Stock Market Prediction Through Sentiment Analysis of Social-Media and Financial Stock Data Using Machine Learning

By

Mohammad Al Ridhawi

Thesis Submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Master of Science in Data Transformation and Innovation

School of Electrical Engineering and Computer Science
Faculty of Engineering

University of Ottawa

University of Ottawa

Abstract

Given the volatility of the stock market and the multitude of financial variables at play, forecasting the value of stocks can be a challenging task. Nonetheless, such prediction task presents a fascinating problem to solve using machine learning.

The stock market can be affected by news events, social media posts, political changes, investor emotions, and the general economy among other factors. Predicting the stock value of a company by simply using financial stock data of its price may be insufficient to give an accurate prediction. Investors often openly express their attitudes towards various stocks on social medial platforms. Hence, combining sentiment analysis from social media and the financial stock value of a company may yield more accurate predictions.

This thesis proposes a method to predict the stock market using sentiment analysis and financial stock data. To estimate the sentiment in social media posts, we use an ensemble-based model that leverages Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN) models. We use an LSTM model for the financial stock prediction. The models are trained on the AAPL, CSCO, IBM, and MSFT stocks, utilizing a combination of the financial stock data and sentiment extracted from social media posts on Twitter between the years 2015-2019. Our experimental results show that the combination of the financial and sentiment information can improve the stock market prediction performance. The proposed solution has achieved a prediction performance of 74.3%.

This thesis is dedicated to my parents.

For their endless love, support and encouragement.

Acknowledgment

First and foremost, I would like to thank my supervisor, Dr. Hussein Al Osman, for his continued support and guidance throughout my graduate studies, from sharing his deep knowledge in this field, to being a friend to talk to when needed. You have been very important in this journey, and I will forever be grateful for providing me with this opportunity.

I would also like to thank all my friends and colleagues at the University of Ottawa who have given me continuous support and encouragement throughout my two years of graduate studies at the University of Ottawa.

I would also like to express my deepest gratitude to the committee members, Dr. Liam Peyton and Dr. Morad Benyoucef for their individual comments and suggestions which have greatly improved my dissertation.

Finally, and most importantly, I would like to thank my family for their support, patience, and encouragement throughout these years. Without my family, none of this would ever be possible.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
ACIM	Apple, Cisco, IBM, Microsoft
ARIMA	Autoregressive Integrated Moving Average
AAPL	Apple
BPNN	Backpropagation Neural Network
BOW	Bag of Words
CNN	Convolutional Neural Network
CSCO	Cisco
DNN	Deep Neural Networks
DSRM	Design Science Research Methodology
ESE	Ensemble Sentiment Estimation Model
EMH	Efficient Market Hypothesis
ESM	Exponential Smoothing Model
ESN	Echo State Network
FS	Financial Stock Dataset
FSD	Financial Stock Data Model
GRU	Gated Recurrent Network
HAC	Hierarchical Agglomerative Clustering
HAK	Hierarchical Agglomerative Clustering and reverse K-means
HHMM	Hierarchical Hidden Markov Model
HRK	HAC, K-Means, Reverse K-Means
HTML	Hypertext Markup Language
IBM	International Business Machines Corporation
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percent Error
MCC	Matthews Correlation Coefficient
MLP FD	Multi-Layer Perceptron Feature Driven
MLP SWE	Multi-Layer Perceptron Feature Driven

ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
MSFT	Microsoft
NLP	Natural Language Processing
NSE	National Stock Exchange
POS	Part-of-Speech
PIP	Perceptually Important Point
PMI	Point-Wise Mutual Information
PHM	Proposed Hybrid Model
ReLU	Rectified Linear Activation Function
RNN	Recurrent Neural Networks
SFD	Sentiment and Financial Data Model
SET5	SemEval-2017 Task 5
SSE	Shanghai Stock Exchange
SVM	Support Vector Machines
TanH	Hyperbolic Tangent Function
TF-IDF	Term Frequency–Inverse Document Frequency
WWW	World Wide Web
XGBoost	eXtreme Gradient Boosting

Chapter 1: Introduction

Predicting the prices of stocks and identifying the financial variables that can influence them can be a challenging task, especially due to the volatile nature of the stock market (Kiersz, 2015). Studies have shown that the EMH (Efficient Market Hypothesis) theory and the random walk hypothesis of the stock market may be incorrect (Kiersz, 2015). The EMH theory states that the stock market does not follow a linear trend but can be very random, which impedes its prediction (Cao & Tay, 2000). As a result, researchers are constantly trying to find an accurate model that can predict the stock market (Fama, 1965). Warren Buffet has continuously been able to surpass the financial market especially the S&P 500 index fund which is a clear indication of predictability (Bezerra & Albuquerque, 2017). Developing a model that can predict the market and result in higher returns than expected will disprove the EMH theory and increase profits on financial investments.

The EMH theory does not explain the constant fluctuations in stock market pricing during short periods as it assumes that all information regarding a stock are factored into the price and no amount of analysis can allow an investor to predict the stock market (Thune, 2020). These fluctuations may be the result of investors buying or selling a certain stock due to news articles and social media reactions. Stock market prices do not follow constant trends but are rather dynamic and can be affected by news, social media, politics, investor emotions and the general economy (Bezerra & Albuquerque, 2017). Recent research has shown that online sources such as Twitter can be leveraged to determine the mood of investors and estimate the price of the Dow Jones Industrial Average (Chourmouziadis & Chatzoglou, 2016). The number of views on

Wikipedia's financial market page can be a good predictor of stock market trends. Investors are using these financial market pages to make decisions on buying and selling stocks (Hao, 2020).

In the past, predicting the stock market was done using average volume movements, distinct analysis, autoregressive models and correlations (Kumar & Thenmozhi, 2014). Recently, researchers began using artificial intelligence to recognize patterns and random samples of variables that affect the stock market behavior through deep learning and machine learning algorithms (Wang, Wang, Zhang, & Guo, 2012). Machine learning is used frequently in predictive models to analyze complex patterns. Due to the challenging nature of predicting the stock market, testing different machine learning algorithms may allow us to model the intricate patterns associated with the stock market behavior (Hsu, Lessmann, Sung, Ma, & Johnson, 2016).

The two main areas of focus in stock market research are verifying what variables need to be considered and what machine learning algorithm to be used for the prediction. Previous methods focused mostly on historical stock market price and stock specific variables without considering the investors mood on social media platforms (Ren, He, Girshick, & Sun, 2017). Using social media information to conduct sentiment analysis while still considering financial stock market price data, we can potentially improve the performance of machine learning methods that predict the stock market.

1.1 Objectives

Stock prices increase or decrease repeatedly when they are discussed and analyzed via news and social media outlets. Hence, in addition to considering historical trends, we intend to leverage the valuable sentiment information that can be retrieved from social media to gauge investors'

attitudes. The propensity of openly shared sentiments on social media platforms makes them convenient mechanisms to monitor investors' mood shifts. This will allow us to render informed insights regarding the stock market behavior. Our stock market estimator will use machine learning technology to produce its calculations. In this research, our objective is to build a state-of-the-art stock market prediction model, focusing on the short-term price trend prediction. Deep Neural Networks are an exceptional innovation in machine learning which we will utilize to predict these short-term price predictions. Such algorithms may identify patterns and trends that can go unnoticed by the human observer.

1.2 Methodology

Design research is a problem-solving technique for creating and evaluating information technology systems that meet business needs and solve organizational problems (Hevner, March, Park, & Ram, 2004). The Design Science Research Methodology (DSRM) provides a framework grounded in principles, practices, and procedures to address research questions in information systems (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007).

Even though DSRM was introduced over 20 years ago, the methodology continues to evolve. A summary of the DSRM guidelines are shown in Table 1. These guidelines help researchers, reviewers, editors, and readers to effectively apply DSRM (Hevner, March, Park, & Ram, 2004).

Table 1. Design Science Research Guidelines (Hevner, March, Park, & Ram, 2004).

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired

	ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Figure 1. shows the Design Science Research Methodology process that is conducted in six steps. In step one, we need to identify what our problem is and clarify our motivation to solve this problem. In step two, we define our solution's objectives. In step three, we design and develop our solution. In step four, we demonstrate the performance of our model using real life data that was previously seen by the model. In step five, we evaluate our solution against other scholarly publications in a similar field. Finally, in step six, we disseminate our finding through peer-reviewed publications.

