Adj Close and Volume. Investor sentiments were merged into stock market data as a feature.
- For each investor sentiment, there are five types of dataset, one of which is without sentiment using only stock market values.

- Averaged investor sentiment 1 is calculated using equation 1 with stock market values.

- Averaged investor sentiment 2 is calculated using equation 2 with stock market values.

- Non-averaged investor sentiment 1 is calculated using equation 1 with stock market values.

- Non-averaged investor sentiment 2 is calculated using equation 2 with stock market values.

- We apply Min-Max Scaler on three types of dataset as a data preprocessing step in the range of [0,1] for scaling features.

- Time series for training is 5 days. First 5 days data will be taken to predict next day value.

Finally, Dataset containing stock market values with two types of investor sentiment is generated. Dataset contains:

- In total: 1980 days of stock market values.

- 1980 investor sentiment of two different types using eq(3.1) and eq(3.2).

- Features: Open, Low, High, Close, Adj Close, Volume and Investor Sentiment.

4.3 Baseline Assumption

To test the effectiveness of our model, we implemented the following models as baseline. The models are Exponential moving average (EMA) [15], Autoregressive Moving Average (ARIMA) [2] and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) [5] which are widely used as baseline for other work.

4.3.1 Exponential Moving Average

The exponential moving average is a technical chart indicator that tracks the price of stock over time. It is a weighted moving average, which gives more weighting to recent price data. It is calculated as,

$$EMA = Price(t) * k + EMA(y), \tag{4.1}$$

where t = today, y = yesterday, N = number of days and $k = 2/(N + 1)$. The weighting for most recent price is greater for a shorter period of EMA than longer period. Following is performance for EMA,

- EMA 2: MAE = 135.82, RMSE = 181.58, MAPE = 0.79
- EMA 3: MAE = 146.06, RMSE = 195.78, MAPE = 0.85
- EMA 5: MAE = 166.69, RMSE = 224.48, MAPE = 0.98

4.3.2 Auto Regressive Integrated Moving Average

ARIMA is a class of models that explains a given time series based on its past value, its own lags and lagged forecast errors, so that equation can be used to forecast future values. ARIMA(p,d,q) model, where

- p is the number of autoregressive terms
- d is the number of differences needed for stationary
- q is the number of lagged forecast errors in the prediction equation

$$(y_t) = \mu + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} - \theta_1 e_{t-1} - ... - \theta_q e_{t-p} \tag{4.2}$$

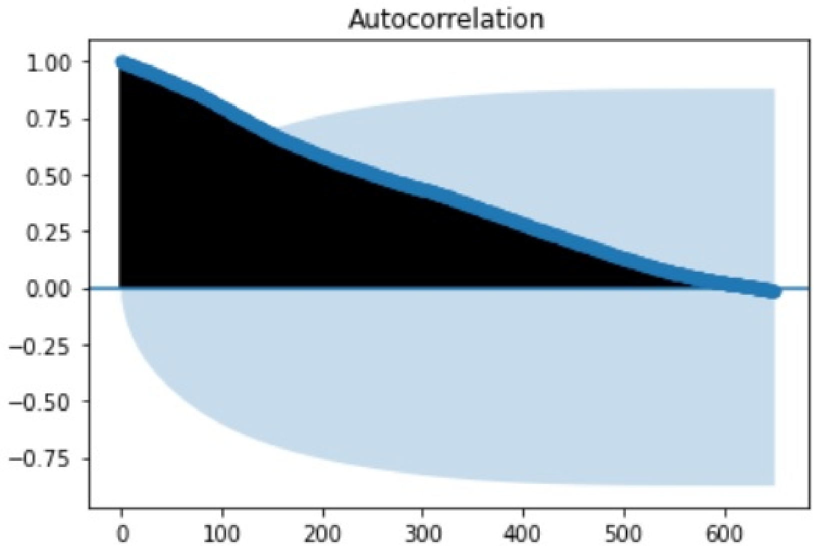

Figure 4.2 Arima auto-correlation. Auto-correlation measures linear relationship between lagged value of time series.

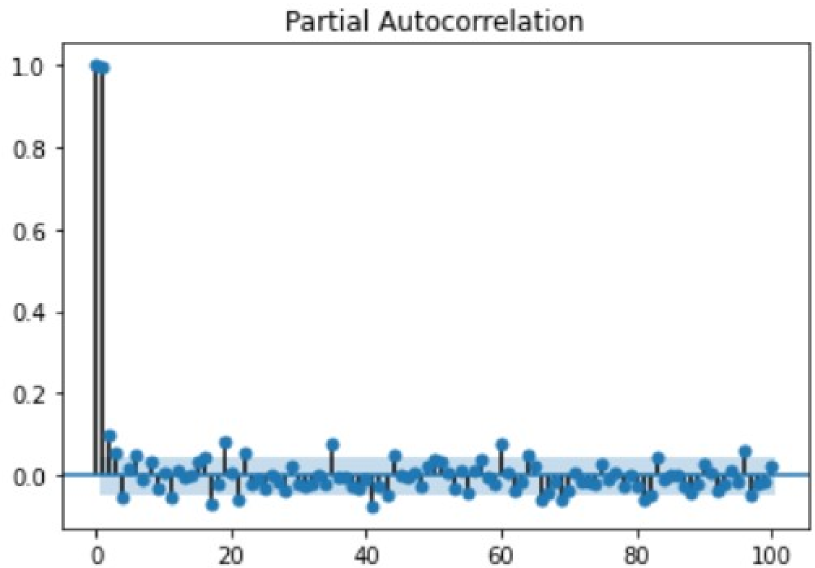

Figure 4.3 Arima partial auto-correlation. Partial autocorrelation function (PACF) gives the partial correlation of a stationary time series with its own lagged values, regressed the values of the time series at all shorter lags.

For ARIMA(p,d,q), where p = 1, d = 0 and q = 2. ARIMA predicted,

- Mean Absolute Error: 626.68
- Root Mean Square Error: 761.51

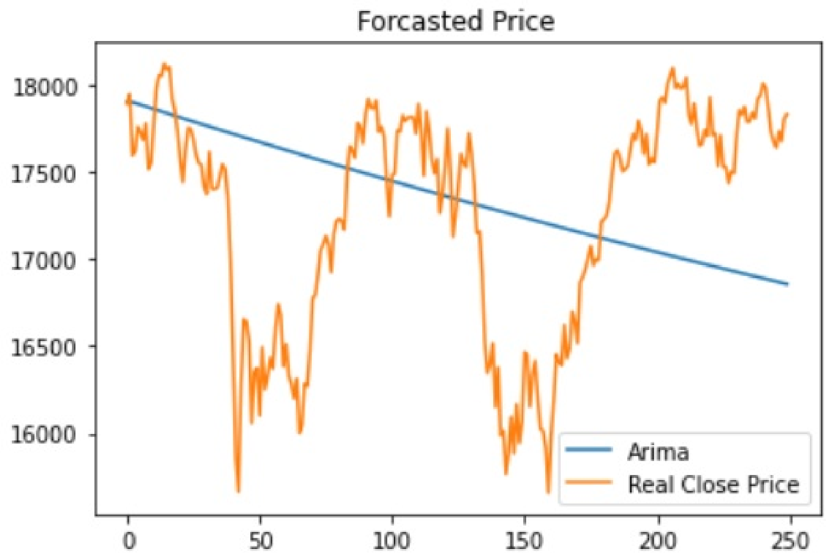

Figure 4.4 Arima prediction vs Real stock Price for testing period of 250 days.

4.3.3 Generalized AutoRegressive Conditional Heteroskedasticity

Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) is a statistical modeling technique used to help predict the volatility of returns in financial assets. It is suitable for time series data where variance of the error term is serially autocorrelated. GARCH(p,q) is described as follows,

- p: number of lag variance to include in the GARCH model.
- q: number of lag residual errors to include in the GARCH model.

For GARCH (2,0), model results are shown in Figure 4.5, Constant Mean - GARCH Model Results

Dep. Variable:	y	**R-squared:**	0.000
Mean Model:	Constant Mean	**Adj. R-squared:**	0.000
Vol Model:	GARCH	**Log-Likelihood:**	470.709
Distribution:	Normal	**AIC:**	-933.418
Method:	Maximum Likelihood	**BIC:**	-911.057
		No. Observations:	1979
Date:	Sun, Apr 25 2021	**Df Residuals:**	1978
Time:	21:11:42	**Df Model:**	1

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	0.5085	7.077e-03	71.855	0.000	[0.495, 0.522]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	1.4358e-03	6.446e-05	22.273	6.760e-110	[1.309e-03,1.562e-03]
alpha[1]	0.2000	8.233e-03	24.290	2.491e-130	[0.184, 0.216]
beta[1]	0.7799	8.850e-03	88.119	0.000	[0.763, 0.797]

Figure 4.5 Result Garch. The following result is achieved with p = 1 and q=1. Result shows AIC and BIC absolute values to be minimum and p-value <= 0.5.

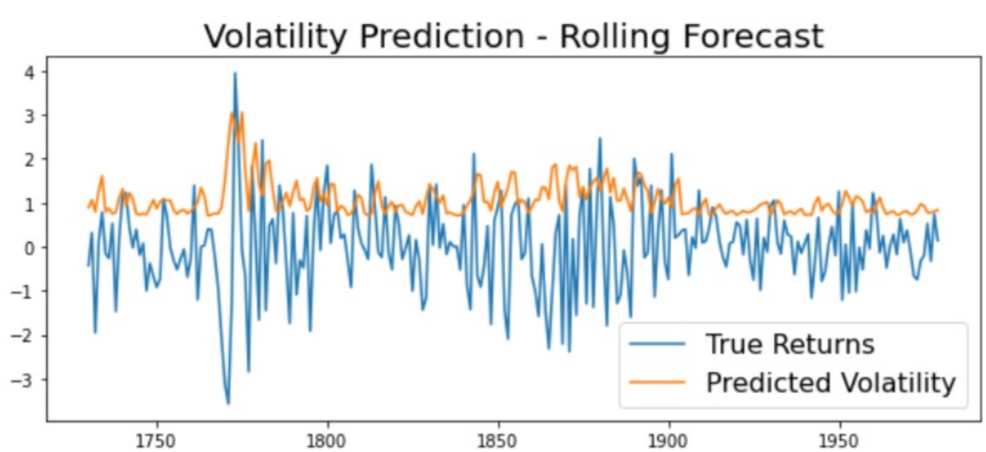

Figure 4.6 Volatility Prediction Garch. It shows the fluctuations in true returns as spike in volatility by the GARCH model.

