CZ4079 : Final Year Project

SCSE 19-0125 : AI Based Stock Market Trending Analysis

Student Name : Chan Wen Jun, Lawrann

Matriculation Number : U1721447K

Supervisors : Dr Wang Zhaoxia & Dr Li Fang

Examiner : Dr Loke Yuan Ren

A final year project report submitted to Nanyang Technological University in partial fulfilment of the requirements for the degree of Bachelor of Computer Engineering (Computer Science)

March 2020

Abstract

Forecasting stock market prices is a challenging task that many researchers seek to solve. With the rising trend of using machine learning models to predict stock market prices, creation of an enhanced stock price prediction model is vital in gaining an edge over the market. In this case, social media platforms such as Twitter presents a wealth of information that could facilitate stock market price prediction. Similarly, news publications have been shown to affect the movement trend of stock market prices.

Presented in this report is the exploration of sentiments, specifically sentiments from Twitter tweets and The New York Times news articles, in the creation of an enhanced Long Short-Term Memory (LSTM) stock price prediction model. Furthermore, this report compares the performance of a baseline LSTM prediction model without inclusion of sentiments against three sentiment enhanced LSTM prediction model.

The extraction of relevant New York Times news articles and Twitter tweets within this report is performed using New York Times API and a Twitter Scrapper. After which, sentiment classification for the extracted news articles is performed using vaderSentiment whereas sentiment classification for the extracted tweets is done using a state-of-the-art natural language processing model, Bidirectional Encoders Representing Transformers (BERT). Additionally, this project introduces a sentiment decay function that incorporates sentiment relevancy to the sentiment scores before input to a LSTM model.

Experimental results obtained from the comparison of the baseline LSTM model against the sentiment enhanced LSTM models have demonstrated improvement in price prediction performance for three LSTM models which incorporated sentiment scores. This report concludes that incorporation of sentiments extracted from Twitter tweets and New York Times news articles in the creation of a LSTM stock market price prediction model enhances the model's performance in stock price prediction.

Lastly, this project suggests further avenues for research through the exploration of employing a wider range of sentiments obtained from other news and social media platforms in the generation of a stock price prediction model.

Acknowledgements

During my two-semester stint working on this final year project, I would like to express my appreciation to everyone who have encouraged and given me advice, helping make this final year project possible.

I would like to extend a special gratitude to Dr Wang Zhaoxia from the Institute of High-Performance Computing for her guidance, support and valuable feedback when I was working on my final year project. Whenever faced with doubts or obstacles during the course of the project, she was ready to provide constructive feedback, specifying areas which I should look into further.

I would also like to extend my gratitude towards Dr Li Fang for her administrative assistance and providing all necessary resources for this final year project.

Furthermore, I would like to thank fellow final year project student Redmond Aldric Goon Yonghao for his sharing inputs with me.

Lastly, I would like to thank my examiner Dr Loke Yuan Ren, for grading this final year project.

Lawrann Chan

March 2020

Table of Contents

Abbreviations

AMPE	Absolute Mean Percentage Error
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
BERT	Bidirectional Encoders Representing Transformers
BP	Back Propagation
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit
LR	Linear Regression
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
MLP	Multi-Layer Perception
MSE	Mean Squared Error
NB	Naïve Bayes
NLP	Natural Language Processing
NN	Neural Network
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SVM	Support Vector Machine
SVR	Support Vector Regression

List of Tables

List of Figures

1. Introduction

1.1. Background

In the financial world, stock market traders aim to maximize profits and minimize risks in investments through the forecasting of the future price of a stock. Two popular literature used in forecasting stock market prices is Fundamental Analysis (FA) and Technical Analysis (TA). FA focuses on using macroeconomic factors to evaluate whether a stock price would rise or fall whereas TA focuses on the identification of trends and patterns within historical stock market data to predict stock price movements *[1]*. Stock market traders employ the use of both FA and TA by analyzing historical stock prices, financial reports, macroeconomic factors, news articles and sentiments from experts to gain insights into the future price of a stock. However, given the complex and volatile nature of stock markets, the predictability of stock market prices is still widely debated.

According to the principles of Efficient Market Hypothesis (EMH) and Random Walk Theory (RWT), stock market prices are unpredictable due to their dynamic and ever-changing characteristics *[1]*. EMH states that it is impossible to outperform the overall market as it is always efficient and RWT suggests that stock prices are unpredictable as they evolve in a random manner. Still, many researchers seek to tackle the task of predicting stock market prices.

In the present day, progress in the artificial intelligence field along with the growth of computing performance has allowed for the creation of multi-layered ANN. ANN is a machine learning technique modelled after human's biological neural network. It has enabled the creation of stock market prediction models which incorporates technical indicators such as historical stock prices to forecast future stock market values. ANN has seen wide use in the creation of stock market prediction models owing to its ability to model complex patterns and problems *[1], [2], [3]*.

1.2. Motivation

Besides the use of TA and FA in stock price forecasting, researchers have identified that sentiments obtained from social media texts and news articles influences stock market price movement *[1], [2], [3], [4]*. Quantifying the opinions of the general public with regard to a certain stock is possible through Sentiment Analysis (SA) of social media texts and news articles. SA, a subset of NLP tasks, is the classification of opinions within texts into positive, negative or neutral. These sentiment scores which represent the opinions of the general public could be leveraged with existing stock price prediction models to further improve stock price prediction.

In one study, comparisons were drawn between the stock price prediction capabilities of an ANN model that took into consideration news article sentiments, and an ANN model built without news article sentiments. The results showed that the former model performed better, as it achieved a lower MSE of 3.57E-05 *[2]*. In another study, an ANN stock trend prediction model build using sentiments obtained from social media texts along with historical stock price achieved portfolio returns of 19.54% within a seven-months period *[3]*. However, utilizing a wider range of sentiments through both social media platforms and news articles in creating a stock price prediction model has been neglected in this area of study.

Moreover, selecting the optimal tool for SA is vital to ensure accurate classification of opinions within texts. A recent state-of-the-art NLP tool, Bidirectional Encoders Representing Transformers (BERT), developed by Google attained exceptional results on eleven NLP tasks and has paved the way in providing precise sentiment classification of texts *[4], [5]*. Hence, using of BERT to perform sentiment classification of extracted texts could allow for more meaningful outcomes.

Lastly, a variant of the ANN, LSTM, is well-suited for time series prediction problems due to its ability to account for past inputs. Further evaluation of its use in stock price prediction should be explored.

1.3. Objective

With the growing trend of using machine learning models to forecast future stock market prices, enhancing a stock price prediction model's accuracy is essential in minimizing risk and generating greater profit for the user. Therefore, the objective of this project is to study and explore the applicability of sentiments acquired through sentiments analysis of Twitter tweets and New York Times news articles in the creation of an enhanced LSTM stock market price prediction model.

This will be done through the implementation and comparison of four LSTM models to predict the next day's closing price.

The four models are:

1. Baseline LSTM model – Features: Closing price
2. Tweet sentiment LSTM model – Features: Closing price, Tweet sentiments
3. Article sentiment LSTM model – Features: Closing price, Article sentiments
4. Tweet and Article sentiment LSTM model – Features: Closing price, Tweet sentiments, Article sentiments.

Results from the baseline model will be used as a point of comparison with the other three LSTM models.

1.4. Project Scope and Organization

The scope of this project:

- Extraction of historical price dataset from Yahoo! Finance
- Extraction of Twitter tweets through web scrapping
- Extraction of New York Times news articles using New York Times API
- Preprocessing of tweets and news articles dataset
- Fine-tuning of BERT model for sentiment classification of tweets
- Performing sentiment classification on preprocessed Twitters tweets using fine-tuned BERT model
- Performing sentiment classification on New York Times News Articles using vaderSentiment
- Implementation of LSTM stock price prediction models
- Evaluation of LSTM stock price prediction models

The remaining part of this report is organized as follows:

- Section 2 presents the literature reviews of existing stock market prediction approaches
- Section 3 states the required datasets and specifications
- Section 4 details the final year project schedule
- Section 5 discuss in detail the data extraction, preprocessing along with proposed implementation methodology
- Section 6 presents the experimental results
- Section 7 discusses the result obtained from the experiment
- Section 8 presents the conclusions and possible future work

2. Literature Review

In this section, research articles relating to the use of NN in the creation of stock price prediction models will be introduced. Additionally, works related to the use of social media sentiments and news article sentiments in stock market prediction will be explored. Lastly, Google's BERT model for sentiment classification will be reviewed.

2.1. Neural Network and Stock Market Prediction

Many machine learning algorithms have been used to create stock trend or price prediction models. However, most researchers have relied on the ANN architecture in building a prediction model *[2], [3], [6], [7], [8].* ANN have been the machine learning algorithm of choice due to the backpropagation, learning phase where weights in the network are adjusted iteratively to improve the model's prediction. This iterative update of weights has given ANN the capabilities of learning complex functions and has made it the go-to technique for addressing many regression and classification problems. Figure 2-1 depicts the architecture of a multi-layered ANN.

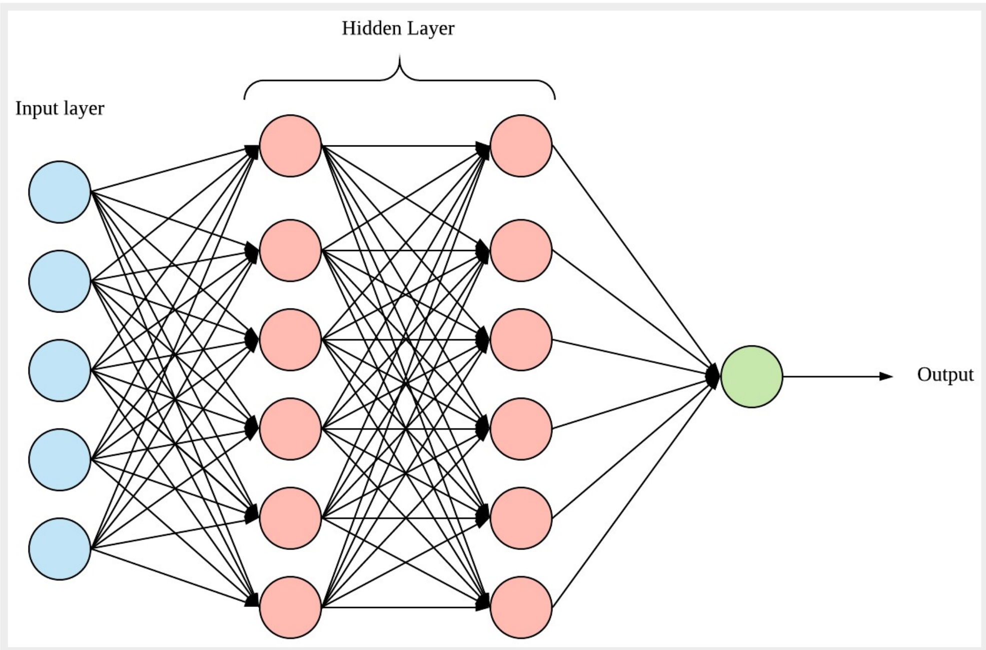


Figure 2-1 Graphical Representation of a Multi-Layered Artificial Neural Network Architecture

A research done by Kara et al. in 2011 comparing the performance of SVM and ANN in the prediction of stock price movement for the Istanbul Stock Exchange has shown that the ANN model performed significantly better achieving a 75.74% accuracy in price trend prediction, as compared to the SVM model accuracy of 71.52% *[6]*.

In *[7]*, Rajesh et al explored the creation of an ANN stock price prediction model which incorporated tweets containing stock symbols for Johnson and Johnson (JNJ) and Apple (AAPL) to predict stock market prices. Similarly, in another paper *[8]*, Sayayong et al explored the use of CNN for stock price prediction of the Thai stock market. Both papers found good results using NN architecture in the creation stock market price prediction models. In the recent few years, many researchers have explored the use of RNN to solve time-series prediction problem.

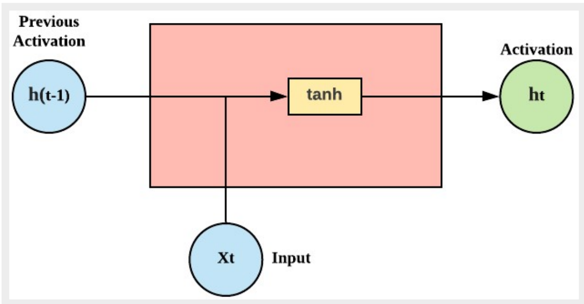


Figure 2-2 Graphical Representation of Recurrent Neural Network Cell

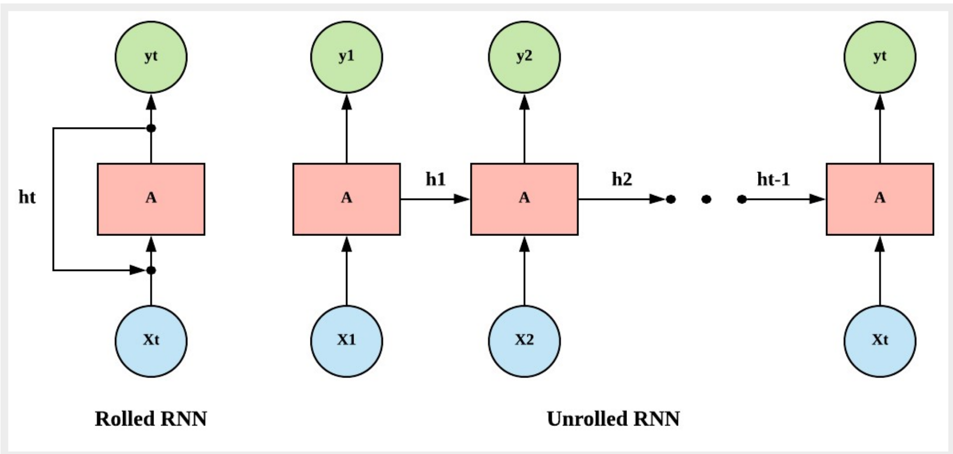


Figure 2-3 Graphical Representation of Recurrent Neural Network

The RNN cell is shown in Figure 2-2 and the RNN architecture is shown in Figure 2-3. RNN, a variant from the ANN, has a recurring connection to itself. This recurring connection allows the RNN to take into consideration inputs at previous time steps to predict the current output. RNN ability to factor inputs from previous timestamps allows it to model time-series prediction problem better than the conventional ANN. However, the RNN, suffers from the issue of vanishing gradient problem where the gradient of the loss function approaches zero due during backpropagation and updating of weights.