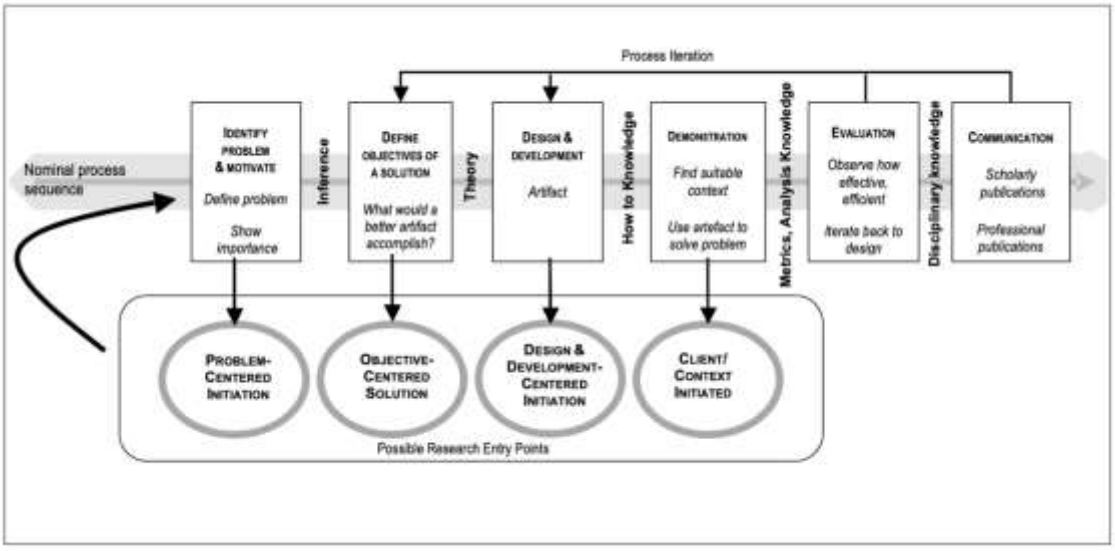

Figure 1. Design Science Research Methodology Process Model (Peffers, Tuunanen, Rothenberger & Chatterjee, 2007)

Following the DSRM process model shown in Figure 1, we start off by identifying the problem and motivation of our research. In this research, we intend to find a solution to predicting the stock market. Conventionally, changes in a stock's value were predicted using its past pricing trends. However, such historical information has a limited capacity to predict the stock market. Hence, we would like to realize a solution that augments financial stock data with social media sentiment information extracted from Twitter. In step two, we define our objectives and how we will find this solution. Our objective is to create a machine learning model that can take financial stock data and sentiment data from Twitter as an input, and produce predictions for stock prices. To demonstrate our solution, we restrict our inputs to four different stocks, namely AAPL, CSCO, IBM, and MSFT. We selected these stocks as they are widely discussed in social media posts. These four stocks are considered large cap stocks which have a market capitalization of 10 billion USD or more. Large cap stocks are less volatile during rough markets and represent most of the U.S equity market and are considered to be core investment stocks. Evidently, the proposed solution can be scaled to over additional stocks given that relevant information is collected. In the third phase, we design and develop our machine learning models. The development and design phase consists of data scraping from Yahoo Finance, and social media platforms, and preprocessing to prepare the data to be inputted into our models. Our models consist of Multi-Layer perceptron (MLP), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN) sub-models. We use two MLP models, the Multi-Layer perceptron Feature Driven (MLP FD), and Multi-Layer perceptron Simple Word Embedding (MLP SWE). The data scraping, preprocessing and modeling will be explained further in detail in chapter 3 and chapter 4. In steps 4 and 5, we evaluate the proposed model on a testing dataset and compare its performance to

existing state of the art research solutions. In step six, we communicate our results through a peer reviewed publication and thesis document to allow other researchers, reviewers, editors, and readers to further advance the work in this field.

1.1 Contributions of the Thesis

We summarize this thesis' contributions as follows:

- We designed and developed 7 machine learning (ML) models to predict the stock market:
    - Ensemble Sentiment Estimation (ESE) Models: MLP FD, MLP SWE, LSTM, CNN, MLP Stacked Ensemble
    - Financial Stock Data (FSD) Model: LSTM (2 Layers)
- We designed and developed an MLP network that fuses the stock market predictions from the ESE and FSD models.

1.4 Research Questions

The goal of our research is to build a machine learning model with the use of deep learning techniques to improve stock price estimation. To achieve this goal, we will use a combination of sentiment analysis from social media messages and financial stock data. Hence, we define two research questions:

1. Can we improve stock price predictions using sentiment analysis of social media messages compared to estimates solely based on financial data?
2. Can we achieve superior performance compared to the state-of-the-art methods in stock price estimation using deep learning techniques?

1.5 Thesis Organization

This thesis consists of five chapters. Chapter 2 presents a background about this research field, and reviews related research in the field of stock market prediction. Chapter 3 details the collection and preprocessing procedure of the social media and the financial stock data. Chapter 4 describes and evaluates the proposed models. Chapter 5 includes a summary of the research and potential future work directions.

Chapter 2: Literature Review and Background Study

In this section, we will discuss existing stock market prediction methods. We will focus on several approaches, such as machine learning, supervised learning, unsupervised learning, and hybrid techniques. We will discuss the current challenges in the field and what future research should address.

2.1 Machine Learning Algorithms

Machine learning algorithms are the integral component of artificial intelligence. They are trained to uncover patterns in data and estimate a function that can calculate outputs based on inputs for particular problems (Xiao, Xiao, Lu, & Wang, 2014). Machine learning models operate in three phases, the training of the algorithm, the validation of the results, and the testing of the algorithm, which is used to determine the algorithm's performance. They require data sets for each one of these phases.

Neural networks were originally designed to mimic the functioning of the human brain. However, modern architectures have veered considerably from this initial objective. The neurons of the network receive inputs and deliver outputs (Laboissiere, Fernandes, & Lage, 2015). They can communicate with different other neurons within the neural network. Connections between neurons are associated with weights that are learned during training.

Support Vector Machine (SVM) is another prominent machine learning approach that is often used for various classification tasks. The difference between neural networks and SVMs is that neural networks attempt to lower the errors during training, while SVMs attempt to lower the error of the upper threshold for classification (Huang, Nakamori, & Wang, 2005). SVM employs a

different method than neural networks. In SVM, the training data is transferred to a higher dimensional space to achieve a linear separation between the classes (Kara, Acar Boyacioglu, & Baykan, 2011).

Another machine learning approach uses decision trees to divide the data into different subsets starting with the inputs until a classification is made. Different decision trees can be combined to ensure better results especially when attempting to predict the stock market (Barak, Arjmand, & Ortobelli, 2017).

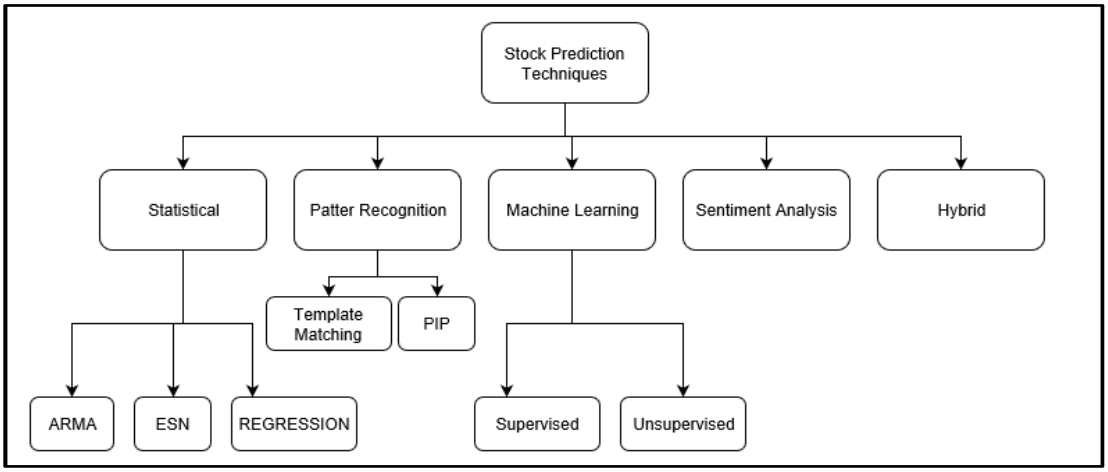

Figure 2. Stock Market Prediction techniques

2.1.1 Supervised Learning for Stock Market Prediction

SVMs and Decision Trees use supervised learning for training. Supervised learning refers to the process of providing a machine learning model with a training dataset that contains sample inputs and outputs during training. The model "learns" by estimating a function that maps the inputs to the outputs. This would allow the model to estimate outputs for inputs it has not seen during training. Supervised learning algorithms can predict the stock market using financial data and other valuable information that impact the stock price. Bernal et al. utilized S&P 500 variables,

such as the stock prices, averages, and volumes as input to an Echo State Network (ESN) to predict the stock market changes (Bernal, Fok, & Pidaparthi, 2012). ESNs are based on Recurrent Neural Networks. This technique achieved a test error of 0.0027 in the daily price change (Bernal, Fok, & Pidaparthi, 2012). The results were confirmed by testing the technique on 50 additional stock prices and reported that the results performed well against the state of the art techniques (Bernal, Fok, & Pidaparthi, 2012). Ballings et al. compared various ML approaches such as Random Forest, AdaBoost and Kernel Factory, Neural Networks, Logistic Regression, SVMs, and K-Nearest Neighbor to predict long-term stock prices. They found Random Forest to be the top performing algorithm (Ballings, Michel, Poel, Hespeels, & Gyrp, 2015).