4.3.4 LSTM

Long Short-Term Memory (LSTM) was trained with only stock market values. In first, there are only 6 features, Open, High, Low, Close, Adj Close and Volume.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 5, 300)            368400
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               160400
_________________________________________________________________
dense (Dense)                (None, 6)                 606
=================================================================
Total params: 529,406
Trainable params: 529,406
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.7 Model 1 (without sentiment): LSTM Architecture.

Figure 4.7 is the architecture of LSTM Model containing 529,406 parameters having 2 LSTM layers of 300 units and 100 units.

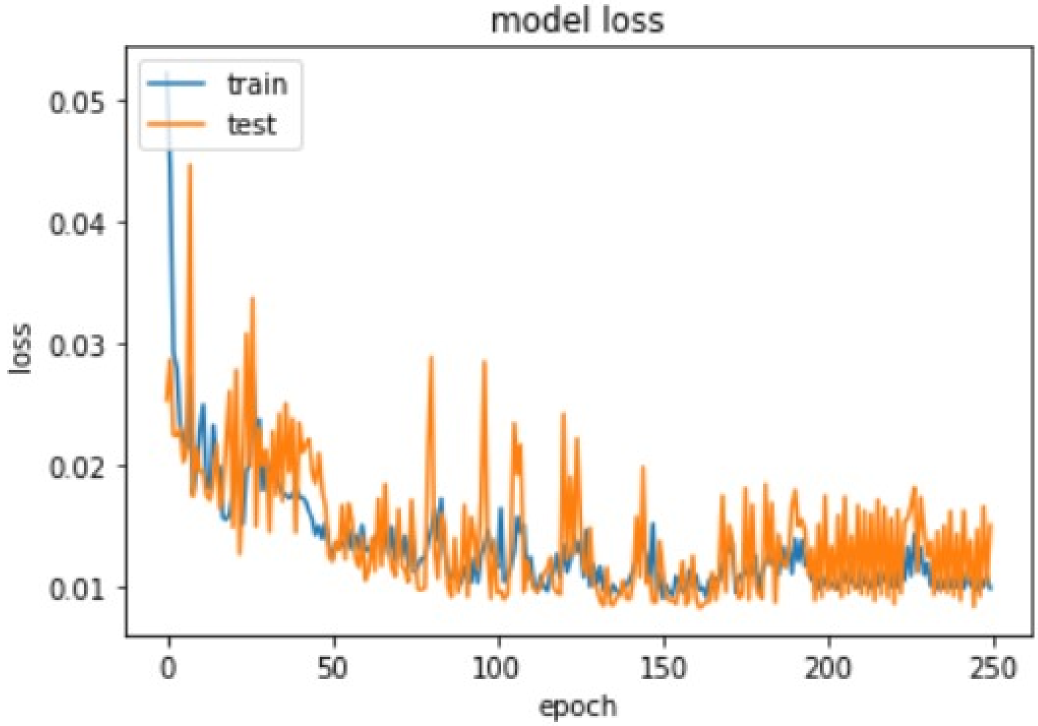

Figure 4.8 Model 1 (without sentiment): Training vs Validation loss.

Figure 4.8 illustrates the plot indicating the loss while training Model 1.

Plot for prediction of true values vs predicted values,

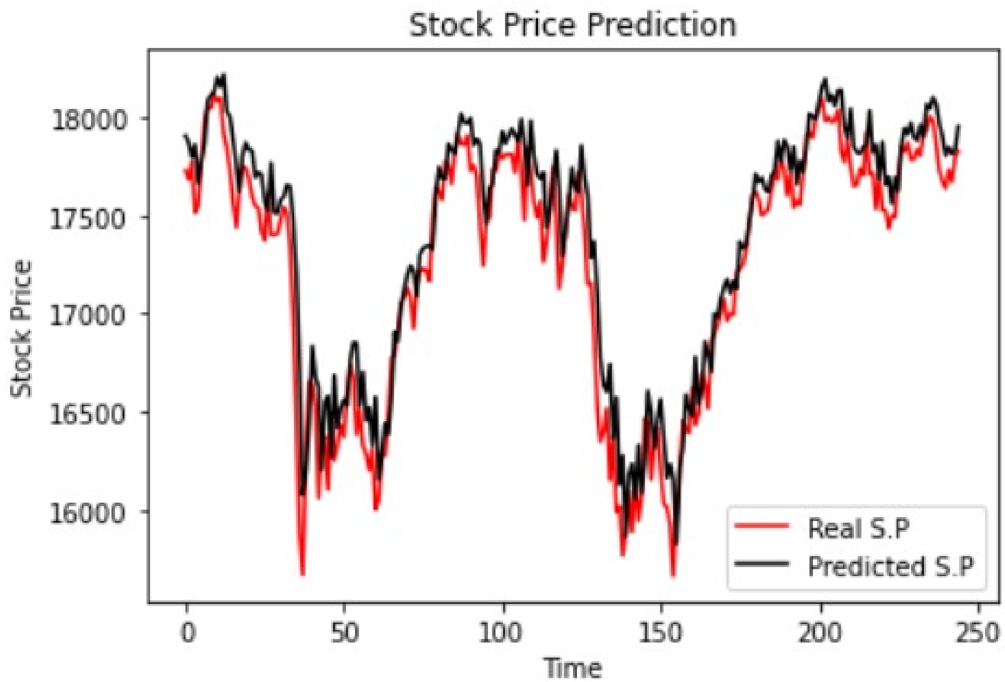

Figure 4.9 Model 1 (without sentiment): Real vs Predicted Close Price.

Figure 4.9 illustrated the performance of the model as predicted close price and comparing it with real close price.

4.3.5 GRU

Gated recurrent unit was trained with stock market values without investor sentiments. Training model architecture for GRU without investor sentiment is shown below,

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru (GRU)                    (None, 5, 300)            277200
_________________________________________________________________
gru_1 (GRU)                  (None, 100)               120600
_________________________________________________________________
dense (Dense)                (None, 6)                 606
=================================================================
Total params: 398,406
Trainable params: 398,406
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.10 Model 2 (without investor sentiment): Architecture.

Figure 4.10 is the architecture of GRU Model without investor sentiment having 399,407 parameters having 2 GRU layers of 300 units and 100 units.

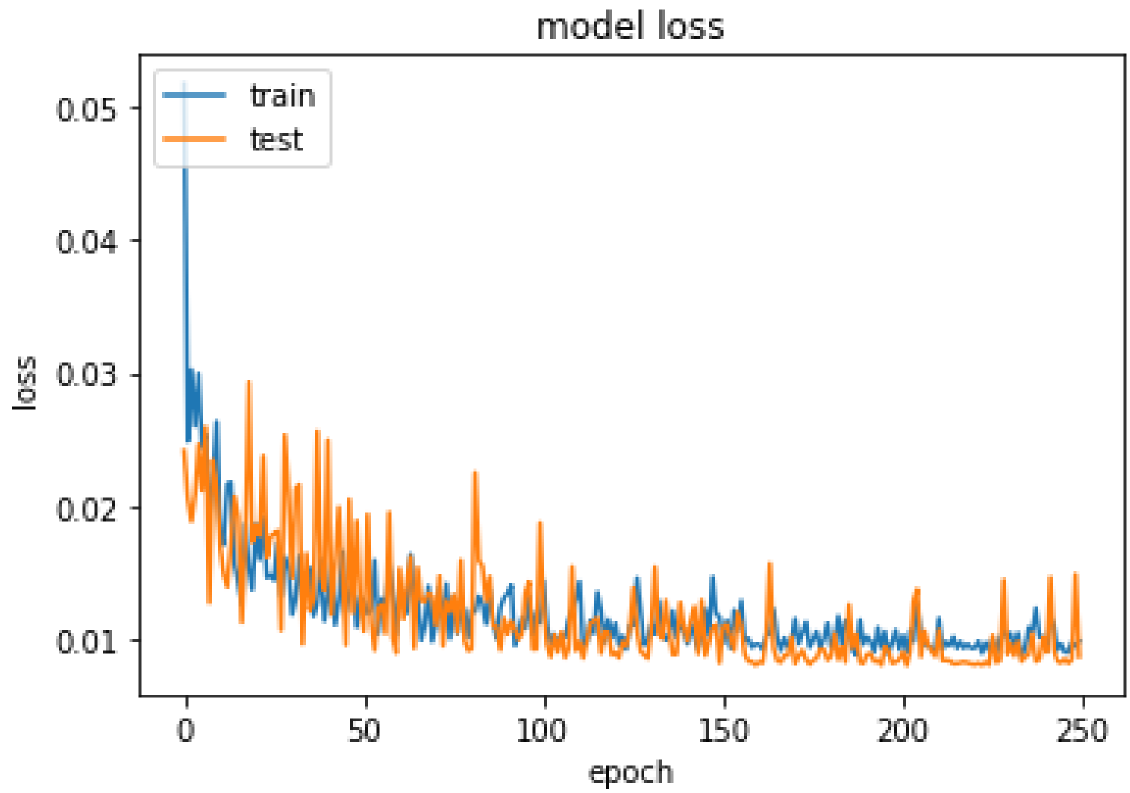

Figure 4.11 Model 2 (without investor sentiment): Training vs Validation loss.

Figure 4.11 illustrated the plot indicating the loss while training Model 2.