In 1997, Hochreiter and Schmidhuber proposed a new RNN called the LSTM *[9]*. The LSTM is an improved structure from the RNN which allows the handling of long sequences and eliminates to the vanishing gradient problem faced in the RNN structure.

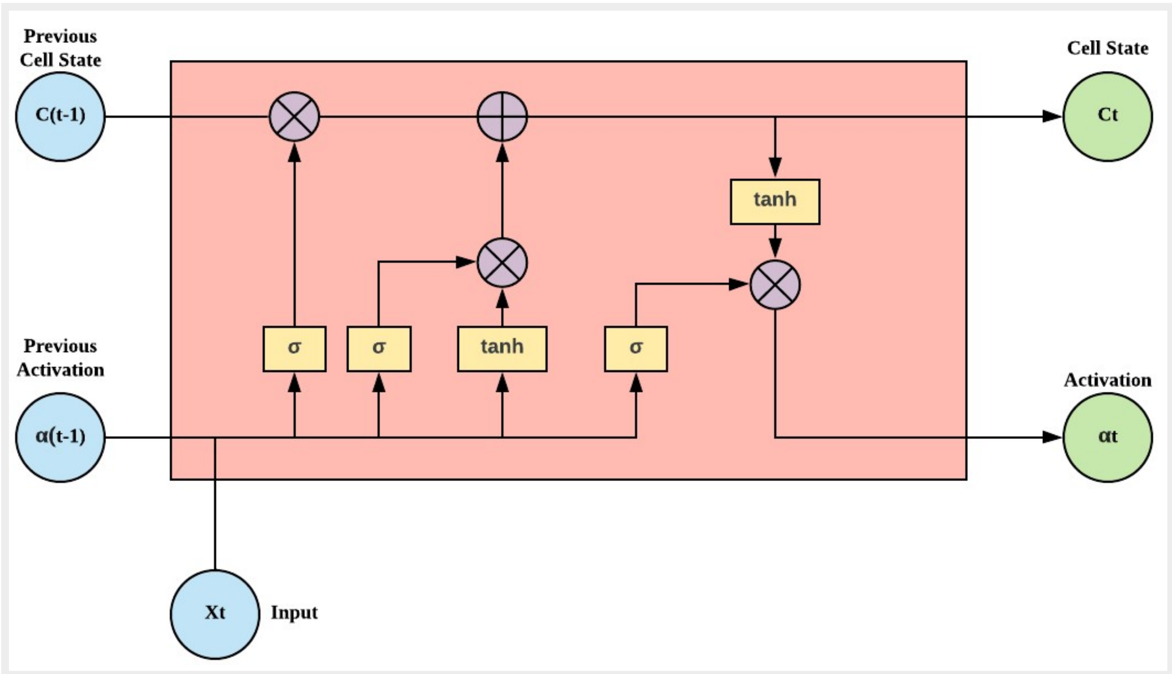


Figure 2-4 Graphical Representation of LSTM cell

Figure 2-4 shows a graphical representation of the LSTM cell. The difference between a RNN cell and LSTM cell is the addition of the input gate, forget gate, output gate and a cell state represented by Sigmoid. The cell state carries information from earlier time steps and additional information gets removed or added onto the cell states via the forget and input gates. This provides LSTM the advantage of preserving gradients over long inputs as compared to the RNN. Additionally, like RNN, LSTM performs well in time-series prediction problems.

In *[10],* Gao et al. explored and demonstrated the use of LSTM along with historical stock market trading data to predict stock market prices. They made use of a 20-day window size to predict the closing price of next day. Evaluating the prediction results from the LSTM against three other models, Moving Average, Estimated Moving Average and SVM. The findings showed that the LSTM model was the most accurate model in price prediction achieving the lowest RMSE, MAE, MAPE, AMPE score as compared to the other three models.

Similarly, in *[11]*, Qian et al. compared the prediction accuracy of three different stock with differing stability using a LSTM model against an ARIMA model. The results from the paper showed that LSTM model had a 66.78% lower error rate as compared to the ARIMA model in price prediction.

Wang et al. *[12]* did a study on comparing the accuracy of using a BP NN against LSTM to predict the trend, high, closing and low price of soybean futures. The LSTM model was able to achieve higher trend prediction accuracy, up to 70% in the high and low trends, as compared to the BP NN model.

In *[13]*, Althealaya et al. did a study to compare the performance of bidirectional LSTM, stack LSTM and MLP ANN in stock market prediction. The results from the study show that both LSTM models attained a lower RMSE as compared to the MLP ANN model in stock price prediction.

A study *[14],* was explored by Zhao et al. on using a time-weighted LSTM model to predict the up and down trends of a stock market. This study focuses on exploring the nature of time in a time series data by exploiting the importance of data at different time points. The paper explored the addition of time-relevant weight functions for different time steps within an input sequence and discovered that closer data have more influence on prediction accuracy. Additionally, the results of the study showed that the time-weighted LSTM model outperformed the SVM, RNN, RF and AB model in trend prediction, achieving an accuracy of 83.91%.

As seen from above articles, many researchers have explored using ANN to model stock market prices. In the recent years, the use of LSTM architecture has been widely researched for modelling the time-series of the stock market. It has shown to perform the best as compared to other ANN and machine learning models *[10], [11], [12], [13] [14]*.

2.2. Social Media Sentiments and Stock Market Prediction

Twitter contains a great amount of information, with an average of 140 million tweets every day. It provides us with a reflection of the publics opinions and have been used to solve prediction problems such as box office movies returns *[15]*. Twitter has been an area of interest for many researchers in gathering public sentiments and sentiment obtained from tweets have been used to identify correlations with stock market prices and trends.

In *[16]*, Aich et al. investigated stock price movements using event related Twitter feeds. The paper identified a correlation between the sentiment score of event related tweets to the price of the stock. A correlation was identified when Samsung released its Galaxy Note 7 and abruptly stopped its production due to a manufacturing defect. This caused the stock price of Samsung to rise initially and fall abruptly after. This rise and fall in stock price were similar observed in a rise and fall of sentiment score of event related tweets.

In *[17]*, a paper exploring the use of twitter post to predict stock market behavior by Li et al. studied the relevance of twitter user's mood (happy, sad, anger, fear, disgust, surprise) with relation to stock price trends. The results of the paper revealed that certain emotions such as happy, sad, and anger have a relatively strong correlation to stock prices, especially vocabularies relating to sad emotions whereas certain moods such as fear and disgust have a low correlation with stock market prices.

Pagolu et al. explored the use of sentiment analysis on twitter data for prediction of stock market trends in *[18]*. The researchers collected stock opening and closing prices along with 2.5 million tweets on Microsoft from 31st August 2015 to 25th August 2016. Only tweets with words $MSFT, #Microsoft, #Windows were gathered to ensure the tweets represented the public opinion and news relating to Microsoft. Historical stock prices were collected from Yahoo! Finance and missing dates within the stock prices arising from closure during public holidays or weekends were approximated using a concave function. Acquired tweets were preprocessed in three steps, tokenization, stop word removal and regex matching. After preprocessing the tweets, manual annotation of 3,216 tweets into positive, negative and neutral was performed. A machine learning model was then trained on the manually annotated tweet performed sentiment classification for the remaining tweets. Using a 3-day aggregate sentiment values along with up or down stock trend as features to train a classifier, results showed an accuracy of 69.01% when using logistic regression and 71.82% when modelled using SVM to predict the stock market trends. The paper pointed out a strong correlation between the movement of stock prices with regards to the public sentiments in tweets.

China has a unique social media landscape in terms of social media platforms. Sina Weibo is the largest and most widely used social media site within China and is akin to the microblogging site Twitter. Correspondingly, there have been research performed on social media sites available in China to predict the stock market prices.

In *[3]*, Sun et al. investigated the relationship between stock price movement and social media sentiments in China. Social media sites in china include Weibo, Stockbar, Snowball ad Circle. Weibo is a microblogging site similar to Twitter. Stockbar and snowball are online forums where users can discuss about stocks. Lastly, Circle is an online chat room for traders to discuss about a specific company's stocks. The research studied how the stock market is discussed within these different social media platforms and how social media activity correlates to the stock trade hours and price changes. The study focused on Circle which shown the highest level of activity and relevancy to the stock market operating times as compared to the other three social media sites. Sentiment analysis is performed through manual annotation of 18,000 randomly chosen post on Circle into positive, negative or neutral. After which an ensemble learning model consisting of LR, SVM, NB SVM and LSTM models is trained used to perform sentiment analysis on the remaining posts. This research identified a correlation between positive and negative post sentiments with stock prices using Granger causality test. A stock trend prediction model is created using the posts sentiments achieving an average accuracy of 57.3%. Additionally, a back test was performed using the model, achieving a promising portfolio return of 19.54%,

In *[19]*, Chen et al. studied the effect of social media posts from famous people and organization on stock market price prediction. Contents posted on official accounts of famous person or organization on Weibo were extracted and used to analyze the public moods. Data preprocessing was performed on the collected contents from these official accounts to remove irrelevant contents not relating to financial news. Stock prices such as the opening, closing, high and low were extracted from the stock market on each training day. After which a two-layer RNN with GRU prediction model was then built using those collected features. The RNN-GRU model achieved a RMSE of 0.803 as compared with SVR and LR model of which had RMSE of 3.28 and 2.43. This paper identified a correlation between posts by influential people on Weibo to stock market prices and showed that an accurate stock price prediction model can be built using social media sentiments.

In *[20]*, Wang et al. studied the relation between sentiment of comments on Weibo to stock prices. Comments about stocks on Weibo were collected after each trading day closes. It is then preprocessed using Chinese segmentation tool, Institute of Computing Technology, Chinese Lexical Analysis System (ICTCLAS), to remove redundant data. After which How Net+, a sentiment dictionary, is used to label each word with a positive or negative value. The sentiment index is then calculated for each comment. Using past historical trading data on small-cap stocks, two SVM models were built, one which includes the sentiment index of comments from Weibo and the other without. The SVM model with sentiment showed a Pearson correlation coefficient of 0.79 whereas the SVM model without sentiment had a Pearson correlation coefficient of 0.56. This showed that the sentiments mined from Weibo messages is influential to stock market prices.

The above research articles have highlighted that public opinions mined from social media sites such as Twitter and Weibo correlates and sways the stock market prices. Sentiment mining and analysis of microblogging sites is imperative to building an enhanced stock price prediction model.

2.3. News Articles Sentiments and Stock Market Prediction

Financial news has a profound effect in stock market price movement. In addition to using sentiments mined of social media sites, sentiments obtained from news articles have been considered by many researchers in the creation of stock market prediction models. Research have shown that news articles sentiment does correlate to stock price movements.

In *[21]*, Alostad et al. explored the prediction of hourly stock market directional trend by using news articles collected from NASDAQ websites. The paper made employed the use of SentiStrength along with Loughran and McDonald financial statement dictionaries for sentiment classification of news articles. Using a LR classifier to create a stock market trend prediction model, the classifier achieved a high directional accuracy of 82.9% predicting AT&T stock trend movement . Results from the paper also identified that breaking news yielded a significant boost in stock trend prediction as compared to when all available news is used in the prediction.

In another research, *[2]*, Wang et al. studied the relationship between news sentiments and stock prices. 10 years of data were taken from the New York Times news articles and Yahoo Finance stock market data from 1st January 2007 to 31st December 2016. The news articles were preprocessed to only contain sections from the Business, National, World, U.S, Politics, Opinion, Tech, Science, Health and Foreign section. After which three types of sentiment analysis was performed on the news articles followed by a voting method to decide the final sentiment score of the news article. Four training methods were experimented in the creation of the NN. Through the comparison of two scenarios to test the effects of news sentiments in predicting stock market price, the first scenario using only the stock market price time series and the second scenario which considers the news article sentiment along with the stock market price time series. The study showed that consideration of news sentiments improved the performance of the NN model.

In addition to utilizing of sentiments from microblogging social media sites to enhance a stock price prediction model, we can deduce from the above articles that sentiments of news articles play a major role in influencing stock market prices as well.

2.4. Bidirectional Encoders Representing Transformers

Given that sentiments from news articles and social media plays a major part in influencing stock market prices, accurate sentiment classification is pivotal to creating an enhance stock price prediction model. Inaccurate sentiment classification could have adverse effects when training a stock price prediction model. BERT is a revolutionary NLP tool developed by Google. It is pre-trained on a massive unlabeled text corpus and only requires further fine-tuning to create state-of-the art models for specific NLP tasks without needing to modify of the model's architecture *[5]*. In *[4]*, Sousa et al studied using BERT for sentiment classification of news articles and sought to identify a correlation between sentiment scores with the movement of the Dow Jones Industrial Index. The research experimented with BERT, NB, SVM and Text CNN in sentiment classification and showed that BERT model outperformed the other models with an accuracy of 82.5%. Using sentiments from economic news to identify stock trend movement, the research achieved a 69% accuracy in predicting future stock trends.

3. Material Resources

In this section, the resources used within this project will be listed.

3.1. Requirement Specifications

This project was implemented using Python programming language, version 3.7.3 and Google Collaboratory.

The following python libraries were utilized within this project

- TensorFlow Keras – For implementation of LSTM models
- Pandas – For data manipulation of tables
- Numpy – For manipulation and handling of multi-dimensional arrays and matrices
- Sklearn (Scikit-learn) – For various machine learning algorithms (e.g normalization, scaling, preprocessing, data split) used in the creation of a LSTM architecture
- Bert-Tensorflow – For the fine-tuning and sentiment classification of Tweets using google BERT uncased model
- VaderSentiment – For sentiment classification of News Articles
- Matplotlib – For plotting of graphs
- Twitterscraper – For the scrapping of Twitter tweets
- Json – For the storing of scrapped articles and tweets
- Csv – For the storing and visualization of scrapped articles, tweets and historical stock data
- Tqdm – For displaying progress bar timing of performance and estimated time to completion

3.2. Datasets

This project made use of a five-year dataset for the training and evaluation of the stock price prediction models. The datasets consist of data ranging from the date 1st of January 2014 to 31st December 2018. Additionally, this project experimented with three different stocks with differing daily trade volumes for the creation of the stock price prediction model.