Milosevic introduced a classification method to predict good and bad stocks. If the stock price increased 10% over a year, then it was considered good, otherwise, it was characterized as a bad stock (Milosevic, 2016). This method used 11 manually selected fundamental ratios as features for the selected machine learning algorithms (Milosevic, 2016). They tested several machine learning approaches, such as Random Forest, Support Vector Machine and Naïve Bayes (Milosevic, 2016). The Random Forest model produced the highest F-Score at 0.751 (Milosevic, 2016). Basak et al. investigated an eXtreme Gradient Boosting (XGBoost) method for long-term stock market prediction which achieved an F-Score between 0.82 and 0.97 for 60 day and 90 day periods (Basak, Kar, Saha, Khaidem, & Dey, 2019). The approach uses technical stock indicators as input to the model. The method was tested on two stocks, Apple and Yahoo.

Long Short-Term Memory (LSTM) networks have consistently rendered accurate predictions of stock market prices. Di Persio and Honchar compared the performance of three types of Recurrent Neural Networks (RNNs), a basic RNN, an LSTM and a Gated Recurrent Network

(GRU) (Di Persio & Honchar, 2017). Their results confirmed that the LSTM performed better than the other RNNs with a 72% accuracy (Di Persio & Honchar, 2017). Hossain et al. applied two Deep Neural Network (DNN) models, an LSTM and a GRU. They trained the models using the S&P 500 index for a dataset that spanned 66 years, from 1950 to 2016 (Hossain, Karim, Thulasiram, Bruce, & Wang, 2018). In their approach, the features go through the LSTM to produce a prediction. The prediction is then passed to the GRU model to produce another prediction. The proposed network achieved a Mean Square Error (MSE) of 0.00098 in prediction (Hossain, Karim, Thulasiram, Bruce, & Wang, 2018). This approach achieved better results than previous neural network models which achieved a MSE of 0.003 for both the GRU model and the LSTM models (Hossain, Karim, Thulasiram, Bruce, & Wang, 2018).

Lv et al. assessed the performance of traditional classifiers, namely, SVM, Logistic Regression, Random Forest, Naïve Bayes, Classification and Regression Tree, and XGBoost. Moreover, they studied the performance of DNN algorithms, namely MLP, Deep Belief network, Stacked Autoencoder, RNN, LSTM and GRU (Lv, Yuan, Li, & Xiang, 2019). They determined that traditional algorithms were able to outperform the advanced DNN models when transaction cost was not taken into consideration. When transaction cost was considered, the DNN models performed better (Lv, Yuan, Li, & Xiang, 2019).

In 2008, researchers used a combination of machine learning algorithms and a statistical approach to make stock market predictions (Markowska-Kaczmar & Dziedzic, 2008) .They applied a supervised feed-forward neural network algorithm with a combination of perceptually important points (PIP) technique to reduce the dimensionality and find the important points of the

patterns (Markowska-Kaczmar & Dziedzic, 2008). The results of the PIP technique were found to be fair, since it only found patterns for shortened time series data with an accuracy of 84% in properly classified series (Markowska-Kaczmar & Dziedzic, 2008).

Pang et al. developed the "stock vector" concept based on the progression of word vector in deep learning (Pang, Zhou, Wang, Lin, & Chang, 2020). The input is no longer a single index or single stock index, but multi-stock high-dimensional historical data. They proposed two models, a Long Short-Term Memory (LSTM) with an embedding layer and an LSTM with an autoencoder to predict the stock market (Pang, Zhou, Wang, Lin, & Chang, 2020). They used the embedding layer or the autoencoder to vectorize the data, in a bid to forecast the stock via the LSTM network (Pang, Zhou, Wang, Lin, & Chang, 2020). The experimental results show that the LSTM with an embedding layer performed slightly better than the one with the autoencoder. Specifically, the accuracies of the LSTM with an embedding layer and the LSTM with an autoencoder was 57.2% and 56.9%, respectively, for the Shanghai A-shares composite index (Pang, Zhou, Wang, Lin, & Chang, 2020). They achieved a price prediction accuracy of 52.4% and 52.5%, for the LSTM with an embedding layer and the LSTM with an autoencoder respectively, on the TMSE, TBEA, and SINOPEC stocks (Pang, Zhou, Wang, Lin, & Chang, 2020).

2.1.2 Unsupervised Learning for Stock Market Prediction

Due to the nature of the stock market, information is not interrelated, which can create challenges when applying supervised learning. Unsupervised learning however, can interrelate unrelated information. Powell et al. compared supervised learning (SVM) and an unsupervised learning (K-Means) (Powell, Foo, & Weatherspoon, 2008). Even though unsupervised learning

was expected to perform better, it was noted that both the SVM and K-Means models produced comparable performance (Powell, Foo, & Weatherspoon, 2008).

Babu et al. combined the Hierarchical Agglomerative Clustering (HAC) and reverse K-means clustering methods in an approach they refer to as HAK (Babu, Geethanjali, & Satyanarayana, 2012). They converted financial articles and stock price information into a feature vector using a text analyzer (Babu, Geethanjali, & Satyanarayana, 2012). Using HAC, the features are divided into clusters (Babu, Geethanjali, & Satyanarayana, 2012). Each cluster is partitioned and clustered into more sub-clusters. The centroids of the sub-clusters are calculated using the K-Means clustering method (Babu, Geethanjali, & Satyanarayana, 2012). For their assessment, they compared HAC, K-Means, Reverse K-Means, HAK, and the SVM (Babu, Geethanjali, & Satyanarayana, 2012). HAK outperformed SVM in terms of accuracy and average profits (Babu, Geethanjali, & Satyanarayana, 2012). The total average profit of HAK is 3.95%, while the total average profit of SVM is 1.46% (Babu, Geethanjali, & Satyanarayana, 2012).

Wu et al. introduced the AproriALL algorithm in combination with K-Means (Wu, Wu, & Lee, 2014). This model used a sliding window to convert stock information into charts (Wu, Wu, & Lee, 2014). Once these charts were created, they were then clustered using the K-Means algorithm, which allowed for the extraction of patterns in the chart (Wu, Wu, & Lee, 2014). AprioriALL was used to extract the common patterns to predict trends (Wu, Wu, & Lee, 2014). This model proved to perform better than similar related work with an average prediction accuracy of 57% (Wu, Wu, & Lee, 2014).

Peachavanish introduced a clustering method to predict the trends and momentum characteristics that permit the predictions of stock prices for short-term periods (Peachavanish, 2016). Peachavanish conducted an experiment on five years of financial stock data prices from the Stock Exchange of Thailand and reported that the proposed method can outperform the stock market in the long run (Peachavanish, 2016).

2.1.3 Sentiment Analysis

The assessment of sentiments for predicting the value of a stock or a company can drive short-term market fluctuations which in return can cause a disconnect between the stock price and the true value of a company. However, the true value of a stock can return to its original value once the true weights of the stock price kicks in. While sentiments can cause a disconnect between the true value and the stock price, it should always be part of the stock market analysis as it can provide insights into how the stock price reacts in the short, medium, and long term trading (Shah, Isah, & Zulkernine, 2019).

A recent study, examined a supervised machine learning method proposed to evaluate how a news report affects the stock price 20 minutes after its release (Schumaker & Chen, 2009). The approach investigated the use of three different textual feature types for sentiment analysis; The Bag-of-Words, the Noun Phrases, and the Named Entities (Schumaker & Chen, 2009). The closing prices of the last 60 minutes of the stocks were collected and passed through a Support Vector Regression model to predict the stock price in the following 20 minutes based on the stock prices and sentiment information collected (Schumaker & Chen, 2009). From the results, the researchers concluded that the performance was better than the linear regression model with a directional

accuracy of 57.1%, measures of closeness at 0.04261, and simulated trading at a 2.06% return when using news article terms and stock prices (Schumaker & Chen, 2009). The researchers also found that the textual method performed better at a directional accuracy of 58.2%, measures of closeness at 0.04433, and simulated trading at 2.84% when using news article terms and stock prices (Schumaker & Chen, 2009). The authors also stated that the Noun Phrases features performed much better than the Bag-of-Words and Named Entities features after testing was completed (Schumaker & Chen, 2009).

Bollen et al. were one of the original researchers to analyze Twitter data and use original techniques like the Google Profile of Mood States and Opinion Finder to predict trends in the Dow Jones Industrial Average (Bollen, Mao, & Zeng, 2011). Bollen et al. used public events such as the presidential election and Thanksgiving to measure public sentiments using the Google Profile of Mood States and Opinion Finder (Bollen, Mao, & Zeng, 2011). The researchers extracted approximately 10 million tweets between February and December 2008 and used closing prices of the Dow Jones Industrial Average between the same dates and applied it to the Self-Organizing Fuzzy Neural Network (Bollen, Mao, & Zeng, 2011). The performance and accuracy of this technique was 87.6% when used to predict the Dow Jones Industrial Average (Bollen, Mao, & Zeng, 2011).

Many sentiment analysis studies use the groundwork research conducted by (Bollen, Mao, & Zeng, 2011). Another research group implemented the same approach of (Bollen, Mao, & Zeng, 2011), but rather than using 10 million tweets, they used over 400 million tweets (Mittal & Goel, 2012). The collected Twitter posts contained data from weekends and holidays, which required replacement with the average value of the week (Mittal & Goel, 2012). A portfolio management

strategy was then applied to the work by (Bollen, Mao, & Zeng, 2011) to ensure steady prices and address the fluctuations in the stock price (Mittal & Goel, 2012). They omitted the data for the days that were associated with high stock volatility to improve their prediction accuracy (Mittal & Goel, 2012). Their technique also used four types of moods, happy, alert, calm, and kind. They obtained an accuracy of 75% and as a result were able to achieve decent profits over a testing period of 40 days. They discovered that calm and happiness gave similar results during their predictions, demonstrating that certain moods were harder to differentiate (Mittal & Goel, 2012).