Plot for prediction of true values vs predicted values,

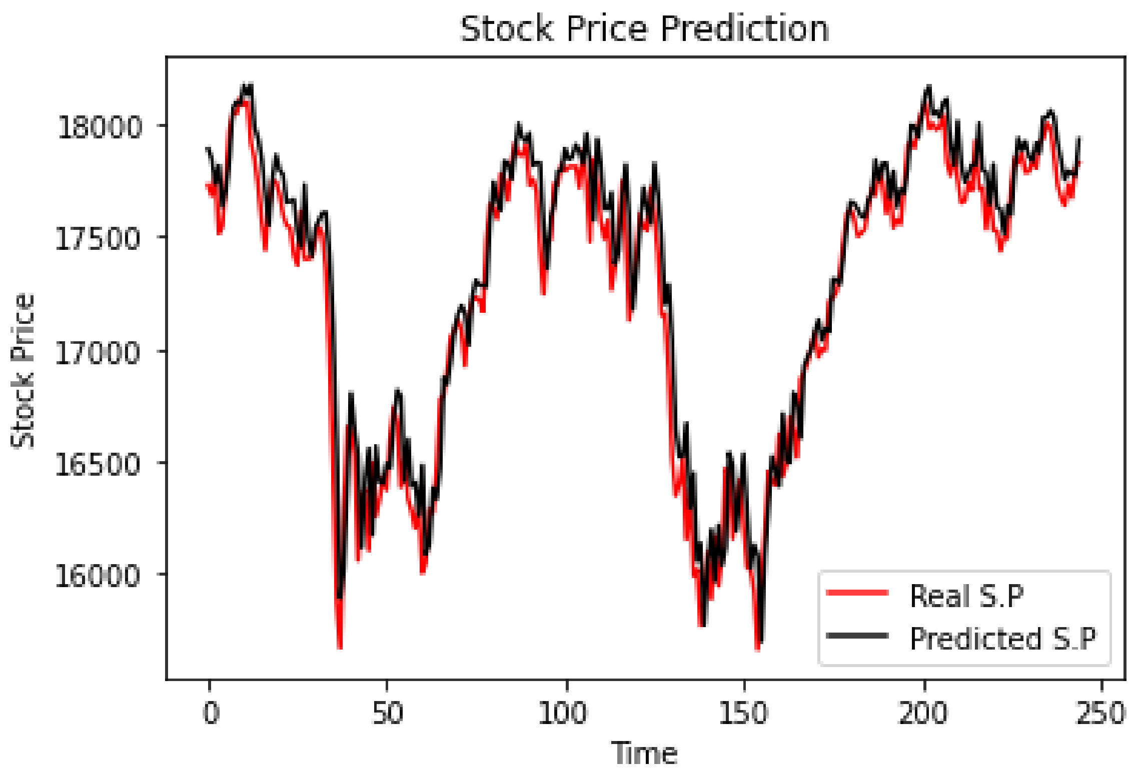

Figure 4.12 Model 2 (without investor sentiment): Real vs Predicted Close Price.

Figure 4.12 illustrated the performance of the model as predicted close price and comparing it with real close price.

4.4 Our Model

4.4.1 LSTM + Investor Sentiment

For stock market with averaged investor sentiment eq (1), we have used 7 features, Open, High, Low, Close, Adj Close, Volume and investor sentiment calculated from eq (1). Training model 1 Architecture is,

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_2 (LSTM)                (None, 5, 300)            369600
_________________________________________________________________
lstm_3 (LSTM)                (None, 100)               160400
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 707
=================================================================
Total params: 530,707
Trainable params: 530,707
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.13 Model 3 (with averaged investor sentiment 1): LSTM Architecture.

Figure 4.13 is the architecture of LSTM Model with averaged sentiment 1 having 530,707 parameters having 2 LSTM layers of 300 units and 100 units.

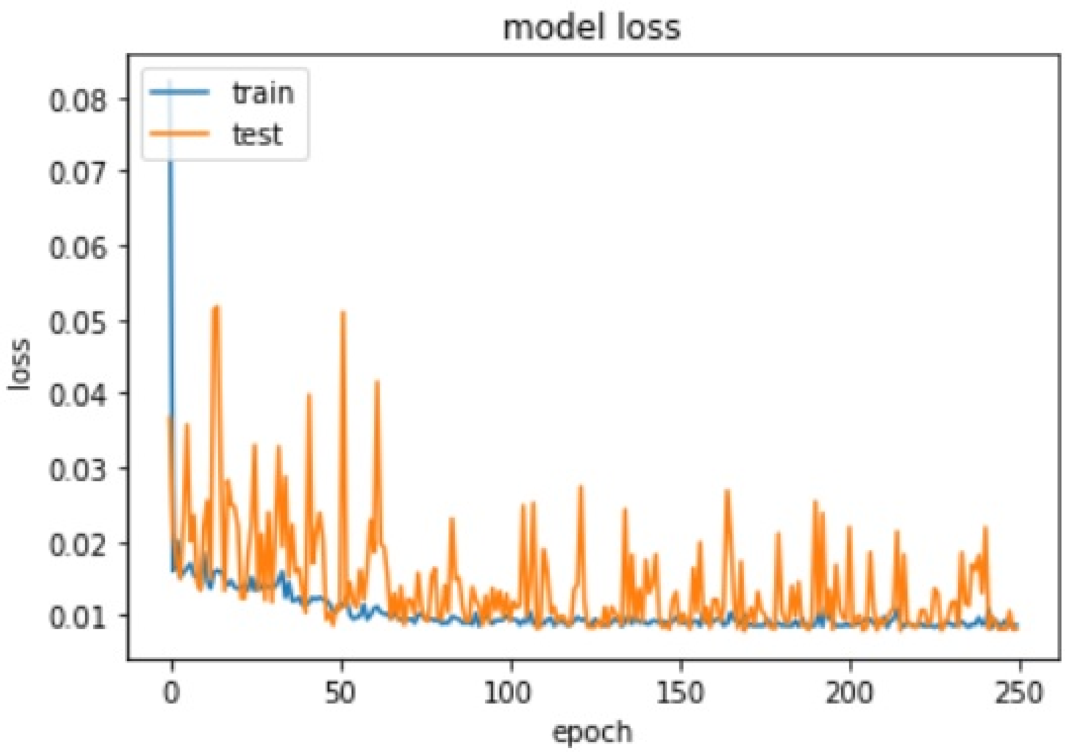

Figure 4.14 Model 3 (with averaged investor sentiment 1): Training vs Validation loss.

Figure 4.14 illustrated the plot indicating the loss while training Model 3.

Plot for prediction of true values vs predicted values,

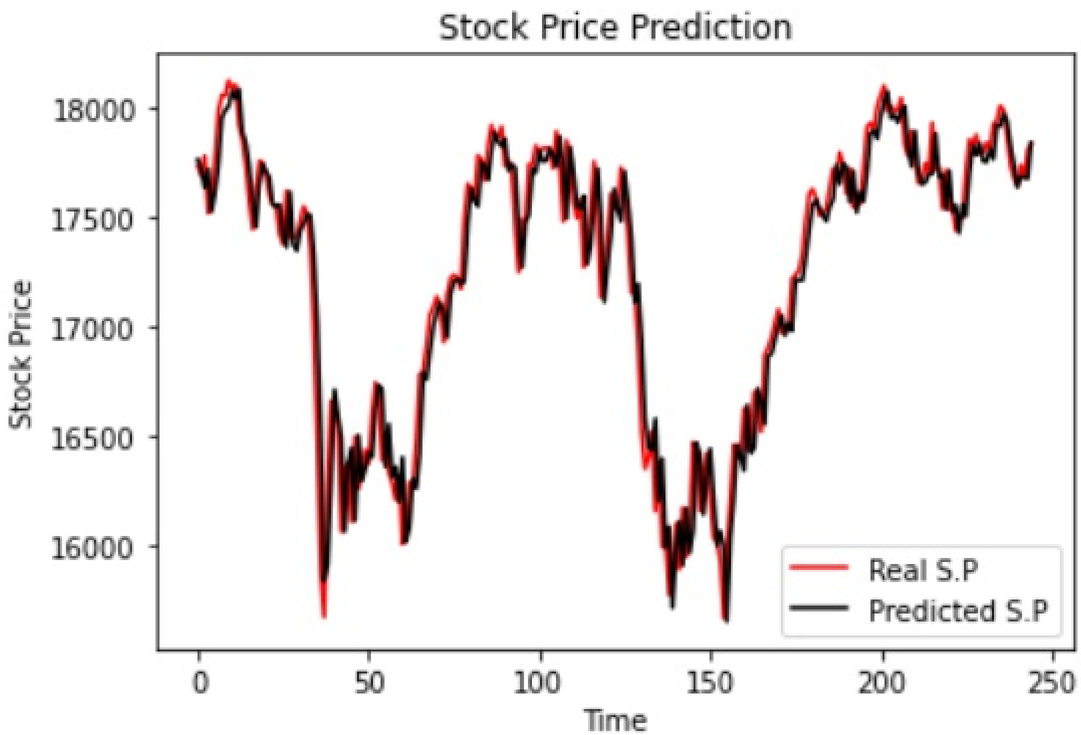

Figure 4.15 Model 3 (with averaged investor sentiment 1): Real vs Predicted Close Price..

Figure 4.15 illustrated the performance of the model as predicted close price and comparing it with real close price.

For stock market with averaged sentiment eq (2), we have used 7 features, 6 from stock market values and investor sentiment calculated from eq (2). Training model Architecture is,

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_5 (LSTM)                (None, 5, 300)            369600
_________________________________________________________________
lstm_6 (LSTM)                (None, 100)               160400
_________________________________________________________________
dense_5 (Dense)              (None, 7)                 707
=================================================================
Total params: 530,707
Trainable params: 530,707
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.16 Model 4 (with averaged investor sentiment 2): LSTM Architecture.

Figure 4.16 is the architecture of LSTM Model with averaged sentiment 2 having 530,707 parameters having 2 LSTM layers of 300 units and 100 units.

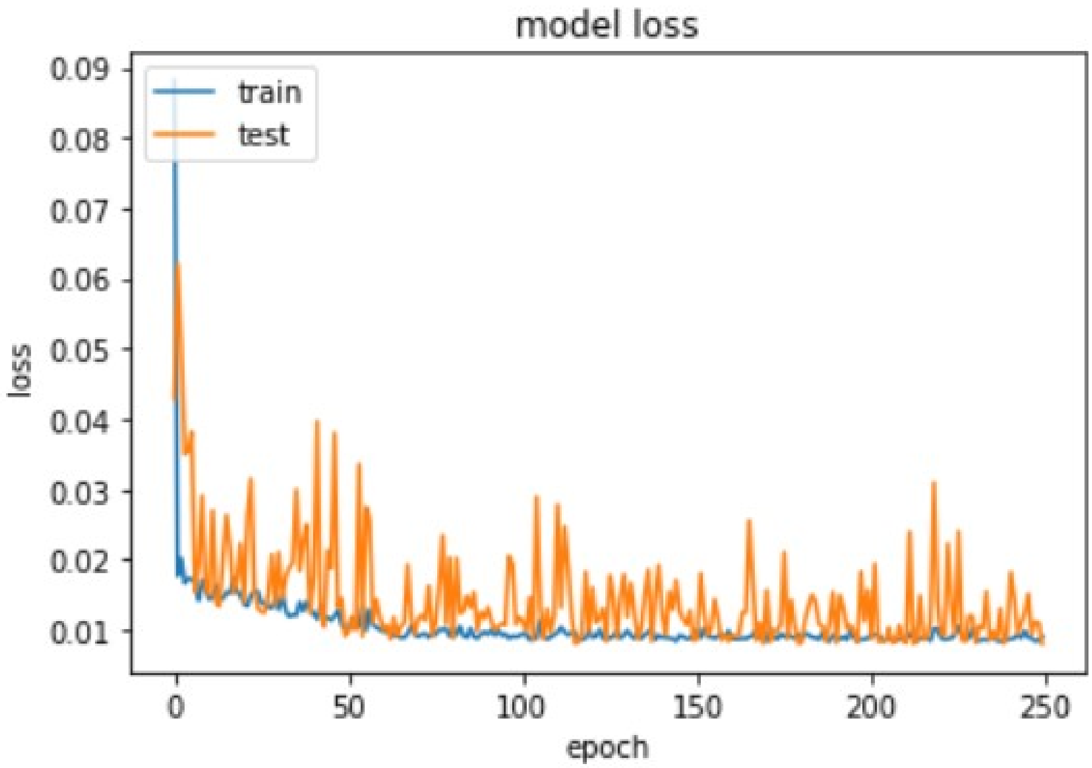

Figure 4.17 Model 4 (with averaged investor sentiment 2): Training vs Validation loss.