The three selected stocks are:

1. SPDR S&P 500 ETF Trust ($SPY)
2. Citigroup Inc ($C)
3. Activision Blizzard, Inc ($ATVI)

Of the three selected stocks, $SPY has highest daily trade volume with average for 100 million trades a day. $C averages of 20 million trades a day and $ATVI averages 7.5 million trades a day.

For each of the selected stocks, the following datasets are required:

1. Stock's historical price dataset
2. Stock related Twitter tweets
3. Stock related news articles from New York Times

4. Project Schedule

This section presents the schedule of this project broken down into different timelines.

Table 4-1 Project schedule and description

S/N	Date	Description
1	12th Aug 2019 to 18th Aug 2019	• New York Times API exploration • Twitter API exploration
2	18th Aug 2019 to 1st Sep 2019	• Extraction of news articles using New York Times API • Extraction of tweets using a twitter scrapper
3	1st Sep 2019 to 8th Sep 2019	• Exploration of data preprocessing techniques • Exploration of sentiment classification techniques • Exploration of LSTM models for time-series prediction
4	8th Sep 2019 to 14th Sep 2019	• Performing data preprocessing on extracted news articles and tweets
5	14th Sep 2019 to 8th Nov 2019	• Exploration of BERT model • Perform fine-tuning of BERT model • Sentiment classification of tweets using BERT model
6	8th Nov 2019 – 15th Nov 2019	• Sentiment classification of news articles using vaderSentiment
7	15th Nov 2019 – 30th Nov 2019	• Refining preprocessing and sentiment classification
8	1st Dec 2019 - 1st Feb 2020	• Implementation and evaluation of LSTM models
9	1st Feb 2020 - 23rd Mar 2020	• Work on Final Year Report

5. Proposed Implementation

This section will explain in detail the extraction, preprocessing and sentiment classification of the collected dataset followed by the architecture of the LSTM model used to create our stock price prediction model.

5.1. Dataset Extraction

Extraction of dataset is not a menial task, if the dataset extracted is inaccurate or does not consists of all the relevant and required data, it would affect the final LSTM model generated. Extraction of the stock historical price, Twitter tweets and New York Times news article will be explained in this section.

5.1.1. Historical Price dataset

Historical price dataset for the three selected stocks were downloaded from Yahoo! Finance website. The dataset downloaded consist of historical price data from 1st of January 2014 to 31st December 2018.

Table 5-1 Meta data of Historical Stock Price

Field Name	Description
Date	Date which the stock is traded
Open	Price which the stock trades upon opening of the stock exchange
High	Highest price point of the stock during the day
Low	Lowest price point of the stock during the day
Close	Price which the stock is last traded during the closing of the stock exchange
Volume	Total quantity of trades for the stocks
Adj Close	Closing price after adjustments for splits and dividend distributions

Table 5-1 shows the meta data provided when the stock historical prices are downloaded from Yahoo Finance website.

Date	Open	High	Low	Close	Adj Close	Volume
1/2/2014	183.98	184.07	182.48	182.92	162.584	119636900
1/3/2014	183.23	183.6	182.63	182.89	162.557	81390600
1/6/2014	183.49	183.56	182.08	182.36	162.086	108028200
1/7/2014	183.09	183.79	182.95	183.48	163.082	86144200
1/8/2014	183.45	183.83	182.89	183.52	163.117	96582300
1/9/2014	184.11	184.13	182.8	183.64	163.224	90683400
1/10/2014	183.95	184.22	183.01	184.14	163.668	102026400
1/13/2014	183.67	184.18	181.34	181.69	161.491	149892000
1/14/2014	182.29	183.77	181.95	183.67	163.251	105016100
1/15/2014	184.1	184.94	183.71	184.66	164.131	98525800
1/16/2014	184.28	184.66	183.83	184.42	163.917	72290600
1/17/2014	184.1	184.45	183.32	183.64	163.224	107848700
1/21/2014	184.7	184.77	183.05	184.18	163.704	88621200
1/22/2014	184.49	184.57	183.91	184.3	163.811	61270900
1/23/2014	183.37	183.4	181.82	182.79	162.468	132496900
1/24/2014	181.6	181.66	178.83	178.89	159.002	208677100
1/27/2014	179.06	179.52	177.12	178.01	158.22	180843100
1/28/2014	178.14	179.3	178.12	179.07	159.162	110463200
1/29/2014	177.58	178.55	176.88	177.35	157.633	216597300
1/30/2014	178.83	179.81	178.26	179.23	159.304	118938100

Figure 5-1 Snippet of historical price dataset from Yahoo! Finance

Figure 5-1 is a snippet of the extracted stock historical price dataset. As seen from the Date column, the dates are not continuous as certain days such as weekend or New Year's Day stock trading operations are closed.

5.1.2. Twitter Tweets dataset

Twitter Application Program Interface (API) was initially explored for the extraction of Twitter tweets. Twitter API had a rate limit of 300 request every 15 minutes and given the large number of tweets required, using Twitter API to extract tweets was not possible. Additionally, gathering tweets using Twitter streaming API was also not feasible as Twitter limits the number of streamed tweets available to the user. Moreover, due to the course of this FYP ranging two semesters, collection of Twitter dataset using the streaming API would not be effective. To overcome this issue, I made use of a Massachusetts Institute of Technology (MIT) licensed Twitter Scrapper developed by Taspinar *[22]*. As the repository was not regularly updated, I made some changes to code and automated the scrapping of tweets from the 1st January 2014 to 31st December 2018. A speedier extraction rate of 25,000 tweets per hour was achieved using the Twitter Scraper.

Table 5-2 Search query and number of tweets extracted for respective stocks

Stock Name	Query Parameters	Number of Tweets Extracted
SPDR S&P 500 ETF Trust	S&P 500 S&P500 #S&P500 $SPY SPDR S&P 500 Trust ETF	2,972,169
Citigroup Inc	#Citi NYSE:C Citigroup $C	793,382
Activision Blizzard, Inc	Activision Blizzard Blizzard Inc $ATVI #ATVI	183,872

For the purpose of ensuring only tweets related to the selected company were extracted, different search parameters were used for the three selected stocks when scrapping for the Twitter dataset. Due to the large number of tweets extracted, scrapped tweets were stored into json format based on the month and year of the tweet. Table 5-2 indicates the different search query parameter used to extract the tweets and the number of tweets obtained for each stock from the 1st January 2014 to 31st December 2018. A total of 3,949,423 tweets were scrapped for all three selected stocks.

Table 5-3 Meta data of the extracted tweets

Field Name	Description
Hashtags	The hashtags present within the tweet
Is_replied	If tweet has a reply
Is_reply_to	If tweet is a reply to another tweet
Screen_name	Display name of the user
Links	Links present within the tweet
Parent_tweet_id	ID of the parent tweet
Likes	Number of likes for the tweet
Replies	Number of replies to the tweet
Reply_to_users	Username of which the tweet replied to
Retweets	Number of retweets
Text	Content of the tweet
Text_html	Html of the text within the tweet
Timestamp	Time and date which the tweet was posted
Timestamp_epochs	The time and date the tweet was posted in epochs
Tweet_id	The unique ID of the tweet
Tweet_url	The hyperlink parameter to the tweet
User_id	Unique ID of the user
Username	Login name of the user

As seen from Table 5-3, there are many unrequired fields within the extracted tweets. Removal of unrequired fields will be detailed in the following section of this report.

5.1.3. News Articles dataset

For the news article dataset, New York Times API along was used for extracting articles related to the selected stocks from 1st January 2014 to 31st December 2018. New York Times API had a rate limit of 10 request per minute and 4000 requests per day. Extraction of news articles using the API was achieved by implementing a six second sleep before every request. Due to the lesser number of articles as compared to tweets, the extraction of all relevant articles was completed within a day. In contrast to the method of gathering relevant Twitter tweets which made use of specific stock related terms, I used a general search term when gathering relevant New York Times news articles.

Table 5-4 Search query and number of articles extracted for respective stocks

Stock Name	Query Parameter	Number of Articles Extracted
SPDR S&P 500 ETF Trust	S&P	3,865
Citigroup Inc	Citi	24,515
Activision Blizzard, Inc	Activision	100

As seen from Table 5-4 there were considerably more articles extracted for Citigroup Inc as compared to SPDR S&P 500 ETF Trust and Activision Inc. This was due to general search term used for extracting articles. As the search term used for extraction of news articles was generic, other news articles not related to Citigroup such as articles related to Citi field which is a baseball park in New York City, and Citi bike, a bike sharing system, were extracted as well.

The reason why a general search term was used was to ensure every article related to the selected company was extracted. In the preprocessing stage, irrelevant news articles within the dataset will be identified and filtered away. A total of 28,480 news articles were extracted for the three selected stocks.

Table 5-5 Meta data of the extracted articles

Field Name	Description
Id	Unique ID of the article
Abstract	Abstract paragraph of the article
Headline	The headline of the article
Desk	The desk which the article is from (e.g Editorial, Foreign)
Section	The section which the article is from
Snippet	The snippet paragraph of the article
Lead_paragraph	The lead paragraph of the article
Source	The source which the article was from
Type	The type of article (e.g News, Letter)
Url	Url to the news article
Locations	Location which this news occurred
Word_count	Number of words within the article
Subjects	Key subjects of the article

As seen from Table 5-5, there are many unrequired fields within the extracted articles. The following sections will detail the removal of these unrequired fields.

5.2. Preprocessing Datasets

Before the creation of the LSTM stock price prediction models, proper preprocessing and sentiment classification of the dataset collected must be done. Thorough and accurate data preprocessing is crucial to the model's performance as any irrelevant tweets or news article which are not filtered out could alter the overall sentiment classification for the trading day and affect the stock price prediction model created.

5.2.1. Preprocessing of Tweets

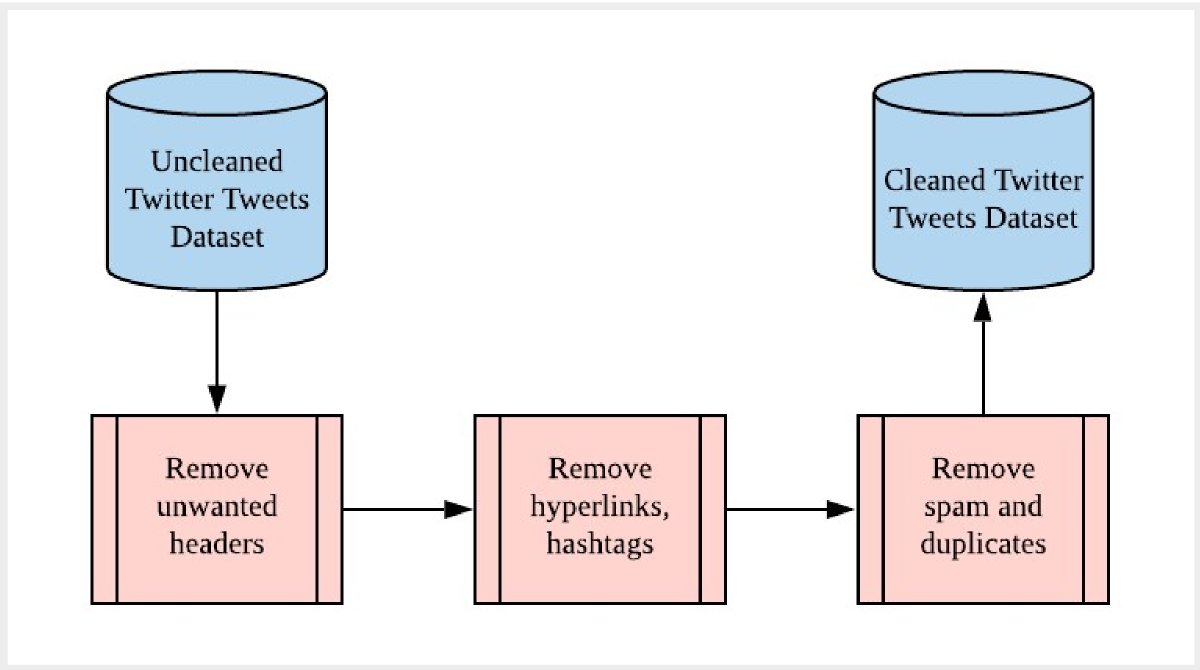


Figure 5-2 Flowchart for Preprocessing of Twitter Tweets

Figure 5-2 shows the flowchart for how Twitter tweets were preprocessed. The preprocessing of collected Twitter tweets consists of three steps performed by three separate python scripts.