Different text analysis methods were applied to 8-K reports to see how a stock price would change by either going up or down, or remaining neutral when 8-K reports were considered (Lee, Surdeanu, MacCartney, & Jurafsky, 2014). The model was first applied to the financial stock data, then on both the financial stock and the text data (Lee, Surdeanu, MacCartney, & Jurafsky, 2014). Using both the financial stock and text analysis improved the accuracy by 10% (Lee, Surdeanu, MacCartney, & Jurafsky, 2014). The research also states that the text analysis of the 8-K reports would only improve the short-term predictions rather than long-term predictions (Lee, Surdeanu, MacCartney, & Jurafsky, 2014).

News articles are an integral part of predicting the stock market. Combining news articles and machine learning can help prediction accuracy (Kalyanaraman, Kazi, Tondulkar, & Oswal, 2014). A proposed research model was to combine sentiment analysis of news articles and two machine learning algorithms, linear regression using gradient descent and linear regression using normal equation, to obtain accurate stock market predictions (Kalyanaraman, Kazi, Tondulkar, & Oswal, 2014). Two dictionaries were made to divide the positive and negative news due to the insufficient amount of open source dictionaries (Kalyanaraman, Kazi, Tondulkar, & Oswal, 2014). Once the

words in the articles were divided into positive or negative, they were supplied to a linear regression model (Kalyanaraman, Kazi, Tondulkar, & Oswal, 2014). Kalyanaraman et al. were able to obtain a prediction accuracy of 60% with linear regression using normal equation and 81.82% with linear regressing using gradient descent (Kalyanaraman, Kazi, Tondulkar, & Oswal, 2014).

In an alternate study, Cakra et al. attempted to predict the price, price changes, and margin percentage of Indonesian stocks using a simple sentiment analysis model combined with classification techniques, and a linear regression model (Cakra & Distiawan Trisedya, 2015). A tweet was either considered a positive, negative, or neutral tweet, but the neutral tweets were not further considered as they might not be related to the classified topic (Cakra & Distiawan Trisedya, 2015). The collection of tweets was separated into single words. The words that belong to the lexicon were chosen for analysis, while the other words that were checked manually to assess their similarity to words from the lexicon (Cakra & Distiawan Trisedya, 2015). Once the word check was completed and categorized as either lexicon or non-lexicon, they were classified as positive or negative words (Cakra & Distiawan Trisedya, 2015). Based on the machine learning algorithms tests, the researchers were able to get a prediction accuracy of 60.39% using the Random Forest model (Cakra & Distiawan Trisedya, 2015). While on the prediction accuracy of the price changes, the researchers achieved an accuracy of 67.37% using the Naïve Bayes algorithm and 66.34% when using the Random Forest algorithm (Cakra & Distiawan Trisedya, 2015).

Other than news article sentiment analysis, Twitter is also considered an important tool for predicting the stock market (Pagolu, Reddy, Panda, & Majhi, 2016). N-Gram and Word2Vec, which is a 2-layer neural network, were used to analyze the polarity of sentiments behind the tweets

(Pagolu, Reddy, Panda, & Majhi, 2016). The researchers achieved an accuracy of around 70% and noted that the correlation between price and sentiments was 71.82% (Pagolu, Reddy, Panda, & Majhi, 2016). The researchers showed that the accuracy improved as the size of the dataset increased (Pagolu, Reddy, Panda, & Majhi, 2016). Xu and Cohen introduced StockNet, a neural network model that uses neural variational inference to address the intractable posterior inference (Xu & Cohen, 2018). Xu et al. also provide a hybrid objective with temporal auxiliary to flexibly capture predictive dependencies (Xu & Cohen, 2018). The researchers achieved a performance accuracy of 58.23% and a Matthews Correlation Coefficient (MCC) of 0.080796 (Xu & Cohen, 2018).

2.1.4 Multiple Approaches

Aside from supervised and unsupervised learning, there are various other methods to predict stock prices which use a combination of approaches to improve performance. In 2006, Tiwari et al. combined two methods, the statistical Hierarchical Hidden Markov model (HHMM) with a supervised learning technique using decision trees to predict the Bombay Stock Exchange based on the historical closing prices, dividends and earnings of a stock (Tiwari, Pandit, & Richhariya, 2006). After the features were extracted from the dataset, the Decision Tree model selects the relevant features, a set-based classifier was used for the prediction, and the HHMM model assessed the predictions and generated a final prediction (Tiwari, Pandit, & Richhariya, 2006). This technique resulted in a 92.1 percent prediction accuracy for stock market prices (Tiwari, Pandit, & Richhariya, 2006).

In the study completed by Ding et al. the authors proposed a solution that combines both sentiment analysis and neural network models for the prediction of the S&P 500 index (Ding, Zhang, & Liu, 2015). A deep Convolutional Neural Network model was trained to determine the short and long-term effects of different news events and their influence on the index. Results showed that a directional accuracy rate of up to 64.21% is achieved on the S&P 500 index price prediction and up to 65.48% is achieved on the individual stock price prediction (Ding, Zhang, & Liu, 2015).

Rather et al. developed a hybrid solution for financial variable prediction that uses both linear and non-linear models for the prediction of the National Stock Exchange (NSE) stock prices (Rather, Agarwal, & Sastry, 2015). Recurrent Neural Network (RNN), Echo State Network (ESN), and Autoregressive Integrated Moving Average (ARIMA) are among the models combined in the solution (Rather, Agarwal, & Sastry, 2015). A weighting approach was used on the three models to determine the prediction result effect with the aid of a genetic algorithm (Rather, Agarwal, & Sastry, 2015). The proposed hybrid solution achieves a Mean Absolute Error (MAE) of 0.0009 and a Mean Squared Error (MSE) of 0.0127 (Rather, Agarwal, & Sastry, 2015). Moreover, the hybrid solution achieved higher prediction rates in comparison to RNN (Rather, Agarwal, & Sastry, 2015).

Another hybrid solution was developed by Wang et al., that combines exponential smoothing model (ESM), ARIMA, and a Backpropagation Neural Network (BPNN) model (Wang, Wang, Zhang, & Guo, 2012). They refer to their solution as the Proposed Hybrid Model (PHM), which benefits from the strengths of each of the three sub-models to predict weekly financial variables

(Wang, Wang, Zhang, & Guo, 2012). Results show that PHM outperforms each individual sub-model and other traditional solutions, with a directional accuracy rate of up to 70.16% when tested on the Shenzhen Integrated Index and DIJA (Wang, Wang, Zhang, & Guo, 2012).

An extended work of Wang et al. was developed by Creighton et al. The work applies a hybrid model approach to predict daily stock prices with different indices such as the S&P 400 index and the S&P 500 index (Creighton & Zulkernine, 2017). Although results showed that the proposed hybrid solution did not outperform individual sub-models for daily predictions, the study provided insight into sub-model accuracy. BPNN has showed to provide the most accurate financial variable prediction with a directional accuracy of 75.5%, while PHM, ESM and ARIMA provide better variable predictions for longer time ranges and suffer under daily predictions (Creighton & Zulkernine, 2017). PHM, ESM, and ARIMA achieved a directional accuracy of 45.1%, 47.4%, and 44.9% respectively (Creighton & Zulkernine, 2017).

A hybrid model that combines a deep learning approach with a sentiment analysis model for stock price prediction was proposed by Jing et al. They employed a Convolutional Neural Network model for classifying the investors' hidden sentiments, by extracting data from a major stock forum (Jing, Wu, & Wang, 2021). Jing et al. proposed a hybrid research model by applying the Long Short-Term Memory (LSTM) Neural Network approach for analyzing the technical indicators from the stock market and the extracted sentiment results (Jing, Wu, & Wang, 2021). This work was conducted using real-life experiments from six key industries of three time intervals on the Shanghai Stock Exchange (SSE) to validate the effectiveness and applicability of the proposed model (Jing, Wu, & Wang, 2021). The experiment results indicated that the proposed model

achieved superior performance when classifying investor sentiments than the baseline classifiers (Jing, Wu, & Wang, 2021). This hybrid approach performed better in predicting stock prices compared to the single model and the models that didn't use sentiment analysis (Jing, Wu, & Wang, 2021). The proposed hybrid model by Jing et al. achieved an average MAPE of 0.0449 (Jing, Wu, & Wang, 2021).

2.2 Summary of Related Work

We summarize the related work we surveyed in this chapter in Table 5. We highlight whether each approach used financial, social media, and/or news data for stock market prediction. Moreover, we specify the ML approach adopted.