Figure 4.17 illustrates the plot indicating the loss while training Model 4.

Plot for prediction of true values vs predicted values,

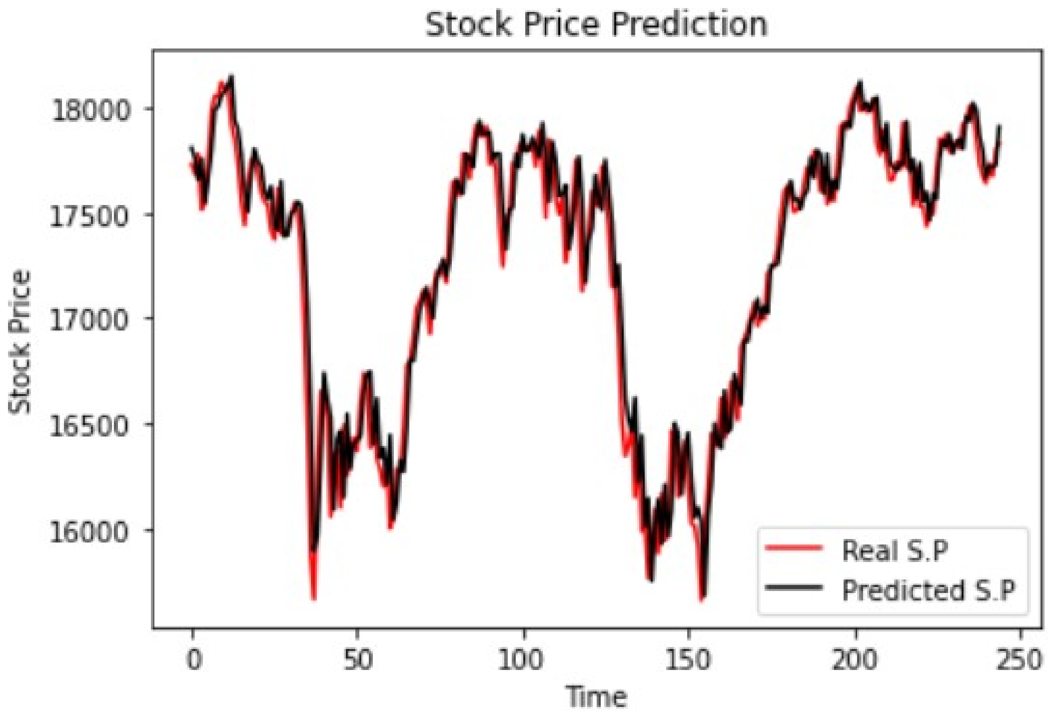

Figure 4.18 Model 4 (with averaged investor Sentiment 2): Real vs Predicted Close Price..

Figure 4.18 illustrated the performance of the model as predicted close price and comparing it with real close price.

For stock market with non-averaged sentiment eq (1), we have used 7 features, 6 from stock market values and investor sentiment calculated from eq (2). Training model Architecture is,

```
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_8 (LSTM)                (None, 5, 300)            369600
_________________________________________________________________
lstm_9 (LSTM)                (None, 100)               160400
_________________________________________________________________
dense_9 (Dense)              (None, 7)                 707
=================================================================
Total params: 530,707
Trainable params: 530,707
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.19 Model 5 (with non-average sentiment 1): Architecture.

This is the architecture of LSTM Model with non-averaged sentiment 1 having 530,707 parameters having 2 LSTM layers of 300 units and 100 units.

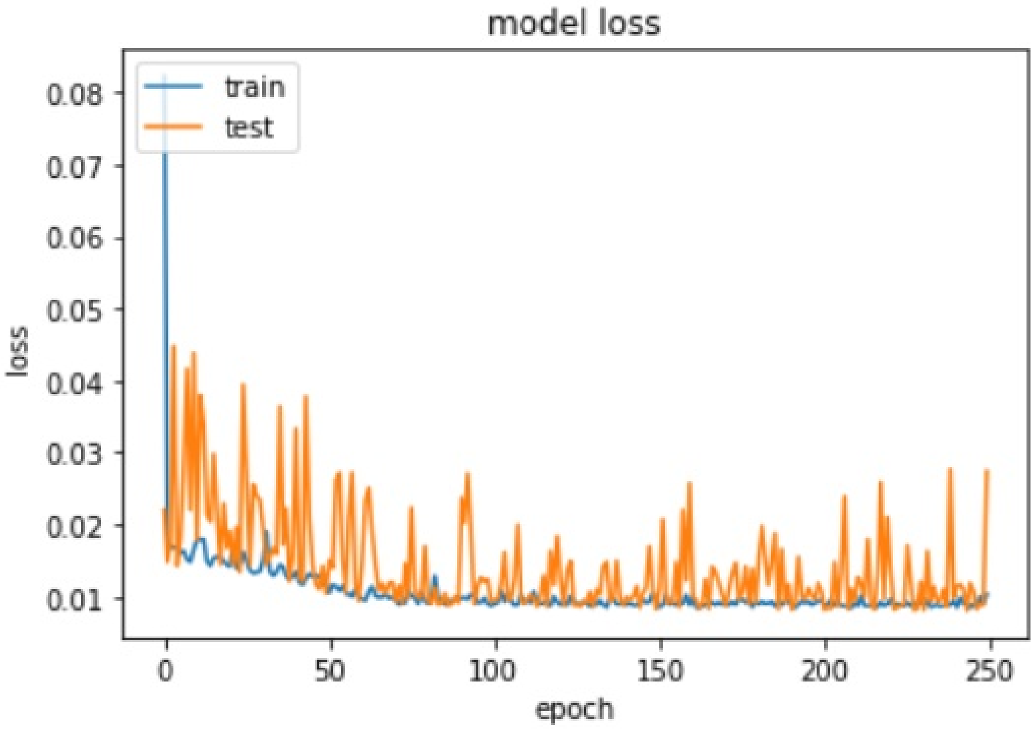

Figure 4.20 Model 5 (with non-average sentimment 1): Training vs Validation loss.

Figure 4.20 illustrates the plot indicating the loss while training Model 5.

Plot for prediction of true values vs predicted values,

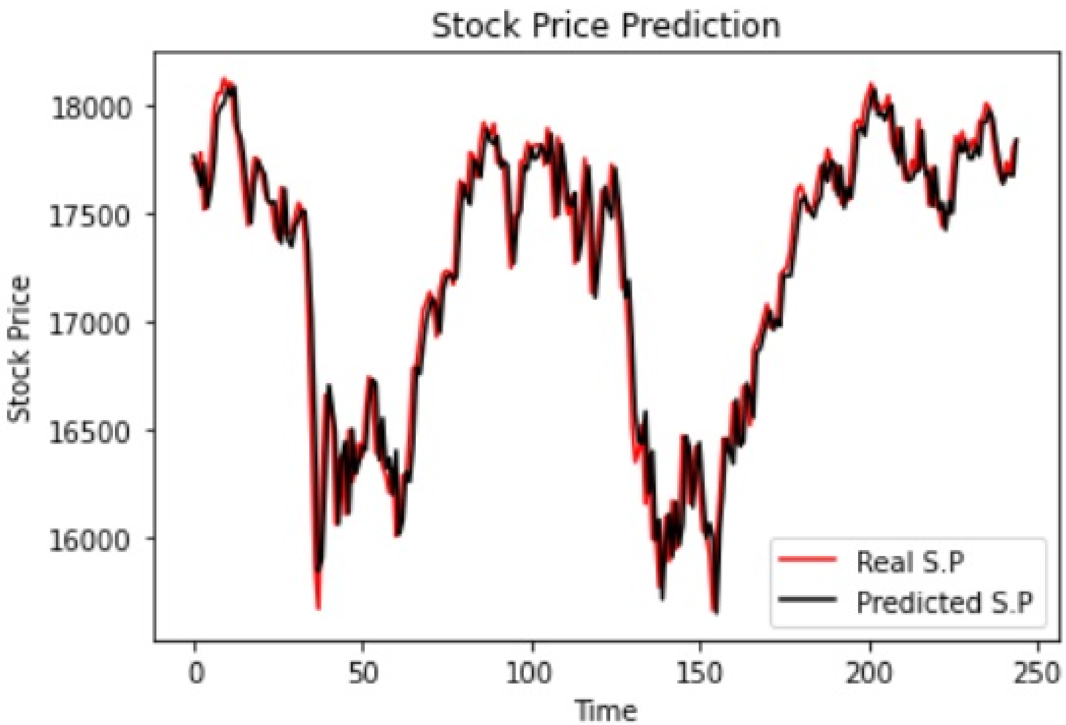

Figure 4.21 Model 5 (with non-average sentiment 1): Real vs Predicted Close Price..

Figure 4.21 illustrated the performance of the model as predicted close price and comparing it with real close price.

For stock market with non-averaged sentiment eq (2), we have used 7 features, 6 from stock market values and investor sentiment calculated from eq (2). Training model Architecture is,

```
Model: "sequential_13"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_11 (LSTM)               (None, 5, 300)            369600
_________________________________________________________________
lstm_12 (LSTM)               (None, 100)               160400
_________________________________________________________________
dense_13 (Dense)             (None, 7)                 707
=================================================================
Total params: 530,707
Trainable params: 530,707
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.22 Model 6 (with non-average sentiment 2): LSTM Architecture.

Figure 4.22 is the architecture of LSTM Model with non-averaged sentiment 2 having 530,707 parameters having 2 LSTM layers of 300 units and 100 units.