1. Removing unwanted headers
2. Removing hyperlinks and hashtags
3. Identifying and removing duplicated tweets

As seen earlier in Table 5-3, there were many unrequired headers within the tweet dataset that has been collected. The first step in preprocessing is to remove these unwanted headers within the dataset. Additionally, after each step in the preprocessing, the tweets will be stored in both csv and json format. Csv format is used to allow for easy eyeballing of the twitter dataset and identifying any common irrelevant data that can be filtered away. Json format is still kept for the ease of loading the dataset into a python script for preprocessing.

username	date	text
Justice4UK	1/1/2014	NEW MUSIC VIDEO: Max Wolfe; Doctors of Ruin. http://bit.ly/1l1kmDmÂ . #citigroup,#barclays,#ubs,#wellsfargo
Dom_YC	1/1/2014	$C Happy 2k14 http://instagram.com/p/ipTD3QIoDu/Â
ReikiYoda	1/1/2014	The Seven Levels of Living: http://bit.ly/1dUqgkmÂ . #citigroup,#barclays,#ubs,#wellsfargo
glubb234	1/1/2014	$C - Eleven companies that had a great run in 2013 -> http://stock-news.net/191381Â #stock #stocks #stockaction
passionportfol	1/1/2014	#APA - Apache downgraded to Neutral from Buy at Citigroup - http://bit.ly/1eqXE2IÂ from http://bit.ly/14tVNUAÂ
Justice4UK	1/1/2014	NEW MUSIC VIDEO: Max Wolfe; Doctors of Ruin. http://bit.ly/1l1kmDmÂ . #citigroup,#barclays,#ubs,#wellsfargo
Istanbul_Jobs	1/1/2014	#Istanbul #Jobs Senior Auditor - Markets - Citigroup - Istanbul - Function background/context: Internal Audit is... http://ow.ly/2CzSUPÂ
bestjobsonline	1/1/2014	Residential Mortgage Portfolio Manager - http://bit.ly/1a1zPPMÂ #jobs #Citi #O'Fallon
earningsimpact	1/1/2014	$C reduced assets in Citi Holding by $9 billion in the quarter. Holdings assets now total 122 billion http://stks.co/j03hVÂ
glubb234	1/1/2014	$C - US banks and payouts: facts and theories -> http://stock-news.net/191292Â #stock #stocks #stockaction
RichMatesDare	1/1/2014	Citigroup: work in progress: One of the biggest reasons the bank is at a discount to its peers is the $53bn of... http://on.ft.com/1gnphM3Â

Figure 5-3 Snippet of Tweets after removing unwanted headers

Figure 5-3 shows the csv file after the 1st step of preprocessing, removing unwanted headers, is performed. Only the username, date and text of the tweet is kept whereas the other headers are filtered out. From the Figure 5-3, it can be observed that there are hyperlinks and hashtags within the text field which would affect the sentiment classification performed using our BERT model. In the next step of preprocessing, the hyperlinks and hashtags within texts will be removed.

username	date	text
Justice4UK	1/1/2014	new music video: max wolfe; doctors of ruin. citigroup
Dom_YC	1/1/2014	citigroup happy 2k14
ReikiYoda	1/1/2014	the seven levels of living: . citigroup
glubb234	1/1/2014	citigroup - eleven companies that had a great run in 2013
passionportfol	1/1/2014	apache downgraded to neutral from buy at citigroup - from
Justice4UK	1/1/2014	new music video: max wolfe; doctors of ruin. citigroup
Istanbul_Jobs	1/1/2014	senior auditor - markets - citigroup - istanbul - function background/context: internal audit is...
bestjobsonline	1/1/2014	residential mortgage portfolio manager - citigroup
earningsimpact	1/1/2014	citigroup reduced assets in citi holding by billion in the quarter. holdings assets now total 122 billion
glubb234	1/1/2014	citigroup - us banks and payouts: facts and theories
RichMatesDare	1/1/2014	citigroup: work in progress: one of the biggest reasons the bank is at a discount to its peers is the of...

Figure 5-4 Snippet of Tweets after removing hyperlinks and hashtags

Figure 5-4 shows a snippet of Twitter tweets after the 2nd step of preprocessing is performed. Removal of links and hashtags from the tweets was performed using regex pattern matching in python script. Besides removal of hyperlinks and hashtags, all texts are transformed to lowercase and terms such as “$C”, “citi group”, “#citigroup”, “#citi” which represent the company Citigroup are converted to the term “citigroup” to ensure similarity between all the tweets.

username	date	text
Kaf00	1/29/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup
casuist	1/29/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup
PlayResilience	1/29/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup
azureblade7	1/29/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup
TheBubbleBubble	1/29/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup
OlduvaiNovel	1/29/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup | zero hedge
NoAgendaNewsRSS	1/29/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup: spot the similarit...
opesent	1/29/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup
zerohedge	1/29/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup
heroinnovators	1/30/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup
ccharpenet	1/30/2014	"a funny old world" - the em carry trade collapse 'deja vu, all over again' from citigroup
mrtoxo	1/9/2014	- [$$] citigroup seen as top big bank for 2014 ->
mrtoxo	1/9/2014	- [$$] citigroup seen as top big bank for 2014 ->
stocknews99	1/9/2014	- [$$] citigroup seen as top big bank for 2014 ->

Figure 5-5 Snippet of duplicated tweets within dataset

The final step in preprocessing is the identification and removal of duplicate tweets. As seen from Figure 5-5, there are many duplicated tweets. Moreover, some duplicated tweets are not entirely similar to the original tweet due to additional text trailing the original texts.

```python
with open('clean2\citicleanjson\citi_tweets_removelinks_'+yr+mth+'.json') as json_file:
    data = json.load(json_file)
    new_data = []
    tuple_list = []
    del_list = []
    for i in data: # keys: username, date, text
        tuple_list.append((i['username'],i['date'], i['text']))

    list_len = len(tuple_list)
    for header in tqdm(range(len(tuple_list)), position=0, leave=True):
        runner = header + 1
        while (runner<list_len and tuple_list[header][1] == tuple_list[runner][1]):
            string_len = int(len(tuple_list[runner][2])*0.6)
            if ((tuple_list[runner][2])[:string_len] in tuple_list[header][2]):
                del_list.append(runner)
            runner = runner + 1

    tqdm.write("Citi" + str(yr) + str(mth))
    tqdm.write("# Tweets Before: " + str(len(tuple_list)))
    del_list = unique(del_list)
    tqdm.write("# Duplicate: " + str(len(del_list)))
    del_list.sort()
    del_list.reverse()
    for i in del_list:
        del tuple_list[i]
```

Figure 5-6 Snippet of python code used for removing duplicate tweets

In order to remove all the duplicated tweets from the dataset, I created a python script which checks if the starting 60% of a tweet's text appears in another tweet. If the check returns true, the tweet is considered a duplicate and filtered away. To ensure fast execution, the time taken to remove all duplicates tweets within the dataset was also considered during the creation of the script.

Table 5-6 Number of tweets before and after preprocessing

Stock	# Tweets Before Preprocessing	# Tweets After Preprocessing
SPDR S&P 500 ETF Trust	2,972,169	2,075,689
Citigroup Inc	793,382	446,342
Activision Blizzard, Inc	183,872	87,934
Total	3,949,423	2,609,965

Table 5-6 shows the number of tweets before and after preprocessing is performed. As seen from the table, roughly one-third of tweets extracted before preprocessing has been filtered away, reducing the initial count of 3,949,432 tweets to 2,609,965 tweets.

5.2.2. Preprocessing of News Articles

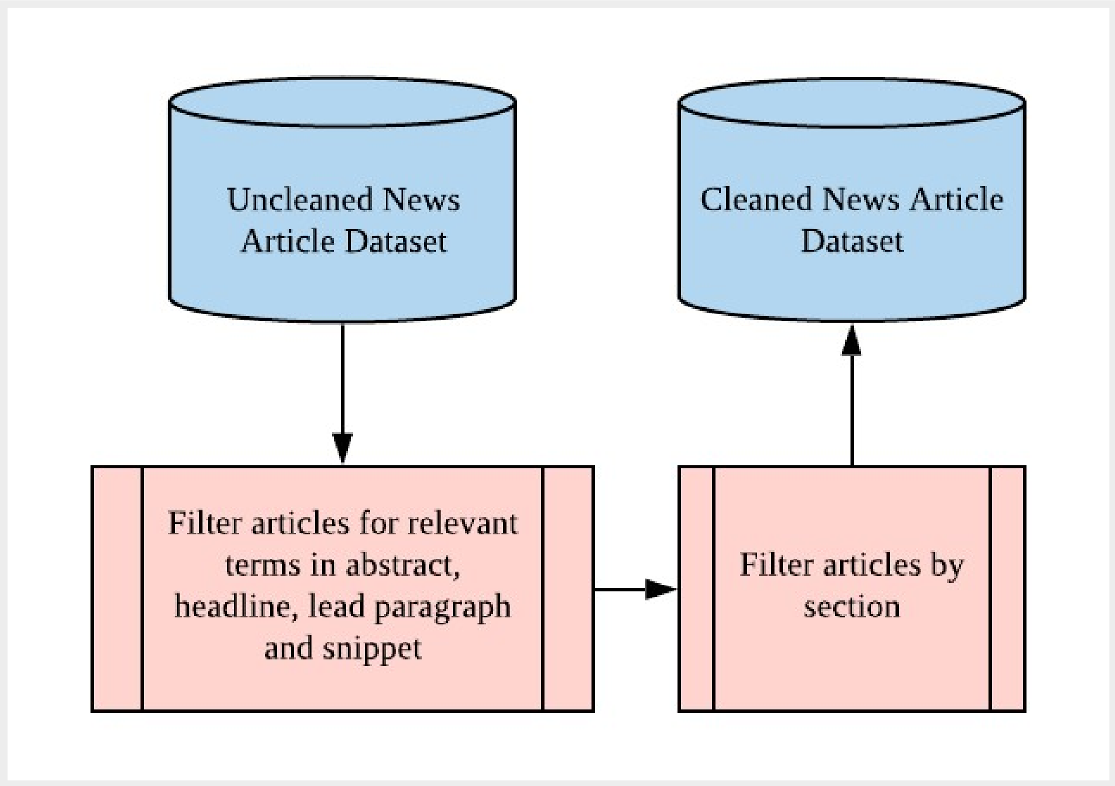


Figure 5-7 Flowchart for Preprocessing of New York Times News Articles

Figure 5-7 shows the flowchart of the steps in the preprocessing of news articles. Preprocessing of articles consists of two steps using a single python script.

1. Filter articles without relevant terms in the abstract, headline, lead paragraph or snippet
2. Filter articles from irrelevant sections

As a general query term was used to extract news articles earlier, certain articles that had been extracted were not related to the companies selected. Hence, the 1st step in preprocessing is to remove all the irrelevant articles from the news article dataset.

Table 5-7 Company specific relevant and irrelevant terms

Stock	Relevant Terms	Irrelevant Terms
SPDR S&P 500 ETF Trust	S.&.P S&P S.& P. S.&P. Standard & Poor	-
Citigroup Inc	Citibank Citigroup Citi Citi's Citi Group Citi Bank	Citi Bike Citi Field
Activision Blizzard, Inc	Activision Blizzard Atvi	-

In Table 5-7, the relevant and irrelevant terms identified are listed for each company. Relevant articles were identified by checking for any relevant terms within the article's abstract, headline, lead paragraph and snippet texts. Any articles that do not contain any relevant terms within the abstract, headline, lead paragraph or snippet texts are filtered away. Besides checking for relevant terms, a further check was performed to identify if the article contained irrelevant terms. This was done specifically for Citigroup Inc news article dataset. Through manual eyeballing of the news article dataset I identified that certain articles were not related to Citigroup but contained the term "citi". Articles containing terms such as "citi field" were not related to the company. I removed any articles which contained irrelevant fields from the Citigroup article dataset.

The 2nd step in preprocessing the news articles is removing any news articles that are from irrelevant sections. Articles from sections: "fashion & style", "arts", "books", "food", "sports", "travel", "the upshot", "blogs", "opinion", "u.s.", "your money" have been deemed as irrelevant sections through manual identification.

Table 5-8 Number of articles before and after preprocessing

Stock	# Articles Before Preprocessing	# Articles After Preprocessing
SPDR S&P 500 ETF Trust	3,865	192
Citigroup Inc	24,515	107
Activision Blizzard, Inc	100	73
Total	28,480	372

Table 5-8 shows the before and after number of articles for the three select stocks. The number of articles for Citigroup Inc dropped significantly after removal of irrelevant articles. After preprocessing of news articles, the total number of news articles dropped from 28,480 to 372. The number of articles is correlated to the average daily trade volume for each stock. With the highest number of articles for SPDR S&P 500 ETF Trust as it is a widely traded stock. Whereas Activision Blizzard had the lowest number of relevant articles due to it being the lowest traded stock of the three.

5.3. Fine-tuning BERT Model

In order to attain state-of-the-art sentiment classification for Twitter tweets, I made use of Google's BERT uncased model for sentiment classification. Before performing sentiment classification, fine-tuning of the BERT model is required. To do so, I made use of Sentiment140 *[23]*, a Kaggle dataset of labelled twitter tweets which consists of 1.6 million positive and negative labelled tweets.

Table 5-9 BERT evaluation

Metric	After fine-tuning
Evaluation Accuracy	0.835
F1 Score	0.831
False Negative	1920
False Positive	1373
True Negative	8627
True Positive	8080
Loss	0.540
Precision	0.855
Recall	0.808

Initially, sentiment140 dataset was divided into 90% training and 10% testing. This means having 1,440,000 training data and 160,000 testing data. This led to an out of memory issue while running the fine-tuning on my local GPU. I tried performing the fine-tuning using Google Collaboratory, however it was met with the same issue as well. To overcome this issue, I incrementally fine-tuned the BERT model and saved the model after each increment. To prevent out of memory issues I had to use a small testing set, splitting the 1.6 million datasets into 1,580,000 training data and 20,000 testing data. Each increment in finetuning further trains the BERT model on the next 16,000 training datapoints. Using this increment method of training, I managed to train the BERT model with 192,000 data points. Table 5-9 shows the BERT model achieved 83.5% prediction accuracy after fine-tuning. Refer to Appendix A, B and C for the pie chart portraying the number of positive, negative and neutral tweets in the three selected stocks.