Table 2. Different Techniques Used in Stock Market Prediction

Paper	Type of Source			Machine Learning Approach
	Financial Data	**Social Media**	**News**	
Bernal et al. (2012)	x			ESS, RNN
Milosevic et al. (2016)	x			Random Forest, SVM, NB, Logistic Regression
Dey et al. (2016)	x			XGBoost, SVM, ANN
Di Persio et al. (2017)	x			RNN, LSTM, GRU
Tiwari et al. (2010)	x			HHMM, Decision Trees
Zhang et al. (2018)	x			Random Forest
Schumaker and Chen (2009)	x		x	SVM
Bollen et al. (2011)	x	x		SVM
Lee et al. (2014)	x		x	Random Forest
Kalyanaraman et al. (2014)			x	Linear Regression

Pagolu et al. (2016)	x	x		Random Forest
Hossain et al. (2018)	x			DNN, LSTM, GRU
Lv et al. (2019)	x			SVM, Logistic Regression, Random Forest, Naïve Bayes, Classification and Regression Tree, and XGBoost VS DNN
Powell et al. (2008)	x			SVM vs K-means
Babu et al. (2012)	x		x	HAC and K-means
Wu et al. (2014)	x			AproriALL algorithm in combination with K-Means
Peachavanish (2016)	x			Clustering Method
Mittal et al. (2012)	x	x		SVM
Cakra et al. (2015)	x	x		Random Forest
Xu et al. (2018)	x	x		Neural Network
Markowska-Kaczmar et al. (2008)	x			Supervised Feed Forward Neural Network
Ding et al. (2015)	x		x	CNN
Rather etl al. (2015)	x			RNN, ESN, ARIMA
Wang et al. (2012)	x			BPNN, ESM, ARIMA
Creighton et al. (2017)	x			BPNN, ESM, ARIMA
Pang et al. (2020)	x			LSTM
Jing et al. (2021)	x	x		CNN, LSTM

2.3 Challenges and Further Research

The complex factors that impact stock market can create challenges in daily trades. The constant publication of new financial data, news articles, and social media messages on the internet can influence the stock prices. The implementation of new algorithms for stock prediction may impact the way investments are made. For instance, the implementation of algorithms for stock market prediction decreases the need for a company to invest time to manually predict the financial stock market data. As a result, the cost for large corporations decreases, while small investors may not be able to afford to use the same algorithms due to their high cost of research and development. These algorithms can influence when an investor sells or purchases stocks due to triggered panic, which makes stock market predictions more difficult even with the use of algorithms.

When it comes to stock market prediction algorithm design, the challenges are often associated with improving performance. Once the performance objectives are achieved, companies often keep these algorithms private to avoid reducing their value. In fact, most algorithms used by large corporations are confidential; they are never given to the public and have never been published.

Conventionally, automated stock market prediction research has been focused on short-term rather than long-term stock prediction. The ML-based predictors typically use Artificial Neural Networks, Long Short-Term Memory and Recurrent Neural Networks. Some of these methods are currently being applied to long-term predictions.

Most current prediction algorithms rely solely on financial data. Hence, the integration of additional influential factors into the forecasting equation may result in vast improvements. When real-time data is collected, various factors must be considered such as price changes, events that

change price, and other anomalies that have an influence on price. In particular, modern prediction approaches should consider news articles and social media data due to their almost instantaneous and influential impact on the stock market. However, considering these sources of data has its drawbacks. Social media and news articles contain false content that can influence stock prices resulting in a spike that can create panic amongst investors to either buy or sell. Perhaps such sources of data can be supplemented with the company's official quarterly and annual reports to increase the robustness of the system.

2.5 Chapter Summary

Stock market trading is accessible to almost anyone around the world; However, ease of accessibility can result in positive and negative associated risks. Online trading has changed the way investors trade on the stock market creating an avenue for researchers and data scientists to investigate methods for lessening the risk of trading and potentially increasing profitability. This study and literature review examined the various research methods used to predict the stock market using machine learning algorithms and sentiment analysis. The chapter focused on the challenges currently in the field and examines future possibilities for stock market prediction. By incorporating machine learning algorithms into stock market forecasting, the risks associated with trading can be decreased. As a result, focus can shift from predominately short-term trading to increased long-term trading, through improved accuracy of stock market predictions.

Chapter 3: Data Specifics

The focus of this chapter is to provide an in-depth analysis of the datasets we used to train and test the models proposed in this thesis. Specifics regarding the machine learning models used will be discussed in Chapter 4. This chapter is divided into 4 sections. Section 3.1 focuses on the social media dataset and details regarding data cleaning and feature extraction; Section 3.2 focuses on data feature selection of the social media dataset; Section 3.3 discusses the technique used for dataset splitting and cross validation of the social media dataset; and Section 3.4 discusses the details of the financial stock (FS) dataset.

3.1 The Social Media Datasets

To predict stock market trends, we used a combination of financial stock and social media data. The social media data will be used to extract the emotions expressed by retail investors that purchase stocks in the market.

The Ensemble Sentiment Estimation (ESE) model consists of two datasets: SemEval-2017 Task 5 (SET5)[1] and ACIM. We use the SET5 dataset to train, validate, and test our ESE model. The ESE model is responsible for estimating the sentiment in a social media message. SET5 consists of training and test sets. The dataset comprises a collection of financially relevant microblog messages which have been annotated with a sentiment score by financial experts. Each message in the SET5 training and test sets is annotated with the following information: source (social media platform where the message was posted), id (unique social media id of the message

[1] (https://bitbucket.org/ssix-project/semeval-2017-task-5-subtask-1/src/master/)

posted), cashtag (stock symbol), sentiment (a floating-point value between -1 and +1, where -1 corresponds to most negative, 0 to neutral, and +1 to most positive sentiment), and spans (list of strings that contain the sentiment).

We use the ACIM dataset to test the proposed stock prediction solution. ACIM is a large dataset that we scraped using the official Twitter API to retrieve tweets regarding four stocks: $AAPL, $CSCO, $IBM, and $MSFT from 2015-01-01 and 2019-12-31. This dataset ended on 2019-12-31 due to the stock market volatility caused by the COVID-19 pandemic. We extracted this dataset using the specific company stock symbols (cashtag). We collected the ACIM data using the official Twitter API which thoroughly searches the Twitter platform to retrieve messages mentioning these stocks. The messages in this dataset will be used as the input into the trained ESE model to retrieve a sentiment score. The sentiment score will contribute to the prediction of the stock prices. We will further describe our solution in Chapter 4.

3.1.1 Social Media Data Preparation and Feature Engineering

The social media datasets (SET5 and ACIM) required data cleaning. We conducted feature engineering and selection prior to inputting any information into the ESE model.

Given that we use multiple machine learning models to estimate sentiment (as we discuss in Chapter 4), the social media data preparation procedure differed depending on the type of model used. Some of the models (i.e. CNN and LSTM models) required a data cleaning procedure (see Section 3.1.2). The message is then fed into the pre-trained Google Word2Vec[2] model to map it

[2] https://code.google.com/archive/p/word2vec/

into its vector representation. We feed a maximum of 50 words at a time. Messages that are smaller than 50 words are padded.

Conversely, in the other machine learning models used (i.e. MLP), the data preparation procedure consists of creating numeric features from the text. The ESE model considers numerous features such as the time in which the social media message was created, total likes, the number of capital letters, and counting hashtags. In fact, 64 features are taken from the initial data and are listed in Appendix B for reference. We extract three types of features, namely, stock, linguistic, and sentiment lexicon are used to help consolidate the data. Each is described in the following sub-sections.

Stock Features

The stock features consist of 20 binary features that are extracted from the social media data. Binary data is discrete data that can be in only one of two categories, either yes or no, 1 or 0, off or on, etc. Binary data can be thought of as a special case of ordinal, nominal, count, or interval data. Binary data is a very common outcome variable in machine learning classification problems (Needell, Saab, & Woolf, 2018). The list of binary features are '+number%' (represents a positive percentage, such as +25% which can mean the stock will rise by +25%), '-number%' (represents a negative percentage, such as -25% which means the stock will decrease by -25%), '$number' (represents a dollar amount), 'number% (represents a percentage, such as 10%) , ordinal number (represents a number, such as 1st), etc. The complete list of binary features can be found listed in Appendix B for reference.