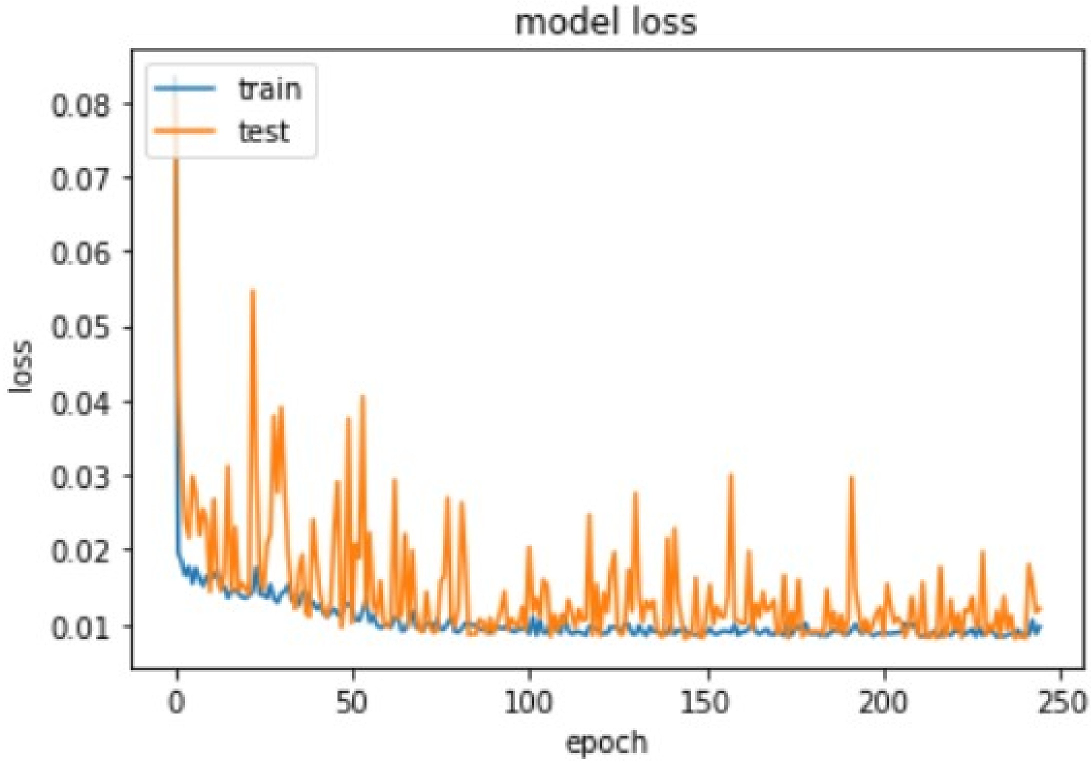

Figure 4.23 Model 6 (with non-average sentiment 2): Training vs Validation loss.

Figure 4.23 illustrates the plot indicating the loss while training Model 6.

Plot for prediction of true values vs predicted values,

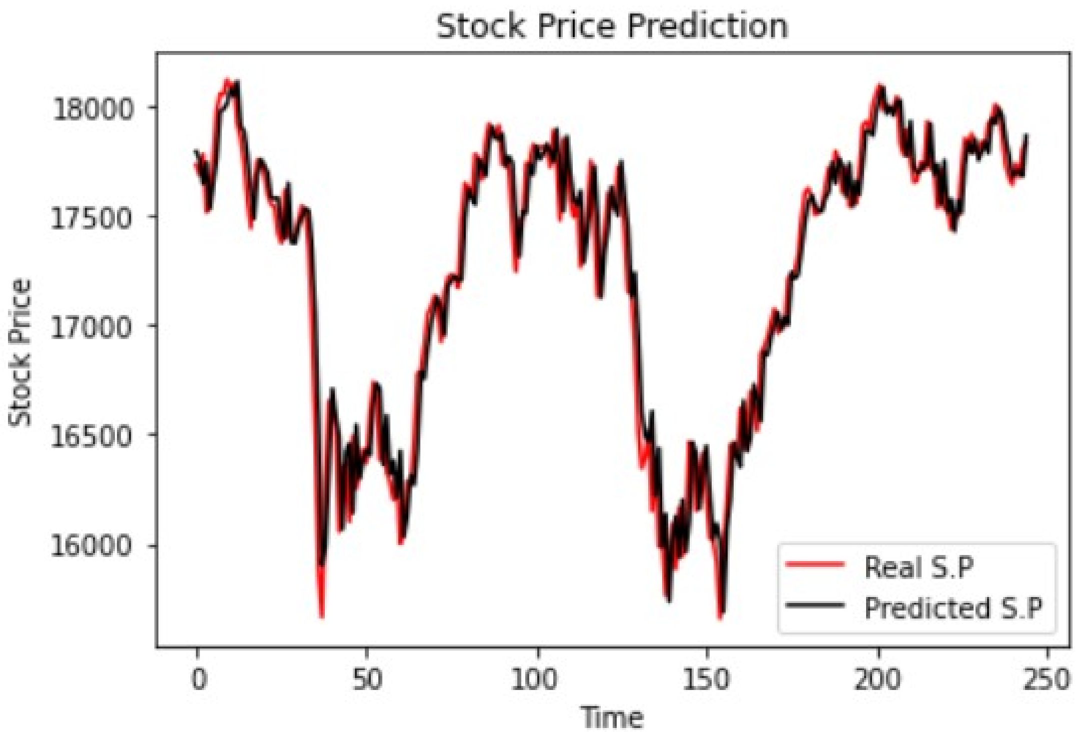

Figure 4.24 Model 6 (with non-average sentiment 2): Real vs Predicted Close Price..

Figure 4.24 illustrated the performance of the model as predicted close price and comparing it with real close price.

4.4.2 GRU + Investor Sentiment

Gated recurrent unit was trained with stock market values with investor sentiments. Training model architecture for GRU with averaged investor sentiment 1 is shown below,

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru (GRU)                    (None, 5, 300)            278100
_________________________________________________________________
gru_1 (GRU)                  (None, 100)               120600
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 707
=================================================================
Total params: 399,407
Trainable params: 399,407
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.25 Model 7 (with averaged investor sentiment 1): Architecture.

Figure 4.25 is the architecture of GRU Model with Averaged investor sentiment 1 having 399,407 parameters having 2 GRU layers of 300 units and 100 units.

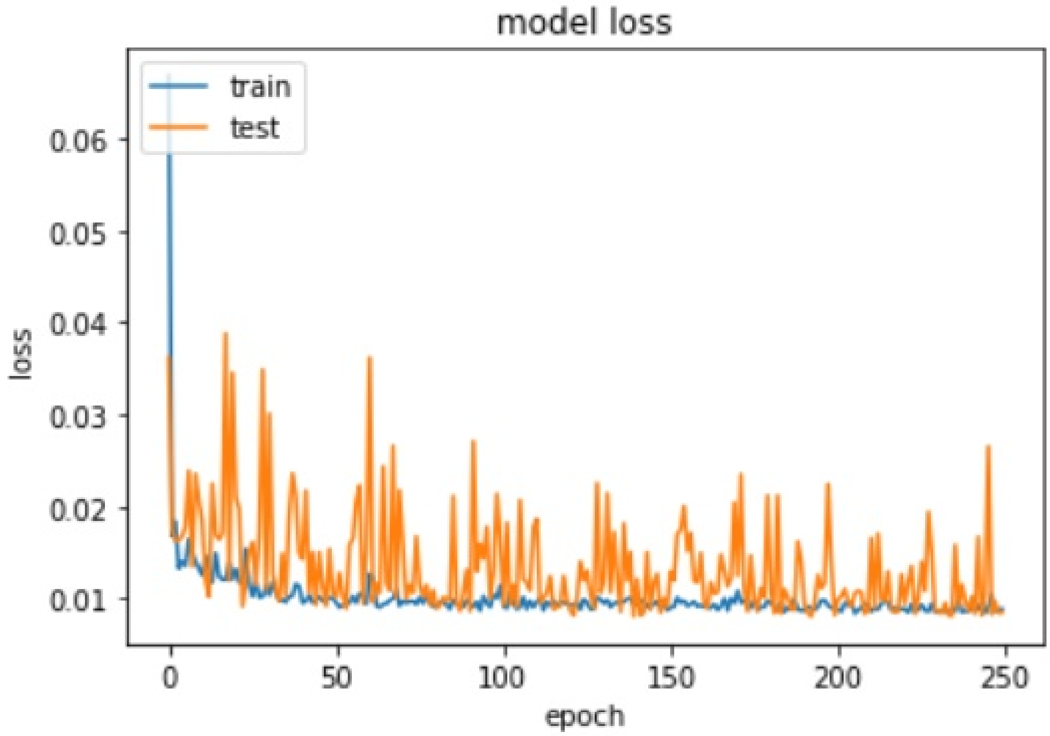

Figure 4.26 Model 7 (with averaged investor sentiment 1): Training vs Validation loss.

Figure 4.26 illustrates the plot indicating the loss while training Model 7.

Plot for prediction of true values vs predicted values,

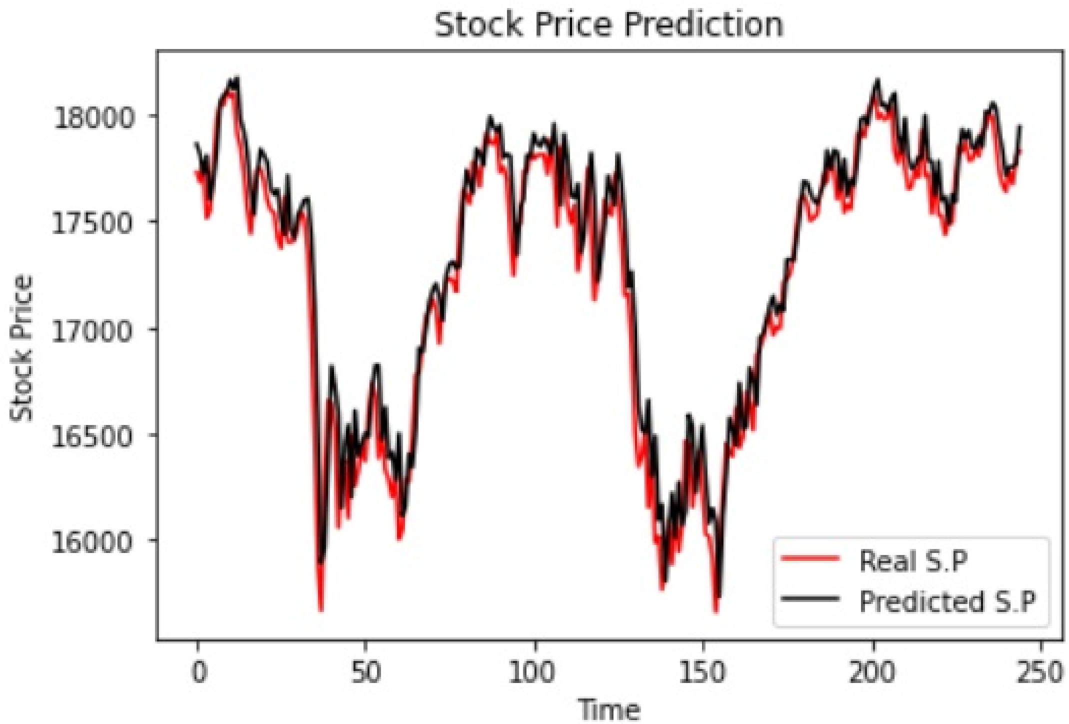

Figure 4.27 Model 7 (with averaged investor sentiment 1): Real vs Predicted Close Price.