5.4. Sentiment Classification of Datasets

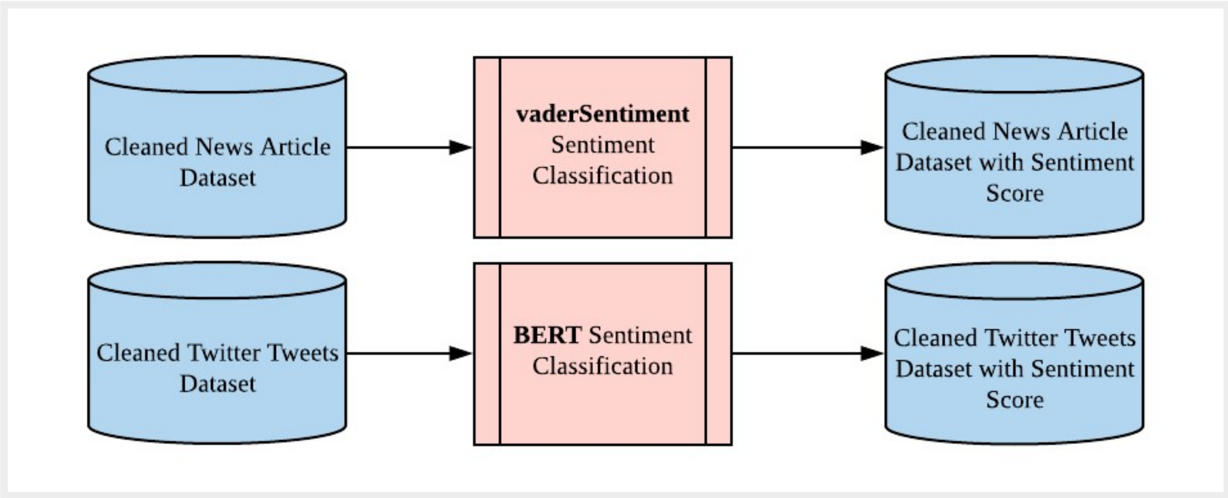


Figure 5-8 Flowchart of sentiment classification

As seen from Figure 5-8, the method of sentiment classification differs for both tweet dataset and news article dataset. The BERT model was only fine-tuned with labelled tweets, hence using it to perform sentiment classification on a news article dataset would not result in precise sentiment classification.

The initial idea was to use a second BERT model that has been fine-tuned for News Article sentiment classification. However, there was no labelled news article dataset available for use online and manual annotation of news articles would be time consuming. Alternatively, I made use of vaderSentiment to perform sentiment classification for the news articles. VaderSentiment *[24]*, Valence Aware Dictionary and sEntiment Reasoner, is an open source tool using rule-based model for sentiment analysis. It has been used in other research *[2]* for sentiment classification of news articles as well. In the following sections, sentiment classification of the respective datasets will be further explained.

5.4.1. Sentiment classification of Tweets dataset

To perform sentiment classification using the fine-tuned BERT model, I made use of Google Collaboratory GPU. Google Collaboratory is a free to use cloud service platform that allows user to leverage the use of Google's GPU on machine learning projects using libraries like Keras and TensorFlow. The use of Google Collaboratory was required to accelerate the sentiment classification process. Performing sentiment classification on my local GPU averages five tweets per second and would require a runtime of at least one week to complete sentiment classification of the extracted twitter dataset. With the use Google Collaboratory GPU, I was able to complete the sentiment classification task within a day.

username	date	text	confidence_1	confidence_2	sentiment
gregnb	1/26/2014	citigroup I	-0.006260309	-5.0766573	-1
EvanOLea	1/26/2014	i'm just ba	-2.5485723	-0.081419595	1
glubb234	1/26/2014	citigroup -	-0.5809861	-0.8194966	0

Figure 5-9 Snippet of BERT Sentiment classification output for Tweets

As seen from Figure 5-9, BERT assigns two different confidence levels, *Confidence_1* and *Confidence_2* to a tweet. *Confidence_1* represents the confidence level that the tweet is Negative whereas *Confidence_2* represents the confidence level that a tweet is Positive

Using the confidence levels assigned by BERT, I assigned sentiments to the tweet based on the following rules.

- A tweet is classified as positive sentiment (1), if *Confidence_2* is higher than *Confidence_1*.
- A tweet is assigned a negative sentiment (-1), if *Confidence_1* is higher than *Confidence_2*.
- A neutral sentiment (0) is assigned if *Confidence_2 – Confidence_1* is within the range of -0.3 to 0.3

5.4.2. Sentiment classification of News Articles dataset

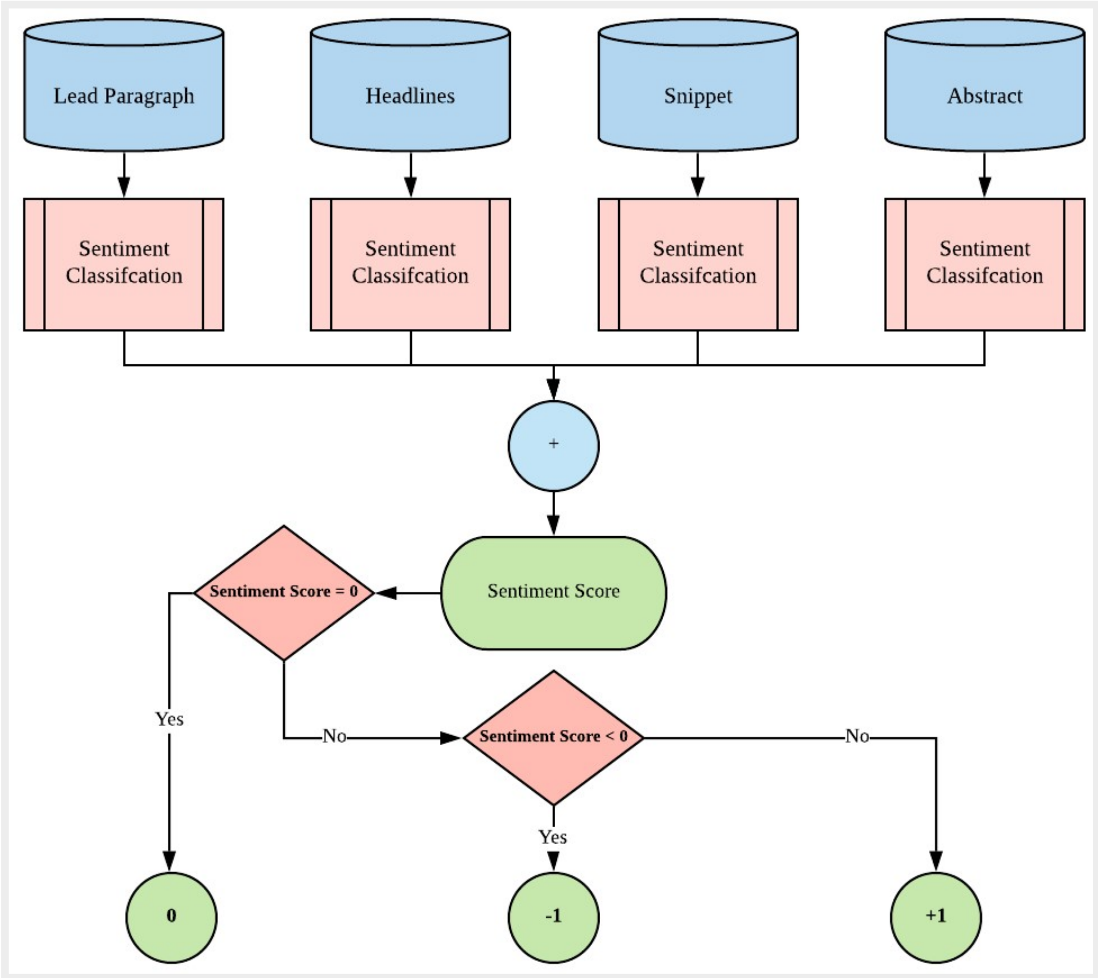


Figure 5-10 Flowchart of vaderSentiment sentiment classification for news articles

date	abstract	headline	snippet	lead_parag	abstract_s	headline_s	snippet_se	lead_parag	sentiment
3/3/2014	A Mexicar	A Record F	A Mexicar	Berkshire I	0	1	0	1	1
3/3/2014	Banamex I	Citigroup A	Banamex I	A headach	0	-1	0	-1	-1
9/2/2014	The advan	Why a Bre	The advan	Nine years	1	1	1	1	1
3/4/2014	A Citigrou	A Calming	A Citigrou	GLOBAL M	0	1	0	-1	0

Figure 5-11 Snippet of output from vaderSentiment sentiment classification

As seen from Figure 5-10, sentiment classification of news articles is done using vaderSentiment to classify the lead paragraph, headlines, snippet and abstract text into positive (1), neutral (0) or negative (-1). vaderSentiment assigns three different confidence level representing the positive, neutral and negative confidence of the text. A simple classification into positive, neutral and negative is done by assigning 1, 0, -1 respectively based on the highest confidence level. After assigning the respective sentiments to the abstract, headline, snippet and lead paragraph, the scores are summed together to get a sentiment score.

Based on the sentiment score which is a summation of the abstract, headline, snippet and lead paragraph sentiments, a sentiment is assigned to the news article

- If the sentiment score is greater than 1, it will be considered as having positive sentiments
- If the sentiment score is less than 1, it will be considered as having negative sentiments
- If it the sentiment score is equal to 0, it will be considered as a neutral sentiment article

Figure 5-11 shows the csv file of news articles after sentiment classification has been performed.

5.4.3. Assigning sentiments score to datasets

After obtaining the sentiments scores for each individual tweets and news articles, the overall tweet sentiment and article sentiment for a single day must be assigned to each trading day that the stock market is open.

The follow formulas are used to calculate the overall sentiments in a day for Twitter tweets and New York Times news article:

Overall tweet sentiment for a day: $$\frac{\Sigma\ Sentiment\ for\ each\ tweet\ for\ that\ day}{Total\ number\ of\ tweets\ for\ that\ day}$$

Overall article sentiment for a day: $$\frac{\Sigma\ Sentiment\ for\ each\ article\ for\ that\ day}{Total\ number\ of\ articles\ for\ that\ day}$$

Date	Open	High	Low	Close	Adj Close	Volume	Tweet_Sentiment	Article_Sentiment
2/20/2014	183.27	184.52	182.6	184.1	163.6328	1.05E+08	0.100392157	-1
2/21/2014	184.45	184.89	183.8	183.89	163.4462	1.18E+08	0.080645161	0
2/24/2014	184.28	186.15	184.2	184.91	164.3528	1.14E+08	0.221943569	-1
2/25/2014	185.06	185.59	184.23	184.84	164.2906	1.17E+08	0.043538355	1
2/26/2014	185.11	185.6	184.33	184.85	164.2994	98677200	0.04038893	0
2/27/2014	184.58	185.87	184.37	185.82	165.1617	93880800	0.014164306	1

Figure 5-12 Sentiment scores assigned to individual trading days

Figure 5-12 shows the tweet sentiment and article sentiment assigned to the historical stock price dataset. As seen from Figure 5-12, there are certain days where the stock exchange is closed. To overcome this issue, sentiments from tweets and news articles of these days were brought forward to the day that the stock market reopens. The calculation for the overall sentiment of the day when the stock market reopens accounts for the number of tweets and articles brought forward.

5.5. Implementation of LSTM models

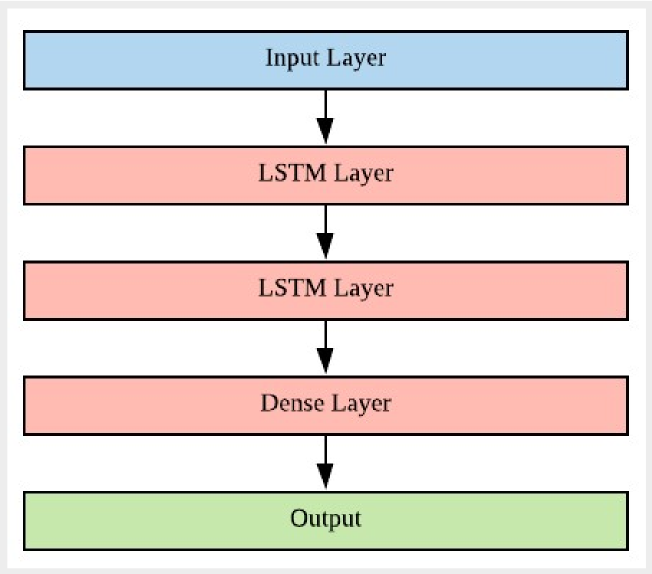


Figure 5-13 Two-Layer LSTM NN Architecture

```
def build_LSTM(amount_of_features, seq_len, num_units, dropout_rate, ):

  layers = [amount_of_features,seq_len,1]
  regressor = Sequential()
  # LSTM Layer 1
  regressor.add(CuDNNLSTM(units=num_units, return_sequences=True, input_shape=(layers[1], layers[0])))
  regressor.add(Dropout(dropout_rate))
  #LSTM Layer 2
  regressor.add(CuDNNLSTM(units=num_units))
  regressor.add(Dropout(dropout_rate))
  # output layer
  regressor.add(Dense(units=1, activation='relu'))
  opt = optimizers.Adam(lr=0.001, decay=1e-6)
  regressor.compile(optimizer=opt, loss='mean_squared_error')

  return regressor
```

Figure 5-14 Snippet of code for Two-Layer LSTM NN Architecture

Figure 5-13 shows the architecture used in creating all our LSTM models and Figure 5-14 shows the code snippet of the LSTM architecture built for our models. The model architecture consists of one input layer, two LSTM layers and one dense layer. The implementation of LSTM stock price prediction models is done on Google Collaboratory to allow for a quicker training time using the provided GPU.

The stock price dataset with sentiment scoring is split into 80% training and 20% testing set. The dataset stretches for five years starting from the 1st January 2014 to 31st Dec 2018 hence the training set consists of data from the year 2014 to 2017 and our testing set will consist of data in the year 2018. A total of four LSTM models will be created for the three selected companies after which the results will be discussed.

The loss function used to measure our model's accuracy is Mean Square Error (MSE) and Root Mean Square Error (RMSE)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

```
lstm = build_LSTM(amount_of_features, seq_len, num_hidden_neuron = num_hid_neuron, dropout_rate = d_rate)
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience = p_val)
mc = ModelCheckpoint(file_name,  monitor='val_loss', mode='min', verbose=1, save_best_only=True)
lstm.fit(X_train1, y_train1,
        validation_data = (X_test, y_test),
        epochs=num_of_epochs,
        batch_size=size_of_batch,
        callbacks=[es, mc], verbose=0)
```

Figure 5-15 Snippet of code for Model Checkpoint and Early Stopping

In this paper, a total of four LSTM models will be built using different features. For the purpose of only using the best results, I made use of Keras Early Stopping with a patience value of 150 epochs to prevent the model from overfitting. Additionally, I made use Keras Model Checkpoint, which saves the model after each epoch if it achieved a lower validation loss as compared to the previous saved model. The last saved model after early stopping is triggered would be used to evaluate the MSE and RMSE for that run. Figure 5-15 shows the code snippet for Model Checkpoint and Early Stopping. Besides the use of Early Stopping and Model Check Point, the MSE and RMSE results for all models built will be taken from an average taken of 10 runs.