Linguistic Features

The linguistic features consist of 22 features which include: *i)* seven integer linguistic features, *i)* six Part-of-Speech (POS) tags, *ii)* one pointwise mutual information, *iv)* four Term Frequency-Inverse Documentation Frequency (TF-IDF) n-gram features, and *v)* four Relevance Frequency (RF) n-gram features. We used the Stanford Natural Language Processing (NLP) python package for the extraction of the POS tag (Qi, Dozat, Zhang, & Manning, 2018).

i) Integer Linguistic Features

There are various factors that affect the sentiment level expressed in textual comments. Capitalization of letters tends to mark something for attention. While repeating letters tends to strengthen the emotion (Parlar, Özel, & Song, 2019). Exclamation marks can also increase or decrease the strength of the sentiment expressed in a text (Parlar, Özel, & Song, 2019). It is intended to indicate strong feelings and convey emotion, as well as indicate shouting or high volume (Parlar, Özel, & Song, 2019). Punctuation marks are important in sentiment analysis, as they can be used to express sentiments (Parlar, Özel, & Song, 2019). Whether one punctuation mark is used or a continuous stream of punctuations, they can indicate different types of emotions (Parlar, Özel, & Song, 2019). The list of integer features in this category includes the number of '!', '?', '$', continuous '!', continuous '?', capitalized words, and hashtags.

ii) Part-of-Speech (POS) Tags

To implement the POS tags, we used the Stanford NLP python package[3]. The objective of POS tagging is to identify the grammatical group of a given word. Whether it is a noun, pronoun, adjective, verb, adverb, etc. based on the context. POS tagging looks for relationships within the sentence and assigns a corresponding tag to the word. Bag of Words (BOW) is a representation of text that describes the occurrence of words within a text. We keep track of word counts and disregard the grammatical details and the word order. The BOW term uses "bag" to refer to the removal of any information regarding the order or structure of a word. Furthermore, the model is only concerned whether known words occur in the text as opposed to where they occur in the text. We extract verbs with the corresponding POS tags 'VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ', and 'VM' [4] from the text as verb features with the BOW form to record the frequency of each tag since verbs contain more subjective tendencies.

iii) Pointwise Mutual Information (PMI)

Pointwise Mutual Information (PMI) is a measure of association used in information theory and statistics. It measures the correlation between two events and is considered a very useful metric in the identification of events that frequently occur simultaneously. PMI can be used in machine learning models to estimate sentiment. The semantic orientation of a word can be measured if a word co-occurs with words that are clearly positive or clearly negative (Turney & Littman, 2003). The co-occurrence strength of words can be measured using PMI which is

[3] https://stanfordnlp.github.io/stanfordnlp/
[4] https://www.nltk.org/book/ch05.html

then averaged over a set of positive and negative words to retrieve a sentiment score (Turney & Littman, 2003).

iv) Term Frequency-Inverse Documentation Frequency (TF-IDF) N-gram

TF-IDF is a statistical measure used to evaluate the relevancy of a word in a document to a collection of documents. To obtain this statistical measure, two metrics must be multiplied, namely, the number of times a word appears in a document, and the inverse document frequency of the word across a set of documents. TF-IDF can be used in many areas, but more importantly it is used for text analysis and the scoring of words in ML algorithms for NLP. This statistical measure was invented to search documents and retrieve information. TF-IDF proportionally increases every time a word appears but is offset if that word is very common. For example, if the words “this”, “what”, and “if” are very common in a document, then they are ranked lower, given that they are not important to the document since they appear frequently in other documents in the corpus. However, if the word appears several times in one document while rarely appearing in another, then that means this word is important and relevant to the document.

To generate TF-IDF N-grams, we must first filter the social media messages. N-grams of texts are extensively used in NLP tasks. They are a set of co-occurring letters, syllables, or words within a given window. Hence, the text is subdivided into overlapping windows of N length (Ganesan, 2020). TF-IDF is then calculated for these windows as opposed to individual words. Examples of the filters needed to generate N-grams are: *i)* removing words that have a

length of less than two characters from the sentence, *ii)* removing hashtags and cashtags, *iii)* removing words with a number, and *iv)* removing unnecessary white spaces.

v) Relevance Frequency (RF) N-gram

RF N-gram is slightly different from N-gram. N-grams are a sequence of tokens (sequence of words) where each token is represented by 1 to represent the occurrence of the word (Lan, Tan, Su, & Lu, 2009). RF N-gram counts the number of occurrences of each token in the positive and negative texts of the training dataset. (Lan, Tan, Su, & Lu, 2009). Once the occurrences have been calculated, we then calculate the weight of each token in unigram, bigram, and trigram using Equation (1).

The difference between the RF N-gram and N-gram is that N-gram features (N = {1, 2, 3}, i.e., unigram, bigram, trigram features) use 1 to represent the occurrence of the words, while RF N-gram use the corresponding RF weight (Lan, Tan, Su, & Lu, 2009). Similar to N-grams, three types of RF N-grams are extracted, namely, unigram, bigram, and trigram. For each token, the weight of the three N-gram types is calculated as shown in Equation (1).

$$rf = max\left(ln\,(2 + \frac{a}{max(1,c)}),\ ln\,(2 + \frac{c}{max(1,a)})\right) \qquad (1)$$

where a is the number of texts in the positive category and c is the number of texts in the negative category (Lan, Tan, Su, & Lu, 2009). RF N-grams can be more effective than N-grams because it endows each word with a weight, which can capture the contribution of that word to the sentiment analysis of the sentence (Lan, Tan, Su, & Lu, 2009).

Sentiment Lexicon Features

Sentiment Lexicons have been known to be decisive features in sentiment analysis tasks. We obtained the sentiment lexicon features using publicly available sentiment lexicons. These publicly available sentiment lexicons are *i)* AFINN (Nielsen, 2011), *ii)* Bing Liu opinion Lexicon (Chourmouziadis & Chatzoglou, 2016), *iii)* NRC Hashtag Sentiment Lexicon (Kiritchenko, Zhu, & Mohammad, 2014), *iv)* General Inquirer Lexicon, and *v)* SentiWordNet (Baccianella, Esuli & Sebastiani, 2010). The publicly available sentiment lexicons are based on single words which contain many English words which have been assigned a score of a positive or a negative sentiment. For each word in the text, five sentiment lexicons are calculated, which are the ratio of positive words, the ratio of negative words, maximum sentiment score, minimum sentiment score, and the sum of the sentiment scores. Examples of the sentiment lexicon features extracted are AFINN sum score, AFINN maximum score, Bing Liu ratio of positive words, General Inquirer ratio of negative words, etc. The complete list of 19 sentiment lexicon features extracted can be found in Appendix B for reference.

i) AFINN

AFINN, which was developed by Finn Arup Nielson, is a simple and popular publicly available sentiment lexicon that contains over 3300 words used for sentiment analysis (Nielsen, 2011). Each of the 3300 plus words contains a polarity score. Words scores range from '-5' (negative) to '+5' (positive) (Nielsen, 2011). This simple and popular lexicon has a built-in library in Python to easily retrieve sentiment scores.

ii) Bing Liu Sentiment Lexicon

The Bing Liu sentiment lexicon is a list of 6800 English words of positive and negative opinion of sentiment words (Hu and Liu, 2004). These words are drawn from product reviews which were labelled using the bootstrapping method that uses WordNet adjective synsets and their antonyms (Hu and Liu, 2004). The list contains about 2006 positive words and 4783 negative words (Hu and Liu, 2004).

iii) National Research Council Canada (NRC) Hashtag Sentiment Lexicon

NRC Hashtag Sentiment Lexicon is a collection of seven lexicons which includes the popular Word-Emotion Association Lexicon. These lexicons can be used in sentiment analysis, product marketing, consumer behavior analysis, and political campaign analysis (Kiritchenko, Zhu, & Mohammad, 2014). Every lexicon has a list of words and are associated to categories such as emotions, sentiment (positive and negative), or color (Kiritchenko, Zhu, & Mohammad, 2014). These lexicons are for English words and are used for the analysis of English texts (Kiritchenko, Zhu, & Mohammad, 2014).

iv) General Inquirer Lexicon

The General Inquirer Lexicon word-list is one of the most popular lexicons used by a number of financial researchers in sentiment computation from given texts. It contains a database of 2000 positive words and 2000 negative words (Stone, Ogilvie, & Daniel, 2007).

v) SentiWordNet

SentiWordNet is an opinion lexicon derived from the WordNet database (Baccianella, Esuli & Sebastiani, 2010). Each term in the WordNet database is associated with a score that

indicates whether it is a positive or negative sentiment (Baccianella, Esuli & Sebastiani, 2010). The WordNet database contains tens of thousands of words, their meanings, POS representation, and the numerical score representation that ranges from 0 to 1 (Baccianella, Esuli & Sebastiani, 2010). Terms are organized according to semantic relations or meanings. These words are all grouped by their synonyms which are called synsets (Baccianella, Esuli & Sebastiani, 2010).

3.1.2 Social Media Data Cleaning

To attain a dataset that is useful for feature extraction and the learning process, we perform several steps to clean the initial social media data. The data is encoded using Hypertext Markup Language (HTML). As such, we remove all HTML encodings. For instance, we eliminate HTML characters such as &amp and $quot. We remove data regarding the user ID and other similar sensitive information within the messages. We eliminate retweet symbols within messages. We replace the Uniform Resource Locators (URLs) with the keyword '_url'. Similarly, we pad the Hypertext Transfer Protocol (HTTP) and World Wide Web (WWW) to a URL. A plethora of slang keywords and abbreviations need to abide by a unified syntax and semantics. As such, we correct abbreviations according to a slang dictionary, namely, the Natural Language Processing (NLP) dictionary[5]. We eliminate common word contractions to ensure a unified syntax. For instance, we replace the word 'I've' with 'I have'. Spelling mistakes are common in social media. Therefore, we incorporate the tedious process of spelling mistake correction as part of data cleansing. We

[5] https://github.com/coastalcph/cs_sst/blob/master/data/res/emnlp_dict.txt

correct word elongation and exaggeration such as ‘victoryyyyy’, which represent ‘victory’, and ‘callendar’ to represent ‘calendar’. Other than words, we clean numeric values and symbols. As such, we verify numeric value types, such as float and whole numbers. We join (i.e. pad) symbols with their respective words. For instance, we correct ‘$ AAPL’ to ‘$AAPL’. We remove unnecessary apostrophes and punctuation splits from the last word in a sentence. Finally, we convert ordinal words into their ordinal number counterpart. For instance, we modify the keywords ‘1st’ or ‘#1’ to the word ‘first’. A full listing and summary of the necessary data cleaning procedure is described in Appendix A.