Figure 4.27 illustrated the plot indicating the loss while training Model 7.

For stock market with averaged investor sentiment eq (2), we have used 7 features, 6 from stock market values and investor sentiment calculated from eq (2). Training model Architecture is,

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_2 (GRU)                  (None, 5, 300)            278100
_________________________________________________________________
gru_3 (GRU)                  (None, 100)               120600
_________________________________________________________________
dense_6 (Dense)              (None, 7)                 707
=================================================================
Total params: 399,407
Trainable params: 399,407
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.28 Model 8 (with averaged investor sentiment 2): GRU Architecture.

Figure 4.28 is the architecture of GRU Model with Averaged investor sentiment 2 having 399,407 parameters having 2 GRU layers of 300 units and 100 units.

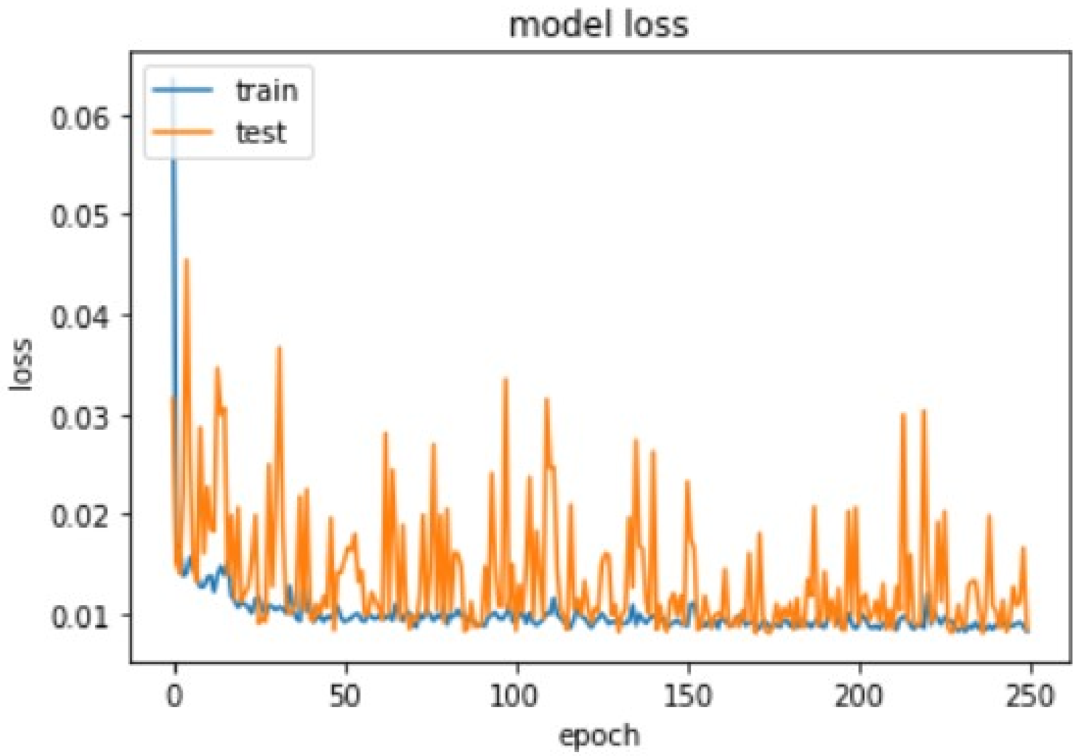

Figure 4.29 Model 8 (with averaged investor sentiment 2): Training vs Validation loss.

Figure 4.29 illustrates the plot indicating the loss while training Model 8.

Plot for prediction of true values vs predicted values,

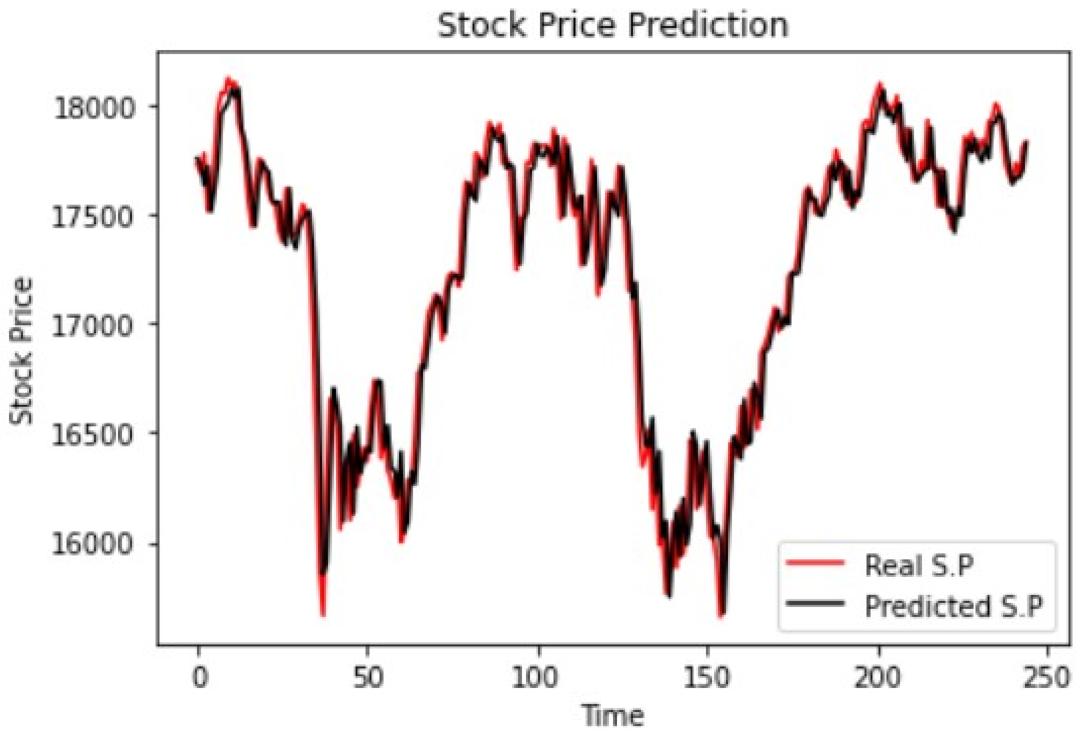

Figure 4.30 Model 8 (with averaged investor sentiment 2): Real vs Predicted Close Price.

Figure 4.30 illustrated the performance of the model as predicted close price and comparing it with real close price.

For stock market with non-averaged investor sentiment eq (1), we have used 7 features, 6 from stock market values and investor sentiment calculated from eq (2). Training model Architecture is,

```
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_4 (GRU)                  (None, 5, 300)            278100
_________________________________________________________________
gru_5 (GRU)                  (None, 100)               120600
_________________________________________________________________
dense_10 (Dense)             (None, 7)                 707
=================================================================
Total params: 399,407
Trainable params: 399,407
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.31 Model 9 (with non-averaged investor sentiment 1): GRU Architecture.

Figure 4.31 is the architecture of GRU Model with non-averaged investor sentiment 1 having 399,407 parameters having 2 GRU layers of 300 units and 100 units.

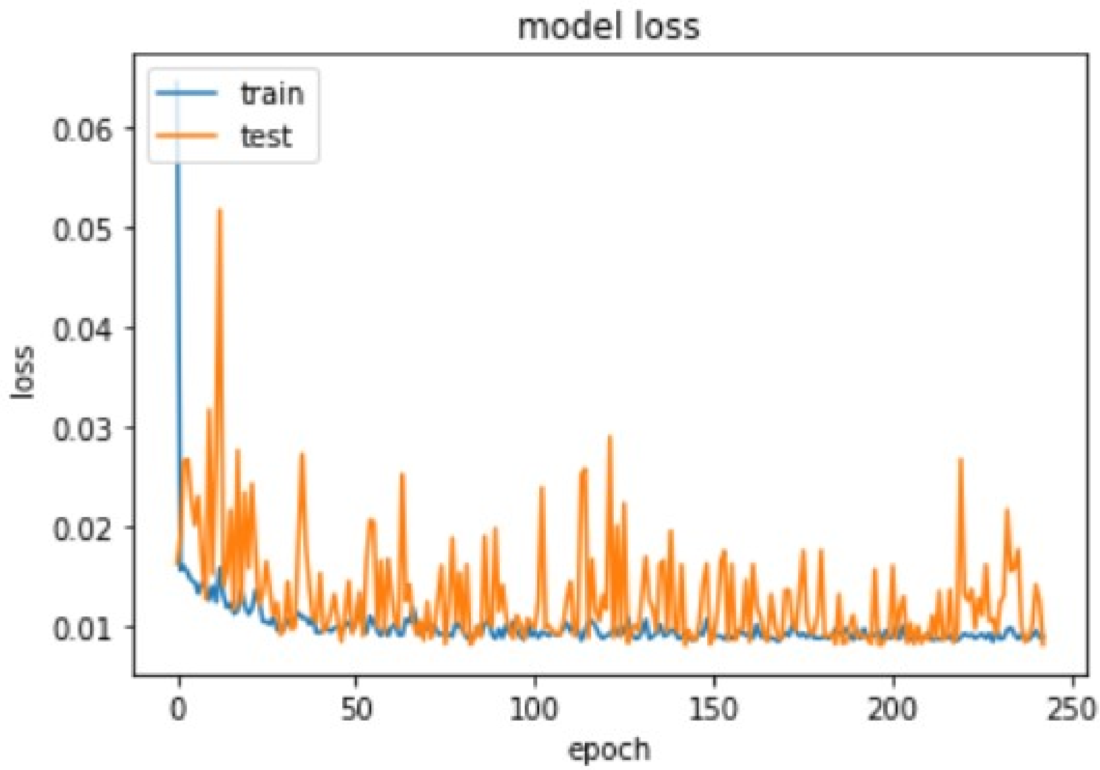

Figure 4.32 Model 9 (with non-averaged investor sentiment 1): Training vs Validation loss.

Figure 4.32 illustrates the plot indicating the loss while training Model 9.

Plot for prediction of true values vs predicted values,

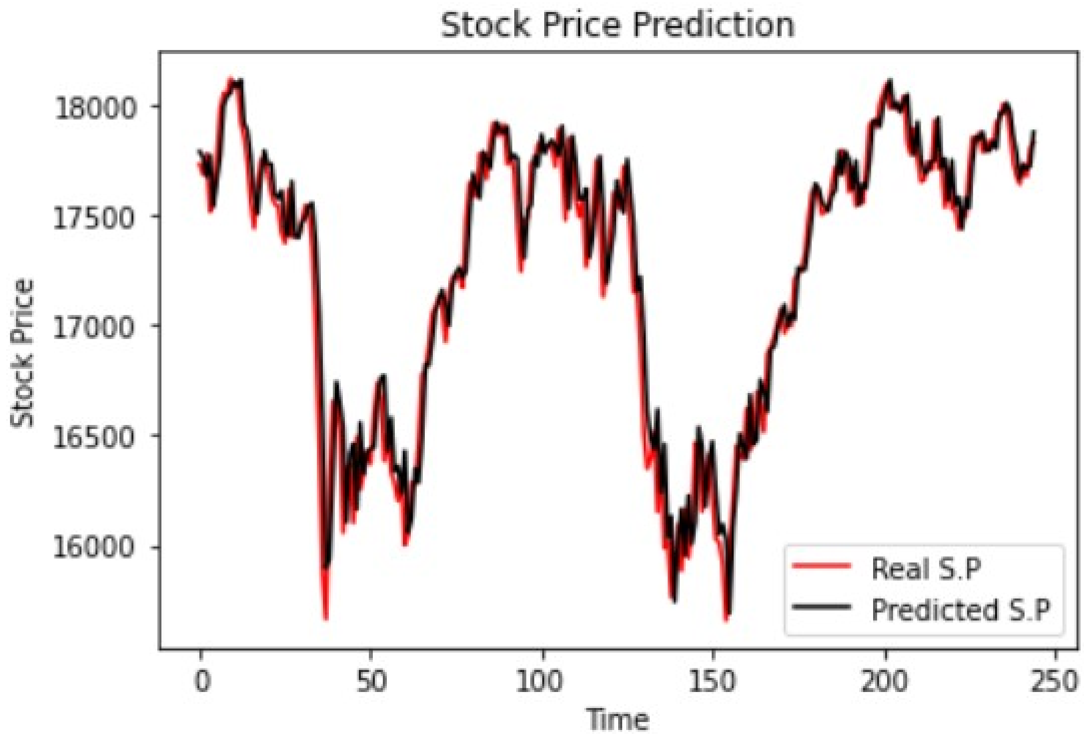

Figure 4.33 Model 9 (with non-averaged investor sentiment 1): Real vs Predicted Close Price.

Figure 4.33 illustrated the performance of the model as predicted close price and comparing it with real close price.

For stock market with non-averaged investor sentiment eq (2), we have used 7 features, 6 from stock market values and investor sentiment calculated from eq (2). Training model Architecture is,