These are the four LSTM model implemented within this paper:

1. Baseline LSTM model
2. Tweet Sentiment LSTM Model
3. Article Sentiment LSTM Model
4. Tweet and Article Sentiment LSTM Model

Baseline LSTM Model

The baseline LSTM model is created using the stocks historical closing price as the only feature in the stock price prediction model. The closing price is scaled between 0 and 1 before input training the LSTM model.

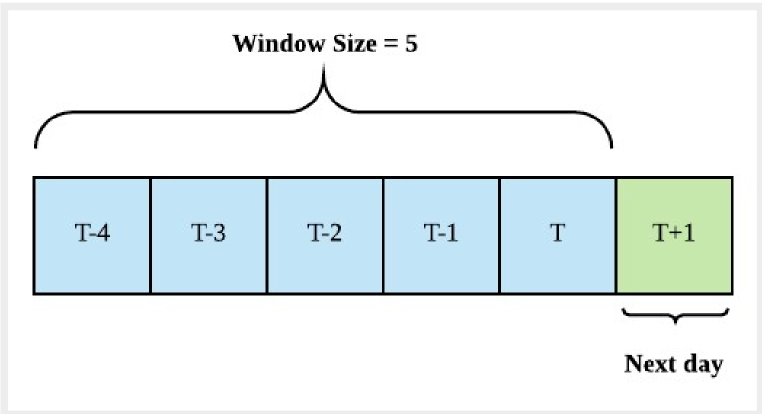


Figure 5-16 Graphical Representation of selecting window size 5

To create the optimal baseline model for each stock, a comparison of window sizes ranging 2 to 10 is done to identify the optimal window size to use. The window size represents the number of consecutive previous days used to predict the next day value. Figure 5-16 depicts a selection of window size 5 for the prediction of the next day's value. After identifying the optimal window size of each stock, the optimal window size will be used in the creation of the other three LSTM models.

Tweet Sentiment LSTM Model

The 2nd LSTM model is created using the stock's closing price along with tweet sentiments. Similar to the baseline model, the closing price is scaled between 0 and 1. For tweet sentiments, a sentiment decay coefficient is applied to the tweet sentiments within the selected window size. This is done to implement sentiment relevancy to the sentiment data.

In view of the fact that positive news which has happened a few days back would not hold as much weight as a negative news that just happened today, the sentiment decay coefficient is introduced to indicate the decaying importance of sentiment data over time. To implement sentiment relevancy to our tweet sentiments, a sentiment decay coefficient which indicates the decaying importance of sentiment data after each day is multiplied to our tweet sentiment scores for each window. Similar time-relevance of data was implemented in both *[2]* and *[14]*.

Formula for applying sentiment decay to sentiment score within the window:

$$\boldsymbol{x_k = x_k(1-\alpha)^k}$$

$\boldsymbol{\alpha}$: Decay coefficient

$\boldsymbol{k}$: Window index, $0 \leq \boldsymbol{k} <$ window size

$\boldsymbol{x_k}$: Tweet sentiment at window index k

The formula for calculating the sentiment score after sentiment decay is $\boldsymbol{x_k(1-\alpha)^k}$ where $\boldsymbol{\alpha}$ is the decay coefficient, $\boldsymbol{x_k}$ is the sentiment score at the $\boldsymbol{k^{th}}$ index within the window size and . The sentiment score, $\boldsymbol{x_k}$, is multiplied by a multiplier, $\boldsymbol{(1-\alpha)^k}$, which varies based $\boldsymbol{k}$. $\boldsymbol{k}$ is a value between 0 and the selected window size.

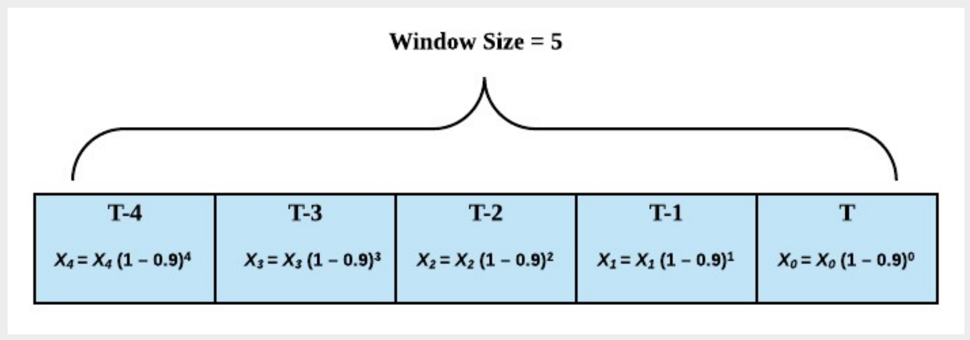


Figure 5-17 Graphical Representation of 90% sentiment decay

Using a 0% sentiment decay, $\boldsymbol{\alpha = 0}$, indicates that the sentiments from yesterday is equally as relevant as the sentiments of today whereas a using a sentiment decay of 90%, $\boldsymbol{\alpha = 0.9}$, would indicate that the sentiments from yesterday is only 10% relevant to the sentiments of today. Sentiments further down the window size would be multiplied by a smaller multiplier $\boldsymbol{(1 - \alpha)^k}$, which is indicative of the decaying relevancy of the sentiment as more days have passed. Figure 5-17 shows the graphical representation of 90% sentiment decay being applied to window size of 5. An exponential decay function where the decay function is changed to exponential, $\boldsymbol{x_k = x_k e^{-k}}$, is experimented with as well.

<u>Article Sentiment LSTM Model</u>

The 3rd LSTM model is created using the tocks closing price along with sentiments obtained from news articles. Similar to the Tweet Sentiment LSTM model, a sentiment decay coefficient is implemented to article sentiment for each window. A test to identify the optimal sentiment decay coefficient for articles sentiments is similarly performed on the three selected companies.

<u>Tweet and Article Sentiment LSTM Model</u>

The final stock price prediction model incorporates both tweet sentiments and news article sentiments along with closing price as features. The optimal sentiment decay coefficient for tweet sentiments and article sentiment identified earlier will be used in the creation of the final LSTM stock price prediction model.

6. Experiment Results

This section will review the results of the four LSTM models created for each of the selected stocks.

6.1. Baseline Model

Table 6-1 Baseline model evaluation for optimal window size

Company	SPDR S&P 500 ETF Trust		Citigroup Inc		Activision Blizzard Inc	
Window Size	MSE	RMSE	MSE	RMSE	MSE	RMSE
2	1.038E-03	3.222E-02	6.288E-04	2.508E-02	1.179E-03	3.434E-02
3	**1.024E-03**	**3.199E-02**	6.441E-04	2.538E-02	**1.140E-03**	**3.376E-02**
4	1.073E-03	3.276E-02	6.354E-04	2.521E-02	1.226E-03	3.502E-02
5	1.119E-03	3.345E-02	6.442E-04	2.538E-02	1.297E-03	3.602E-02
6	1.142E-03	3.380E-02	6.447E-04	2.539E-02	1.343E-03	3.665E-02
7	1.163E-03	3.410E-02	6.458E-04	2.541E-02	1.373E-03	3.706E-02
8	1.177E-03	3.430E-02	6.310E-04	2.512E-02	1.382E-03	3.718E-02
9	1.179E-03	3.434E-02	6.241E-04	2.498E-02	1.364E-03	3.693E-02
10	1.179E-03	3.434E-02	**6.222E-04**	**2.494E-02**	1.394E-03	3.734E-02

Optimal window sized for each stock is identified by the lowest MSE and RMSE result for the respective stocks. From Table 6-1, the optimal window size identified for SPDR S&P 500 ETF Trust is 3, Citigroup Inc is 10 and for Activision Blizzard Inc is 3. Additionally, the lowest MSE and RMSE obtained for each stock is colored in red. All MSE and RMSE values are an average taken from 10 runs.

6.2. Tweet Sentiment Model

Table 6-2 Tweet Sentiment model evaluation

Company	SPDR S&P 500 ETF Trust		Citigroup Inc		Activision Blizzard Inc	
Decay Coefficient	**MSE**	**RMSE**	**MSE**	**RMSE**	**MSE**	**RMSE**
0%	1.029E-03	3.207E-02	6.256E-04	2.501E-02	**1.134E-03**	**3.368E-02**
25%	1.023E-03	3.198E-02	6.265E-04	2.503E-02	**1.133E-03**	**3.365E-02**
50%	1.031E-03	3.211E-02	**6.187E-04**	**2.487E-02**	**1.137E-03**	**3.371E-02**
75%	**1.015E-03**	**3.186E-02**	**6.211E-04**	**2.492E-02**	1.149E-03	3.389E-02
90%	**1.010E-03**	**3.178E-02**	6.269E-04	2.504E-02	1.176E-03	3.429E-02
92.5%	**1.007E-03**	**3.173E-02**	6.285E-04	2.507E-02	1.191E-03	3.451E-02
95%	**1.015E-03**	**3.186E-02**	6.260E-04	2.502E-02	1.205E-03	3.471E-02
97.5%	**1.022E-03**	**3.196E-02**	6.278E-04	2.506E-02	1.222E-03	3.495E-02
Exponential	1.027E-03	3.204E-02	**6.214E-04**	**2.493E-02**	1.141E-03	3.378E-02

Table 6-2 shows the results obtained on the three selected companies using the Tweet Sentiment model along with varying decay coefficients and stock historical closing price. For the three selected companies, a test to identify the optimal sentiment decay coefficient ranging from 0% to 97.5% was performed.

The optimal decay coefficient identified for SPDR S&P 500 ETF Trust is 92.5%, Citigroup Inc is 50% and for Activision Blizzard Inc is 25%. Optimal decay coefficient for each stock is identified by the model which attained the lowest MSE and RMSE. Results that have a lower MSE and RMSE score compared to the baseline model is colored in red. All MSE and RMSE obtained within the Table 6-2 are from an average of 10 runs.

6.3. Article Sentiment Model

Table 6-3 Article Sentiment model evaluation

Company	SPDR S&P 500 ETF Trust		Citigroup Inc		Activision Blizzard Inc	
Decay Coefficient	**MSE**	**RMSE**	**MSE**	**RMSE**	**MSE**	**RMSE**
0%	**1.021E-03**	**3.196E-02**	6.229E-04	2.496E-02	**1.139E-03**	**3.375E-02**
25%	**1.021E-03**	**3.195E-02**	**6.176E-04**	**2.485E-02**	1.148E-03	3.388E-02
50%	**1.022E-03**	**3.197E-02**	6.227E-04	2.495E-02	1.154E-03	3.397E-02
75%	1.042E-03	3.228E-02	6.217E-04	2.493E-02	1.168E-03	3.418E-02
90%	1.072E-03	3.275E-02	**6.209E-04**	**2.492E-02**	1.154E-03	3.397E-02
92.5%	1.082E-03	3.289E-02	**6.221E-04**	**2.494E-02**	1.152E-03	3.395E-02
95%	1.082E-03	3.289E-02	6.231E-04	2.496E-02	1.157E-03	3.401E-02
97.5%	1.058E-03	3.252E-02	**6.220E-04**	**2.494E-02**	1.144E-03	3.383E-02
Exponential	1.029E-03	3.208E-02	**6.178E-04**	**2.486E-02**	1.165E-03	3.414E-02

Table 6-3 shows the results obtained on the three selected companies using the Article Sentiment model along with varying decay coefficients. Similarly, a test to identify the optimal sentiment decay coefficient ranging from 0% to 97.5% was performed.

The optimal decay coefficient identified for SPDR S&P 500 ETF Trust is 25%, Citigroup Inc is 25% and for Activision Blizzard Inc is 0%. Optimal decay coefficient is identified by the model which attained the lowest MSE and RMSE. Results that have a lower MSE and RMSE score compared to the baseline model is colored in red. All MSE and RMSE obtained within the Table 6-3 are from an average of 10 runs.

6.4. Tweet and Article Sentiment Model

Table 6-4 Tweet and Articles Sentiment model evaluation

Company	Optimal Window Size	Tweet Sentiment Decay Coefficient	Article Sentiment Decay Coefficient	MSE	RMSE
SPDR S&P 500 ETF Trust	3	92.5%	25%	1.007E-03	3.173E-02
Citigroup Inc	10	50%	25%	6.144E-04	2.479E-02
Activision Blizzard Inc	3	25%	0%	1.132E-03	3.364E-02

Table 6-4 shows the selected window size, tweet sentiment decay coefficient and article sentiment decay coefficient along with the results obtained on the three selected companies using the Tweet and Article Sentiment model. The MSE and RMSE obtained within Table 6-4 are the results from an average of 10 runs.