3.1.3 Social Media Word Embedding Vector Representation

As part of the data preparation, we must transform the cleaned data into a word embedding vector representation. This was performed using Google News Word2Vec[6] pre-trained model, which allowed a word to be transformed into a 300-dimensional vector. A feature that is important to some learning models, as some models can only take inputs with constant dimensions; using the average pooling method, a 300-dimensional vector was created for each text. Average pooling allows down-sampling, which divides each input into rectangular pooling regions. Each rectangular pooling region was calculated with an average value (Gholamalinezhad & Khosravi, 2020).

[6] https://code.google.com/archive/p/word2vec/

3.2 Social Media Data Feature Selection

Once the feature engineering was completed, we then analyzed the features and selected which features will be inputted into the model. We used four different factors to determine which data was acceptable for use in the model, namely, *i)* missing value analysis, *ii)* constant variable analysis, *iii)* duplicated variable analysis, and *iv)* correlated variable analysis. In total, 64 features were created in the process during the feature engineering. After the analysis, only 52 features remained.

3.2.1 Missing Value Analysis

A function was performed to verify missing values within the data. One of the first steps in data cleaning is to check for missing values. This is because missing data in our dataset can reduce the power of our model or can lead to a biased model. In Appendix C, we describe our analysis of features with missing values.

3.2.2 Constant Variable Analysis

A class was created to verify the constant features in the numerical and categorical columns of the dataset to remove the constant features that were 99% ($\pm$0.01 tolerance) constant. Those features that contain constant values do not provide any information. These features are considered redundant data and their presence in the dataset has no effect on the target. After running this function on the dataset, the features that were removed are: 'POS_VM', '$num', 'num/num/num', '-num%', 'general_inquirer_pos_ratio', and 'general_inquirer_neg_ratio'.

3.2.3 Duplicated Variable Analysis

A function was created to ensure that all columns in the dataset do not contain any duplicates. If duplicates were found, the duplicate features were removed. Duplicated features are identical features, regardless of the variable or column name. If they show the same values for every observation, then they are considered duplicated. After running this analysis on the data, the results indicated that all the data was unique in the dataset.

3.2.4 Correlated Variable Analysis

In addition to the duplicate features, a dataset can also contain correlated features. A function was created to find correlated features within the data. Two features are considered correlated if they are close to each other in the linear space. Correlation between the output observations and the input features is very important and such features should be retained. However, if two or more features are mutually correlated, they convey redundant information to the model and hence only one of the correlated features should be retained to reduce the number of features. After implementing the function, the output showed that only two features were correlated in the dataset, namely, ‘conversation_replies’ and ‘total_likes’, with a correlation factor of 0.991888. As a result, the ‘conversation_replies’ feature was removed from the dataset.

3.3 SET5 Dataset Split and Cross Validation Technique

Once we collected, prepared, and cleaned the social media data and performed feature engineering and selection, there were 52 features left for input into the ESE model.

Next, we split the SET5 dataset into training (70%), validation (20%), and test (10%) sets as seen in Figure 3.

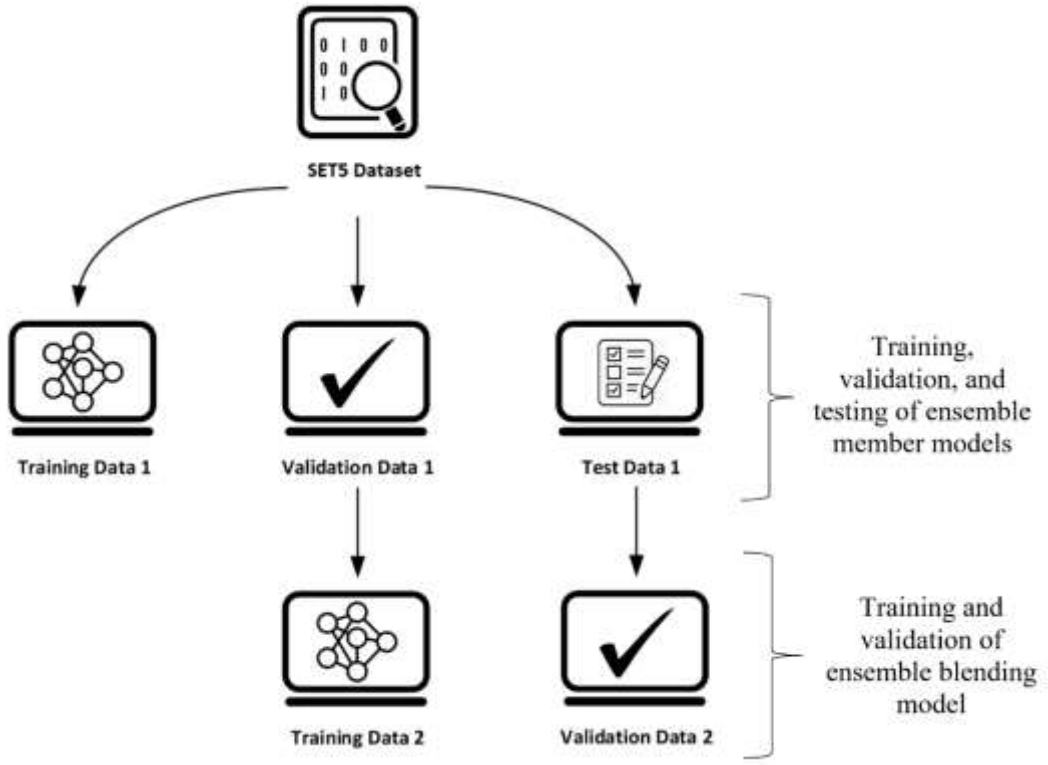

Figure 3. SET5 Dataset Split

As we will explain in Chapter 4, we use an ensemble scheme to estimate the sentiment. The ensemble is composed of four-member models (weak learners). We use a blending model to combine the results from the member models.

All member models are trained and validated using the 'Training Data 1' set through the stratified 10-fold cross-validation. Applying stratified sampling in cross-validation ensures the training and test sets have the same proportion of the feature of interest as in the original dataset. Stratified k-fold cross validation is similar to the regular k-fold cross validation, where the ratio between the target classes are similar in each fold, as it is in the complete dataset. The member models are then tested on the 'Validation Data 1', and 'Test Data 1' sets. The blending model is

then trained on the predictions of the member models for the ‘Training Data 2’ (i.e. ‘Validation Data 1’). It is validated on the predictions of the member models for the ‘Validation Data 2’ (i.e. ‘Testing Data 1’).

3.4 Collection of the Financial Stock Dataset

The Financial Stock (FS) dataset scraping requires less effort compared to the social media dataset collection. We used the Python package ‘yfinance’ to collect the data from the yahoo finance library using date ranges and ticker values to collect all the valuable FS data for the model.

For this analysis, the tickers used to perform the FS dataset scraping were $AAPL, $CSCO, $IBM, and $MSFT. The selected start date was 2015-01-01 and end date was 2019-12-31. This dataset ended on 2019-12-31 because of the volatility of the stock market caused by the COVID-19 pandemic. The FS dataset contains 28,224 records combined for the four years collected of the four selected stocks.

3.4.1 FS Dataset Exploration

To provide a summary of FS dataset, we created plots for each of the four stocks selected in this analysis. A consistent date range was selected so that the trends could be compared over the same time frame.

The plots for the $AAPL stock are shown below with a stock price history (Figure 4) and a volume stock history (Figure 5) summary.