```
Model: "sequential_14"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_6 (GRU)                  (None, 5, 300)            278100
_________________________________________________________________
gru_7 (GRU)                  (None, 100)               120600
_________________________________________________________________
dense_14 (Dense)             (None, 7)                 707
=================================================================
Total params: 399,407
Trainable params: 399,407
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.34 Model 10 (with non-averaged investor sentiment 2: GRU Architecture.

Figure 4.34 is the architecture of GRU Model with non-averaged investor sentiment 2 having 399,407 parameters having 2 GRU layers of 300 units and 100 units.

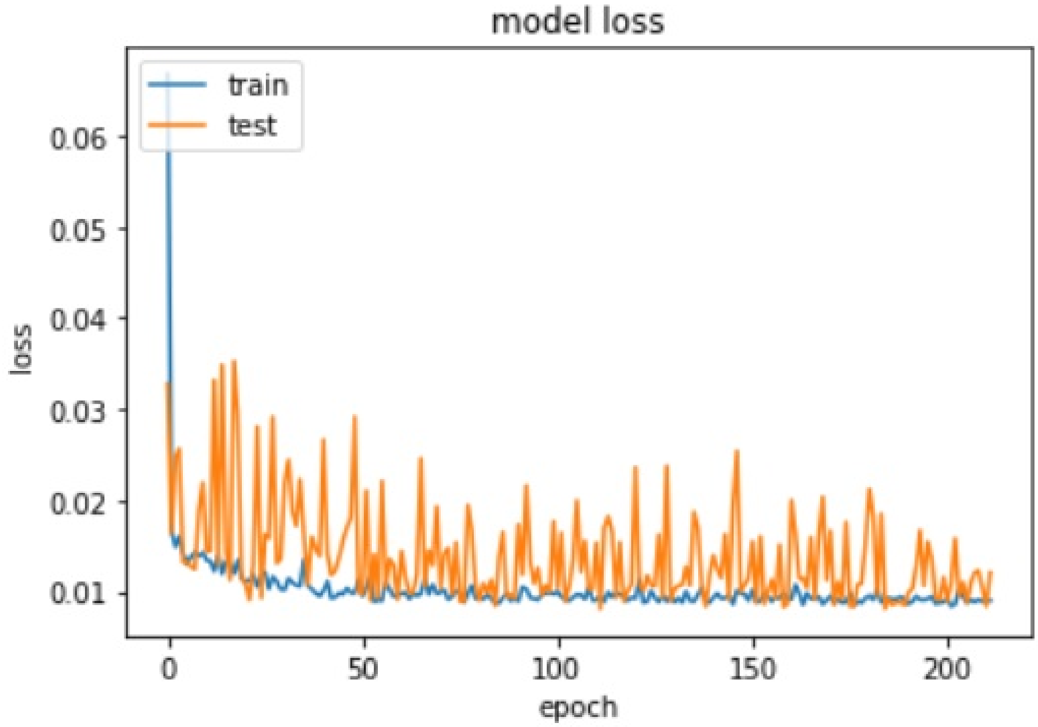

Figure 4.35 Model 10 (with non-averaged investor sentiment): Training vs Validation loss.

Figure 4.35 illustrates the plot indicating the loss while training Model 10.

Plot for prediction of true values vs predicted values,

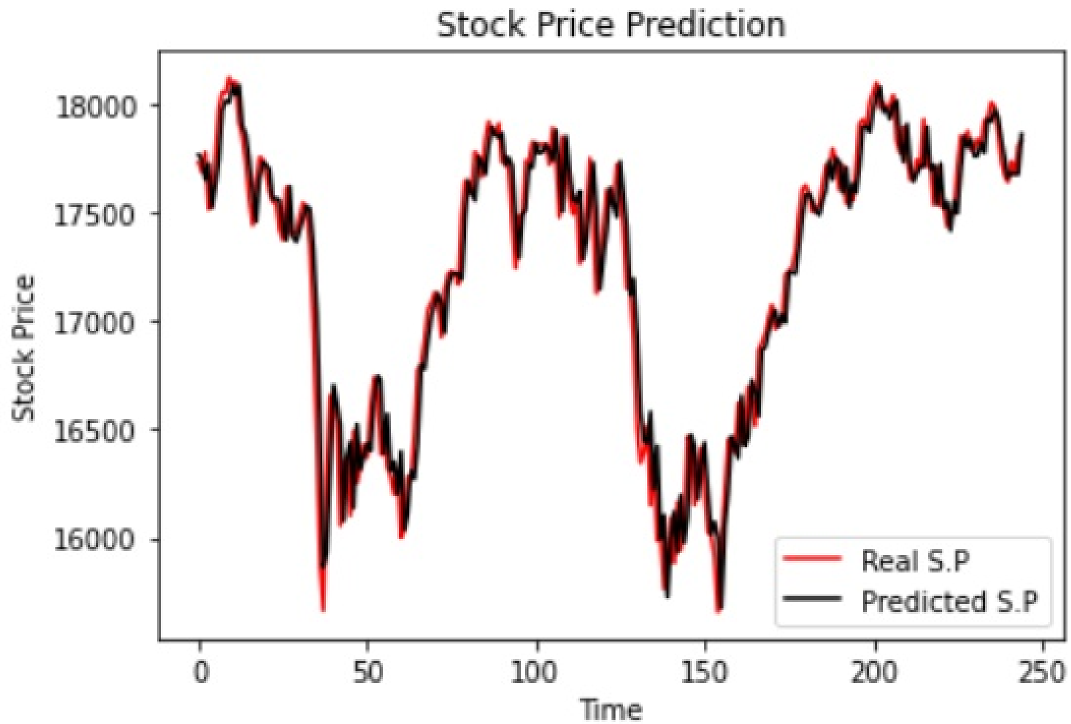

Figure 4.36 Model 10 (with non-averaged investor sentiment 2): Real vs Predicted Close Price.

Figure 4.36 illustrated the performance of the model as predicted close price and comparing it with real close price.

4.4.3 Overall Comparison

Performance of the trained models will be measured by mean absolute error, root mean squared error and mean absolute percentage error. Table 4.1 shows the error calculated by these metrics for evaluation of model's performance.

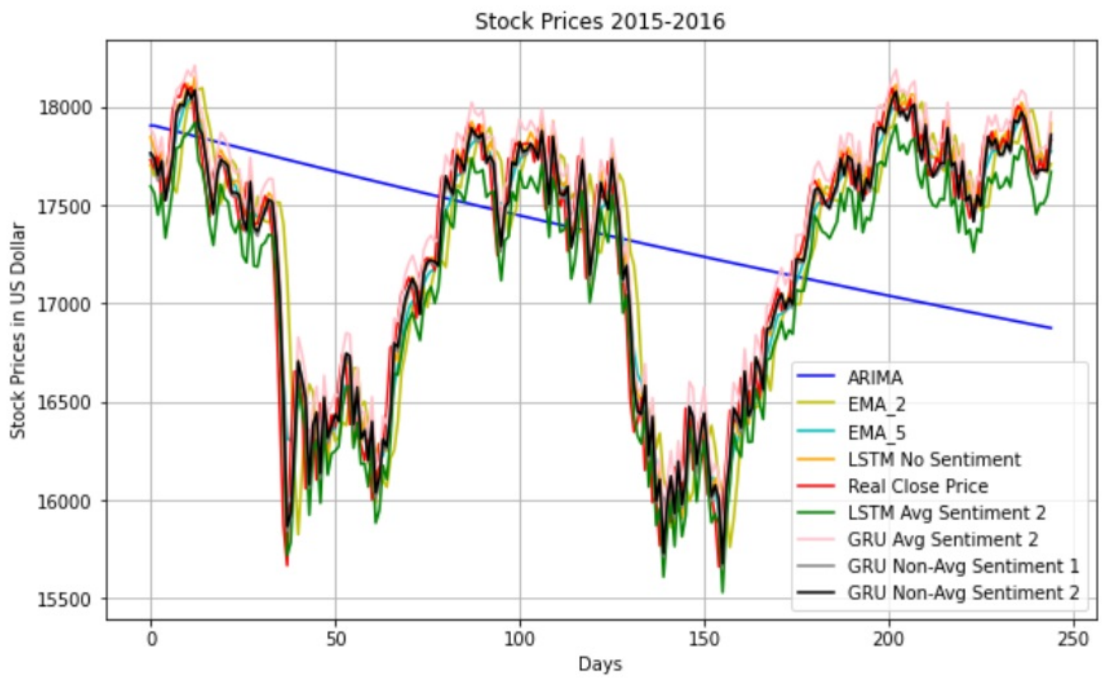

Figure 4.37 Performance: Real vs Predicted models

This plot indicates the performance of the various baseline and Deep Learning Models for comparison of there performance against real stock price.