6.5. Model Plots

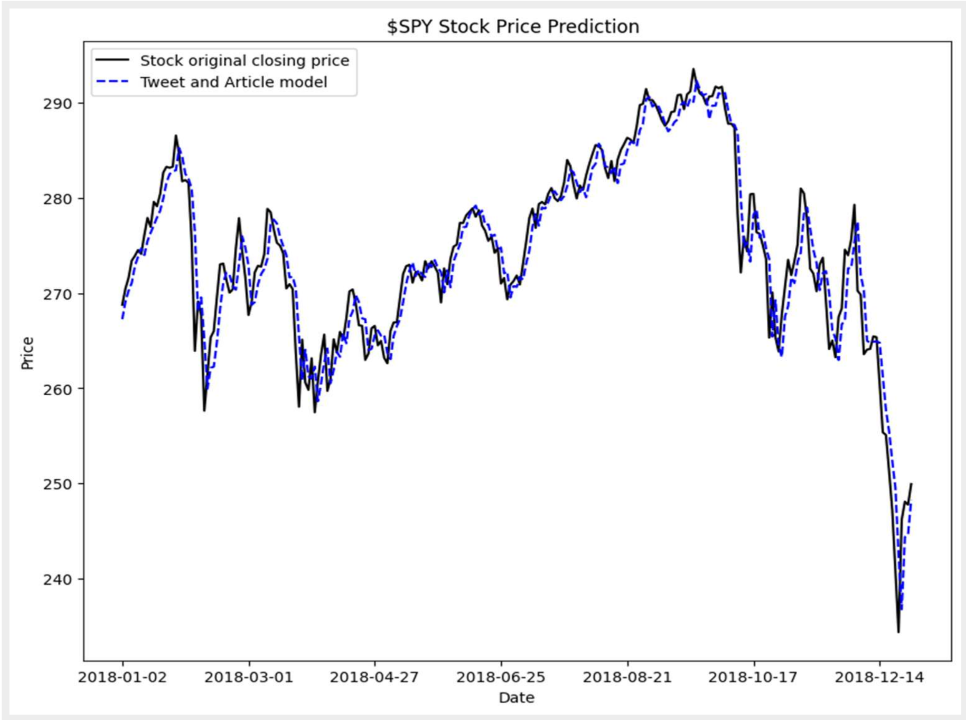


Figure 6-1 Plot for $SPY tweet and article LSTM model

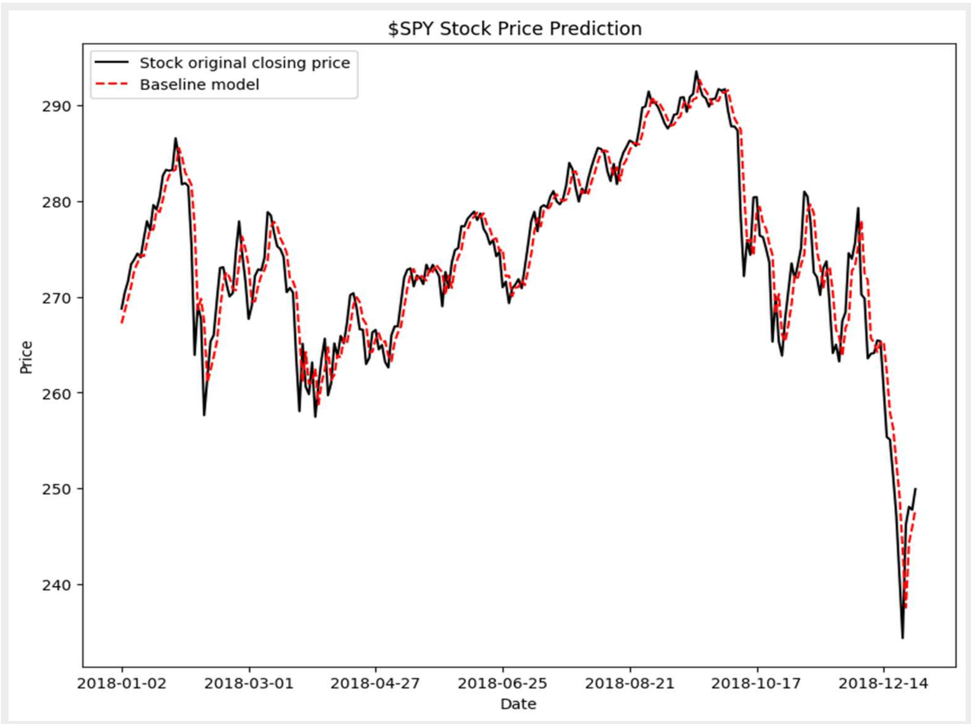


Figure 6-2 Plot for $SPY baseline LSTM model

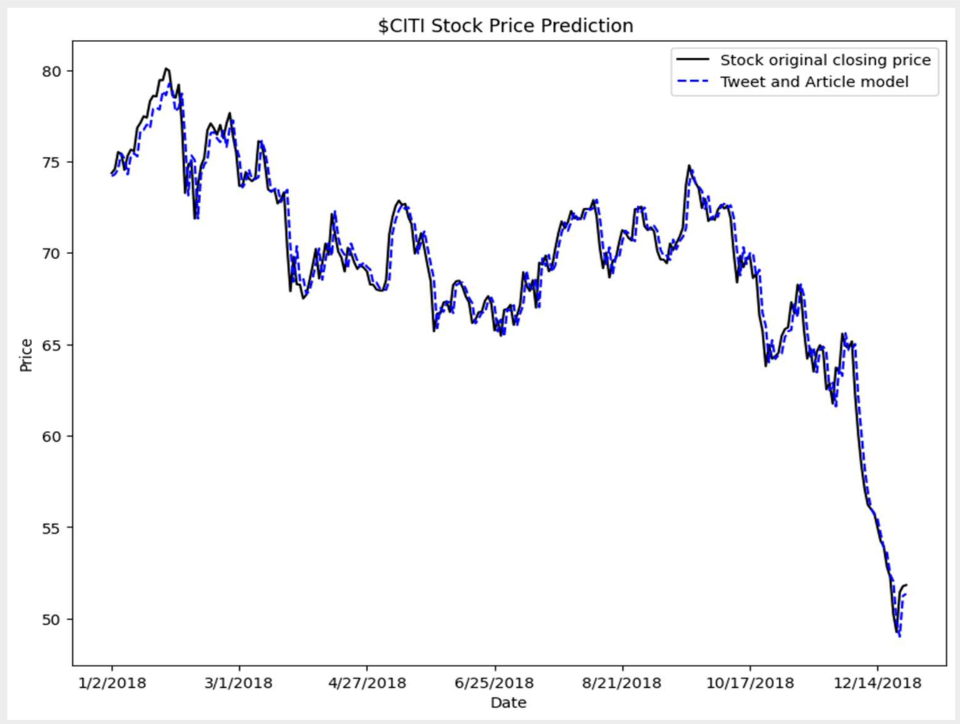


Figure 6-3 Plot for $CITI tweet and article LSTM model

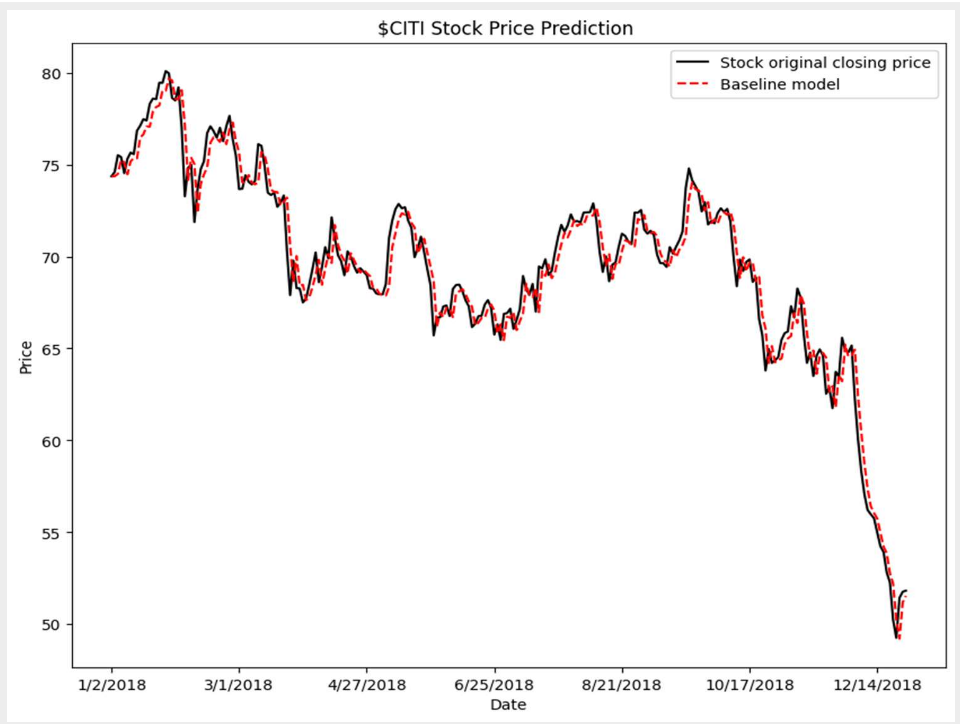


Figure 6-4 Plot for $CITI baseline LSTM model

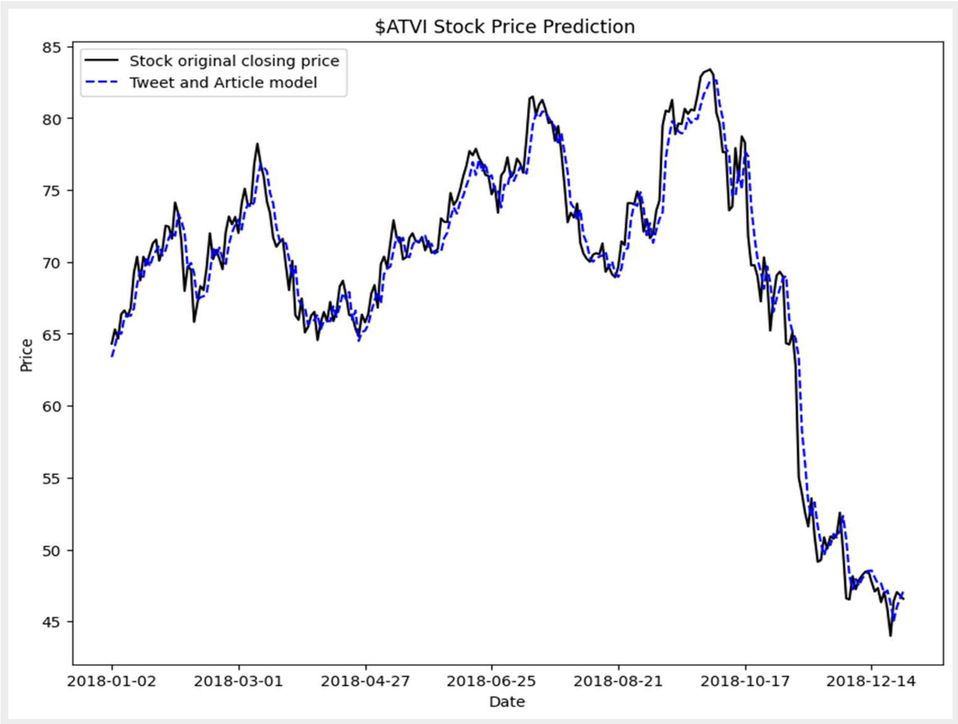


Figure 6-5 Plot for $ATVI tweet and article LSTM model

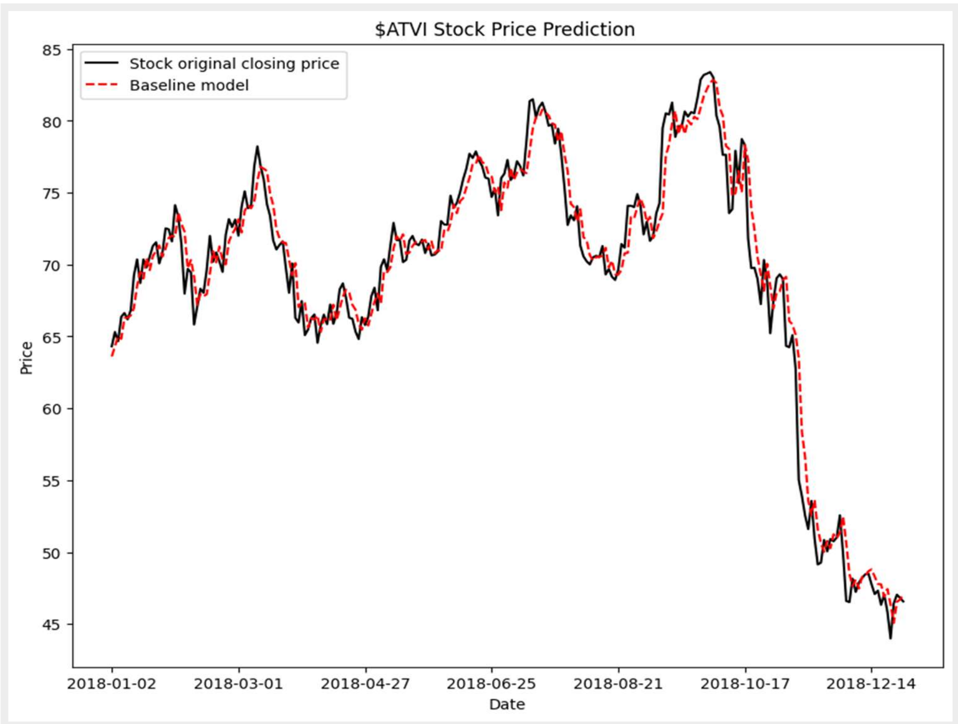


Figure 6-6 Plot for $ATVI baseline LSTM model

7. Discussion of Results

Table 7-1 LSTM models comparison and evaluation

Company		Window Size	Baseline (1)	Tweet Sentiment (2)	Article Sentiment (3)	Tweet and Article Sentiment (4)
SPDR S&P 500 ETF Trust	MSE	3	1.024E-03	**1.007E-03**	1.021E-03	**1.007E-03**
	RMSE		3.199E-02	**3.173E-02**	3.195E-02	**3.173E-02**
Citigroup Inc	MSE	10	6.222E-04	6.187E-04	6.176E-04	**6.144E-04**
	RMSE		2.494E-02	2.487E-02	2.485E-02	**2.479E-02**
Activision Blizzard Inc	MSE	3	1.140E-03	1.133E-03	1.139E-03	**1.132E-03**
	RMSE		3.376E-02	3.365E-02	3.375E-02	**3.364E-02**

Table 7-1 shows the summary of the results obtained from the four implemented models. The best results obtained for each stock is colored in red. To recap, the objective of this project was to explore the usage of sentiments obtained from tweets and news articles to create an enhanced stock price prediction model. This was achieved through the implementation and comparison of four different LSTM models.

Starting with the creation of the baseline model and the identification of optimal window size. The results obtained from the baseline model in Table 6-1 identified differing optimal window sizes for Citigroup Inc, 10, as compared to SPDR S&P 500 ETF Trust and Activision Blizzard Inc, 3. Different stocks have different optimal window sizes and further experiments and testing on using window sizes larger than 10 could be done in future research. The implementation of the baseline model provided the baseline MSE and RMSE results that would be use for comparison with the other three LSTM models.