Table 4.1 Performance of Deep Learning and Baseline Models

	Model	*MAE*	*RMSE*	*MAPE*
(i)	Moving Average(1)	128.62	170.93	4.174
(ii)	Exponential M.A. 2	135.82	181.58	0.79
(iii)	Exponential M.A. 3	146.06	195.78	0.85
(iv)	Exponential M.A. 5	166.697	224.4842	0.98
(v)	GARCH	8853.688	8857.941	51.35
(vi)	ARIMA	626.68	761.511	3.698
(v)	LSTM without Sentiment	169.85	223.37	0.999
(vi)	LSTM Averaged Sentiment 1	130.429	171.906	0.768
(vii)	LSTM Averaged Sentiment 2	131.567	176.223	0.768
(viii)	LSTM Non-Averaged Sentiment 1	130.590	172.098	0.769
(ix)	LSTM Non-Averaged Sentiment 2	130.27	173.132	0.768
(x)	GRU without Sentiment	148.62	193.87	0.874
(xi)	GRU Averaged Sentiment 1	131.11	172.36	0.759
(xii)	GRU Averaged Sentiment 2	127.833	170.696	0.7573
(xiii)	GRU Non-Averaged Sentiment 1	129.731	174.645	0.765
(xiv)	GRU Non-Averaged Sentiment 2	128.240	170.88	0.755

This table shows Mean Absolute Error, Root Mean Squared Error(RMSE) and Mean Absolute Percentage Error (MAPE(percent number)) of all the models trained in this study

Table 4.1 compares the performance of different baseline models as well as deep learning models, like LSTM and GRU with Real stock price. Comparing with baseline models, GRU performed 6% better in RMSE than E.M.A.(2). Performance for ARIMA is not good because of the long term dependency problem. As a result, it's prediction is a slope moving downwards. Performance for GRU with averaged investor sentiment 2 is better than various baselines and LSTM models. Averaged investor sentiment 2, which is averaged investor sentiment score for holidays and weekends does have an impact in predicting stock market. Sentiment 2, which refers to a particular sentiment expressed by number of traders, contributes to the prediction accuracy more than Sentiment 1 in both GRU and LSTM. Since GRU is less complex and it doesn't need memory unit, the model of GRU together with sentiment features outperforms that of LSTM.

CHAPTER 5

CONCLUSION

This thesis has been an interdisciplinary effort which explores stock market prediction. We started by giving an overview of stock market and exploring the most pertinent works in the field as of early 2020, which is important to note because of the pace at which this very new domain is expanding. It then touches upon various models and different sentiment that were the foundations of the novel work that was done here. In this study, we are exploring various recurrent neural networks to be used for training with Dow jones industrial average (DJIA) index. Here, we calculated investor sentiment from top 25 news headlines for each day and applied Vader sentiment to calculate positive, negative, and neutral scores. For calculating sentiment for each day, scores from each headline were averaged and then used to calculate investor sentiment for that given day. In averaged sentiment dataset, sentiment score for holidays and weekends were averaged to the next opening day. In non-averaged dataset, sentiment for weekends and holidays were ignored. So, the reason for averaging the sentiment values to next open day is the know the sentiment of investor before market opens after a weekend or holiday. Before applying the values for training, the dataset scaled using min-max scaler.

Recurrent neural networks which are considered most suitable for predicting time series forecasting data, were used for prediction of stock market. Different baseline models like EMA, arima. garch and LSTM were performed to set a baseline for prediction of recurrent neural networks. Gated Recurrent unit with averaged sentiment 2 performed 6% better than Exponential Moving Average in RMSE. In future work, a technical indicator which is more inclined to than investor sentiment can help us predict the stock market even closer, also with help of better recurrent neural networks the prediction to stock market can become as close to real values.

BIBLIOGRAPHY

[1] E. Alpaydin. Introduction to machine learning. *MIT press*, 2014.

[2] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo. Stock price prediction using the arima model. *16th International Conference on Computer Modelling and Simulation, Cambridge, UK*, 2014.

[3] Malcolm Baker and Jeffrey Wurgler. Investor sentiment and the cross-section of stock returns. *The Journal of Finance*, 61(4):1645–1680, 2006.

[4] Huina Mao Bollen, Johan and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[5] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

[6] Gregory W. Brown and Michael T. Cliff. Using neural networks for forecasting volatility of sp 500 index futures prices. *Journal of Empirical Finance*, 11(1):1–27, 2004.

[7] H. Chen. Customers as advisors: the role of social media in financial. *Management Science*, 54(3):477–491, 2012.

[8] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv:1406.1078*, 2014.

[9] J. Bradford De Long. Noise trader risk in financial markets. *Journal of Political Economy*, 98(4):703–738, 1990.

[10] Eugene F Fama. The behavior of stock-market prices. *The Journal of Business*, 38(1):34–105, 1965.

[11] Erik Cambria Frank Z. Xing and Roy E. Welsch. Natural language based financial forecasting: A survey. *Artificial Intelligence Review*, 50:49–73, 2018.

[12] Bolun Wang Divya Sambasivan Zengbin Zhang Haitao Zheng Gang Wang, Tianyi Wang and Ben Y Zhao. Crowds on wall street: Extracting value from collaborative investing platforms. *18th ACM Conference on Computer Supported Cooperative Work Social Computing*, pages 17–30, 2015.

[13] Qiyuan Gao. Stock market forecasting using recurrent neural network. *Ph.D. Dissertation. University of Missouri–Columbia.*, 2016.

[14] Shaikh A. Hamid and Zahid Iqbal. Using neural networks for forecasting volatility of s&p 500 index futures prices. *Journal of Business Research*, 57(10):1116–1125, 2004.

[15] S. Hansun. A new approach of moving average method in time series analysis. *Conference on New Media Studies (CoNMedia)*, 2013.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[17] Maxwell Stinchcombe Hornik, Kurt and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Netowrk*, 2(5):359–366, 1989.

[18] C.J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Eighth International Conference on Weblogs and Social Media*, 2014.

[19] R. Hyndman and Koehler A. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 2006.

[20] Yi Zhou Kai Chen and Fangyan Dai. A lstm-based method for stock returns prediction: A case study of china stock market. *Big Data (Big Data), 2015 IEEE International Conference*, pages 2823–2824, 2009.

[21] Ken-ichi Kamijo and Tetsuji Tanigawa. Stock price pattern recognition-a recurrent neural network approach. *IJCNN International Joint Conference on. IEEE, 1990*, 1990.

[22] Tze Leung Lai and Haipeng Xing. *Statistical models and methods for financial markets*. Springer, New York, 2008.

[23] Christine X. Jiang Lee, Wayne Y. and Daniel C. Indro. Stock market volatility, excess returns, and the role of investor sentiment. *Journal of banking Finance*, 26(12):2277–2299, 2002.

[24] Feng Li. Do stock market investors understand the risk sentiment of corporate annual reports? *SSRN Electronic Journal*, 2006.

[25] Ma W. Li, Y. International symposium on computational intelligence and design. *Applications of artificial neural networks in financial economics*, 1:211–214, 2010.

[26] B. G. Malkiel. The efficient market hypothesis and its critics. *The Journal of economic perspectives*, 17:59–82, 2003.

[27] Emadzadeh E. Muthaiyah S. Nikfarjam, A. The 2nd international conference on computer and automation engineering. *Text mining approaches for stock market prediction.*, 4:256–260, 2010.

[28] Adekoya A. F. Weyori B. A. Nti, I. K. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, pages 1–51, 2019.

[29] Ping-Feng Pai and Chih-Sheng Lin. A hybrid arima and support vector machines model in stock price forecasting. *Omega*, 33(6):497–505, 2005.

[30] Bo Qian and Khaled Rasheed. Stock market prediction with multiple classifiers. *Applied Intelligence*, 26(1):25–33, 2007.

[31] Trenta F. di Stallo A. L. Battiato S. Rundo, F. Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9:55–74, 2019.

[32] Robert P. Schumaker. Evaluating sentiment in financial news articles. *Decision Support Systems*, 53(3):458–464, 2012.

[33] Gudelek M. U. Ozbayoglu A. M. Sezer, O. B. Financial time series forecasting with deep learning: A systematic literature review: 2005-2019. *arXiv preprint arXiv:1911.13288*, 2019.

[34] Andrei Shleifer and Robert W. Vishny. The limits of arbitrage. *The Journal of Finance*, 52(1):35–55, 1997.

[35] Maytal Saar - Tsechansky Tetlock, Paul C. and Sofus Macskassy. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467, 2008.

[36] Verner R. Tkac, M. Artificial neural networks in business: Two decades of research. *Applied Soft Computing*, 38:788–804, 2010.

[37] Jung-Hua Wang and Jia-Yann Leu. Stock market trend prediction using arima-based neural networks. *IEEE International Conference*, 4:2160–2165, 1996.

[38] Yoshiteru Nakamori Wei Huang and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers Operations Research*, 32(10):2513–2522, 2005.

[39] Murray Z. Frank Werner Antweiler. Is all that talk just noise? the information content of internet stock message boards. *The Journal of Finance*, 59(3), 2004.

[40] Geoffrey E.; Rumelhart David E. Williams, Ronald J.; Hinton. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[41] Zehong Yang Xiaowei Lin and Yixu Song. Short-term stock price prediction based on echo state networks. *Expert systems with applications*, 36(3):7313–7317, 2009.

[42] Cambria E. Welsch R. E. Xing, F. Z. Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, 50:49–73, 2018.

[43] Eric P. Nichols Xiong, Ruoxuan and Yuan Shen. Deep learning stock volatility with google domestic trends. *arXiv:1512.04916*, 2015.