The implementation of the second LSTM model, Tweet sentiment model, aimed to incorporate stock specific tweet sentiments along with closing price to generate an improved stock price prediction model. Based on the premise that sentiments from a few days back should not be equally important to the sentiments of today, a sentiment decay coefficient was introduced to implement relevancy of the sentiments. Seen from the results in Table 6-2, implementation of sentiment decay to the sentiment scores within a window size was crucial in creating an improved LSTM model as compared to the baseline model. A notable point is, if an unsuitable sentiment decay coefficient is selected, it could lead to a poorer performing stock price prediction model. Additionally, it can be noted that different stocks have differing optimal sentiment decay coefficient. These could be said similarly for the third LSTM model which made use of news article sentiments instead of tweet sentiments.

Using the results obtained from both the Tweet sentiment LSTM model and News Article sentiment LSTM models in Table 6-2 and 6-3, it can be concluded that using an optimal sentiment decay coefficient is important to improving the model accuracy and that the optimal sentiment decay differs for each stock. Using an unsuitable sentiment decay coefficient would result in a worse performing model. Additionally, it can be established that sentiments extracted from tweets and news articles does improve a stock price prediction model.

The final LSTM model, Tweet and Article sentiment model, made use of sentiments from both tweets and articles to create an enhanced stock price prediction model. The intention for using sentiments from both tweets and articles is to determine whether the inclusion of sentiments from a more than one source would further enhance the stock price prediction model. Results in Table 7-1 point out that the final LSTM model generated the best models out of the four LSTM models for Citigroup Inc and Activision Blizzard Inc. Whereas the final model for SPDR S&P 500 ETF Trust achieved a similar MSE and RMSE results to its Tweet sentiment LSTM model. These results shown in Table 7-1 positively indicates that the inclusion of sentiments from different sources enhances the overall performance of the stock price prediction model.

Moreover, from Table 7-1, comparing between the results of Article sentiment LSTM model and the Tweet sentiment LSTM model, it can be seen that the for SPDR S&P 500 ETF Trust and Activision Blizzard Inc, the Tweet sentiment model achieved a better result as compared to the Article sentiment model. This is in contrasts with Citigroup Inc where the Article sentiment model achieved a better result as compared to the Tweet sentiment model. A speculation for this contrast in results could be due to the differing window sizes used. SPDR S&P 500 ETF Trust and Activision Blizzard Inc uses a window size of 3 whereas the window size used for Citigroup Inc is 10.

Figure 6-1 to Figure 6-6 shows the graphical plots for both the enhanced LSTM model which incorporated tweet and news article sentiment and baseline LSTM model against the testing data. It can be observed that the baseline model predictions and enhanced model predictions are closely similar. However, slight difference between the enhanced and baseline model can be identified for $SPY (SPDR S&P 500 ETF Trust) and $CITI (Citigroup Inc).

On closer inspection of $SPY model plots in Figure 6-1 and Figure 6-2, a tighter fit to the stock price in the enhanced model plot as compared to the baseline model can be identified. Similarly, taking a closer look at Figure 6-3 and Figure 6-4 for $CITI, a tighter fit for the enhanced model's plot can be observed as compared to the baseline model.

On the contrary, the baseline model plot and enhanced model plot for $ATVI seen in Figure 6-5 and Figure 6-6 is closely similar and the difference between the enhanced and baseline model is ambiguous. This could be due to the fact that the enhanced model for $ATVI only achieved a menial improvement of 0.007 MSE as compared to the baseline model. A speculation for the slight improvement in MSE for the $ATVI model could be correlated to its lower daily stock transaction volume along with lower number of extracted news articles and tweets as compared to the other two selected stocks.

8. Conclusions & Future Works

This section will summarize the accomplishment within this project and address potential research and further works.

8.1. Conclusions

This project proposes the use of sentiments from stock specific Twitter tweets and New York Times news articles in the creation of an enhanced stock price prediction model. Employing the use of BERT for sentiment classification of tweets and vaderSentiment for sentiment classification of news articles, this project implemented and compared a baseline LSTM stock price prediction model against three enhanced LSTM models which incorporated sentiment scorings.

The results achieved from implementation and comparison of the four LSTM stock price prediction model concluded that the consideration of sentiments is essential in building an enhanced stock price prediction model. Results from this project on LSTM models which incorporated sentiments has shown improved performance at stock price prediction as compared to the baseline model which only used the stock's historical closing price.

Moreover, an enhanced stock price prediction model which incorporated sentiment scores from both tweets and news articles was able to produce lower MSE and RMSE result for two out of three of the selected stocks. This result has demonstrated that inclusion of a wider range of sentiments would result in an enhanced stock price prediction model.

Additionally, this project investigated sentiment relevancy based on the premise that sentiments from the past are not as relevant as the sentiments of today. Results from the experiments revealed that sentiment decay plays a crucial role when building incorporating sentiments scoring to the LSTM model. This result is consistent with the proposition that sentiments from the past are not as relevant as present-day sentiments.

8.2. Future Works

Firstly, in this project, the LSTM models built did not incorporate trade volume, number of tweets and number of articles as features to create a stock price prediction more. Future works could explore the possibility of incorporating these features to create an improved stock price prediction model. Usage of a larger dataset such as a 10-year dataset instead of 5-year could be explored as well.

Secondly, future works could experiment with different parameters. This project only tested using window size ranging from 2 to 10, experiment using window sizes larger than 10 could be done. Furthermore, the LSTM model architecture implemented within this project was a two-layer stacked LSTM, further research could be done on experimenting with more LSTM layers. Additionally, experiments could be performed to identify a correlation between sentiment decay and a selected window size.

Thirdly, manual annotation and fine-tuning of a BERT model for news article sentiment classification could be done as well. This would allow for more accurate sentiment classification of the article dataset and hopefully produce a more accurate stock price prediction model.

Lastly, future research could incorporate a wider range of sentiments. This project only made use of sentiments obtained from 2.6 million Twitter tweets and 372 New York Times news articles within a five-year range. In future works, extraction and use of news articles from other websites such as Bloomberg and Reuters along with texts from social media sites such as Weibo, Reddit and Facebook could be explored in the creation of a stock price prediction model. By using a wider set of sentiment data obtained from different sources, it could depict a more accurate reflection of public opinions and hopefully produce an improved stock price prediction model.

References

[1] D. Shah, H. Isah and F. Zulkernine, "Predicting the Effects of News Sentiments on the Stock Market," *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, 2018, pp. 4705-4708.

[2] Z. Wang, S. Ho and Z. Lin, "Stock Market Prediction Analysis by Incorporating Social and News Opinion and Sentiment," *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, Singapore, Singapore, 2018, pp. 1375-1380.

[3] T. Sun, J. Wang, P. Zhang, Y. Cao, B. Liu and D. Wang, "Predicting Stock Price Returns Using Microblog Sentiment for Chinese Stock Market," *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, Chengdu, 2017, pp. 87-96.

[4] M. G. Sousa, K. Sakiyama, L. d. S. Rodrigues, P. H. Moraes, E. R. Fernandes and E. T. Matsubara, "BERT for Stock Market Sentiment Analysis," *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, Portland, OR, USA, 2019, pp. 1597-1601.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers),* 2019, pp. 4171–4186.

[6] Y. Kara, M. Boyacioglu and O. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange," *Expert Syst. Appl*., vol. 38, pp. 5311-5319, 2011.

[7] N. Rajesh and L. Gandy, "CashTagNN: Using sentiment of tweets with CashTags to predict stock market prices," *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, Mohammedia, 2016, pp. 1-4.

[8] L. Sayavong, Z. Wu and S. Chalita, "Research on Stock Price Prediction Method Based on Convolutional Neural Network," *2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, Jishou, China, 2019, pp. 173-176.

[9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation,* vol. 9, no. 8, pp. 1735–1780, 1997.

[10] T. Gao, Y. Chai and Y. Liu, "Applying long short term momory neural networks for predicting stock closing price," *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, 2017, pp. 575-578.

[11] F. Qian and X. Chen, "Stock Prediction Based on LSTM under Different Stability," *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, Chengdu, China, 2019, pp. 483-486.

[12] C. Wang and Q. Gao, "High and Low Prices Prediction of Soybean Futures with LSTM Neural Network," *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2018, pp. 140-143.

[13] K. A. Althelaya, E. M. El-Alfy and S. Mohammed, "Evaluation of bidirectional LSTM for short-and long-term stock market prediction," *2018 9th International Conference on Information and Communication Systems (ICICS)*, Irbid, 2018, pp. 151-156.

[14] Z. Zhao, R. Rao, S. Tu and J. Shi, "Time-Weighted LSTM Model with Redefined Labeling for Stock Trend Prediction," *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, Boston, MA, 2017, pp. 1210-1217.

[15] S. Asur and B. Huberman, "Predicting the Future with Social Media," *in: Proceedings of the ACM International Conference on Web Intelligence*, 2010, pp. 492-499.

[16] S. Aich, H. Kim, M. Sain and B. B. Deo, "Analyzing stock price changes using event related Twitter feeds," *2017 19th International Conference on Advanced Communication Technology (ICACT)*, Bongpyeong, 2017, pp. 484-487.

[17] Q. Li, B. Zhou, and Q. Liu "Can twitter posts predict stock behavior?: A study of stock market with twitter social emotion," *2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, Chengdu, 2016, pp. 359-364.

[18] V. S. Pagolu, K. N. Reddy, G. Panda and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, Paralakhemundi, 2016, pp. 1345-1350.

[19] W. Chen, Y. Zhang, C. K. Yeo, C. T. Lau and B. S. Lee, "Stock market prediction using neural network through news on online social networks," *2017 International Smart Cities Conference (ISC2)*, Wuxi, 2017, pp. 1-6.

[20] Y. Wang and Y. Wang, "Using social media mining technology to assist in price prediction of stock market," *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, Hangzhou, 2016, pp. 1-4.

[21] H. Alostad and H. Davulcu, "Directional Prediction of Stock Prices Using Breaking News on Twitter," *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Singapore, 2015, pp. 523-530.

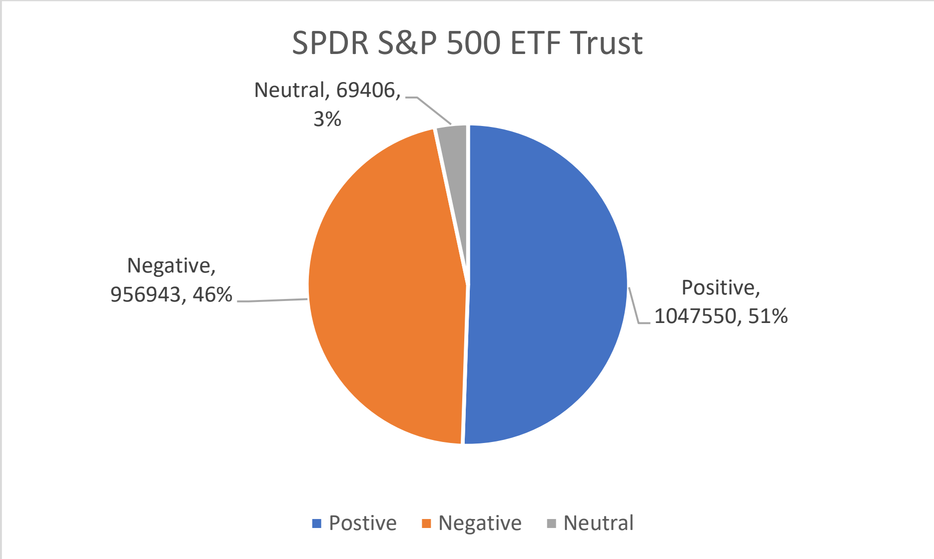
[22] A. Taspinar, "twitterscraper," *github.com*, June, 2016. [Online]. Available: https://github.com/taspinar/twitterscraper/. [Accessed Feb. 22, 2020]

[23] A. Go, R. Bhayani and L. Huang "Twitter sentiment classification using distant supervision," *CS224N Project Report,* Stanford, pp. 1-12, 2009.

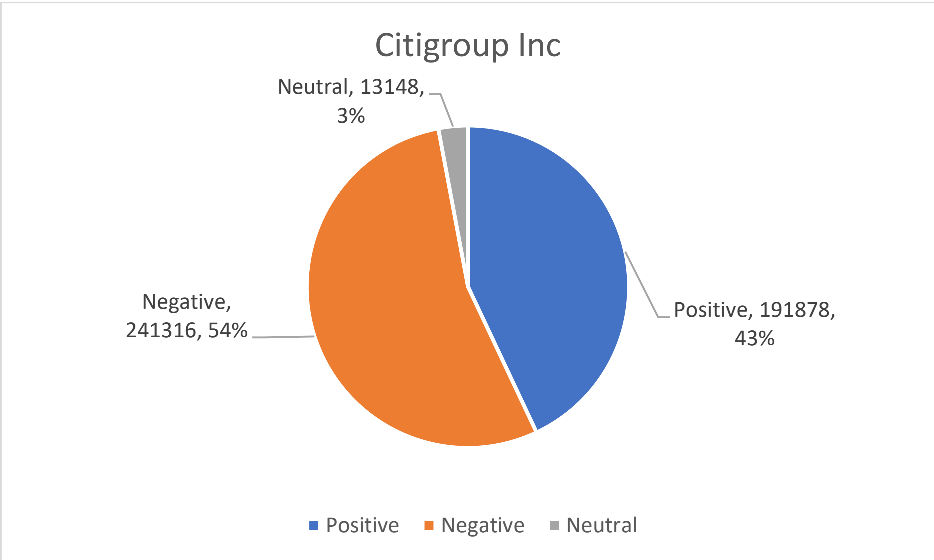
[24] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, Ann Arbor, MI, 2014, pp. 216-226

Appendix

Appendix A: Pie chart of Tweet sentiments for SPDR S&P 500 ETF Trust



Appendix B: Pie chart of Tweet sentiments for Citigroup Inc.



Appendix C: Pie chart of Tweet sentiments for Activision Blizzard Inc.