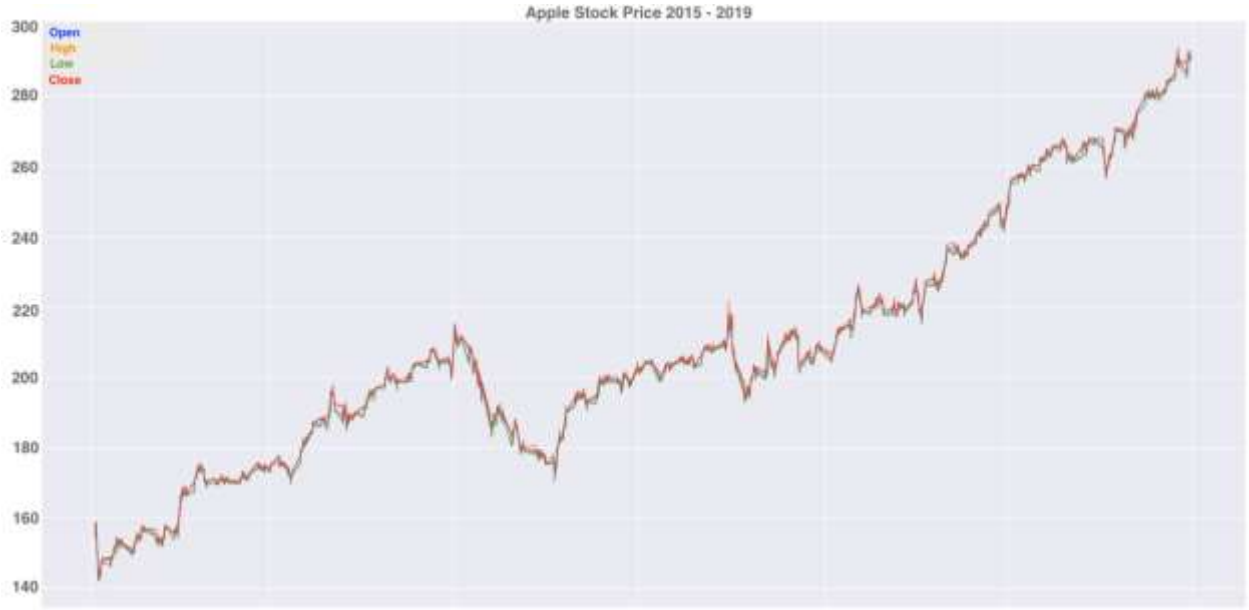

Figure 4. AAPL Stock Price History 2015-2019

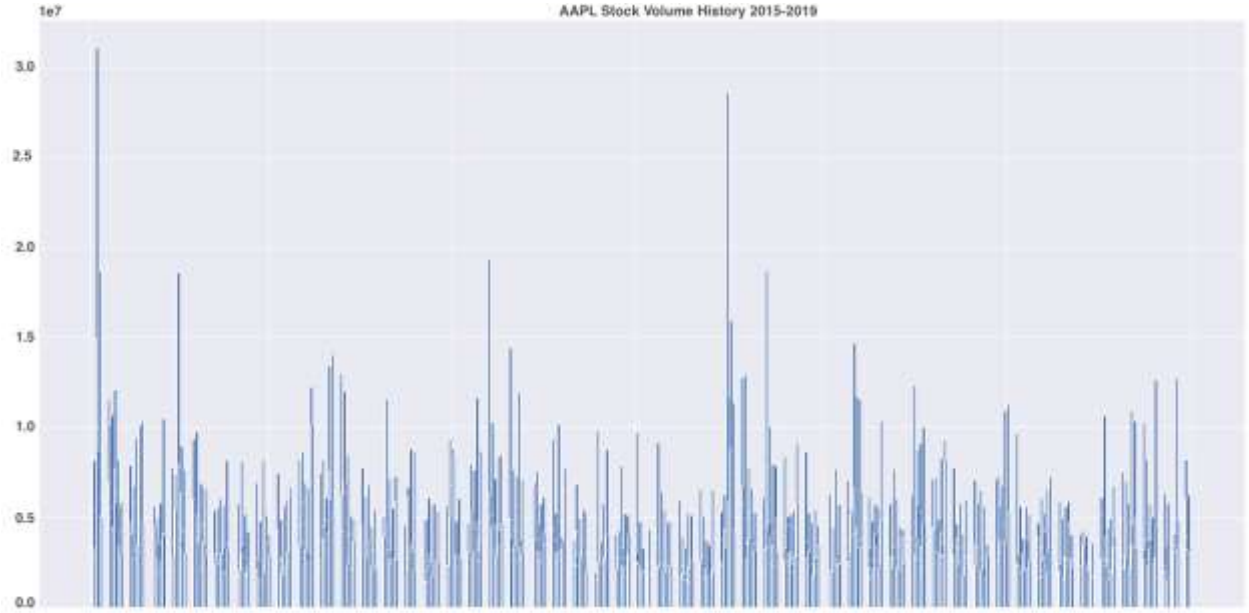

Figure 5. AAPL Stock Volume History 2015-2019

The plots for $CSCO stock are shown below with a stock price history (Figure 6) and a volume stock history (Figure 7) summary.

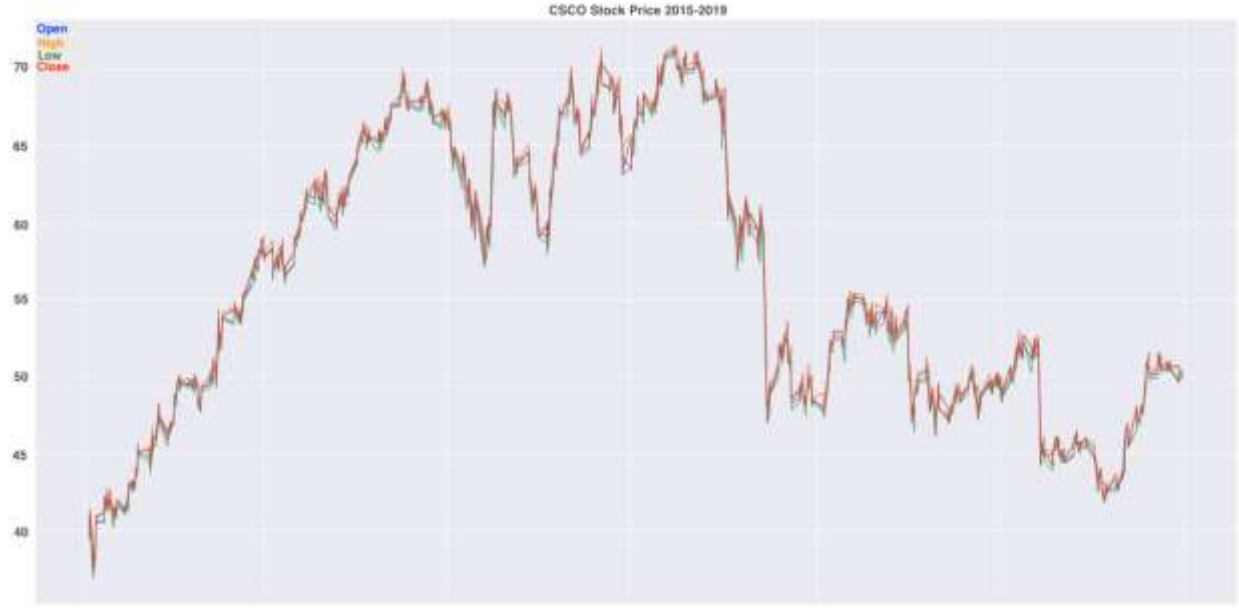

Figure 6. CSCO Stock Price History 2015-2019

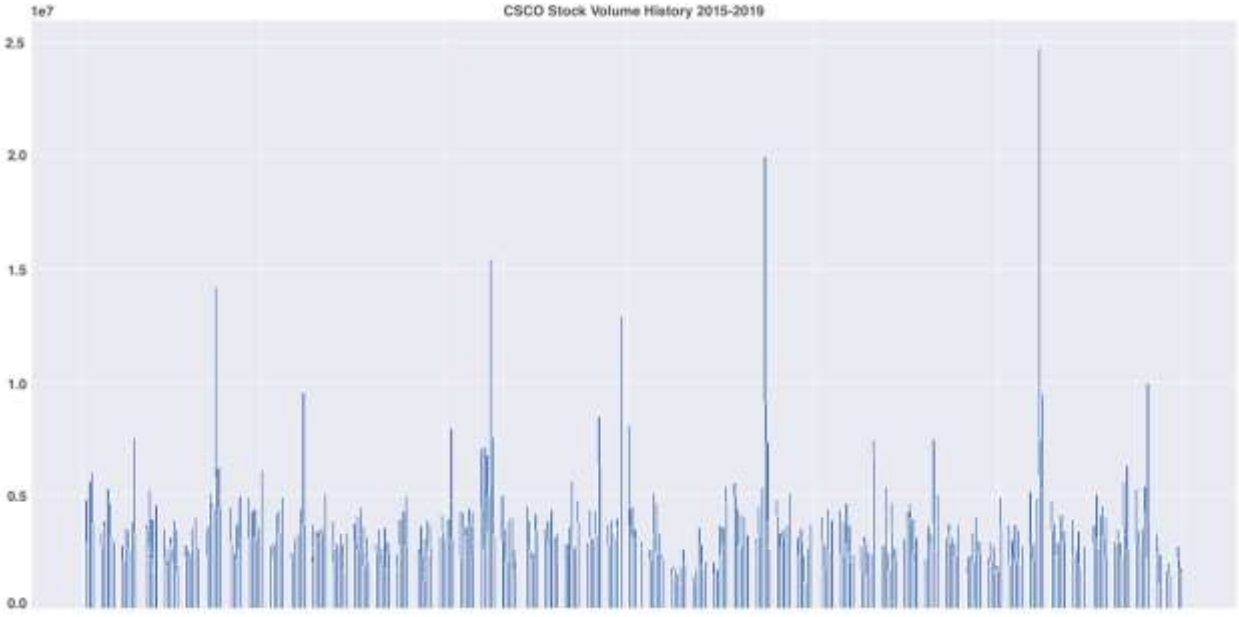

Figure 7. CSCO Stock Volume History 2015-2019

The plots for $IBM stock are shown below with a stock price history (Figure 8) and a volume stock history (Figure 9) summary.

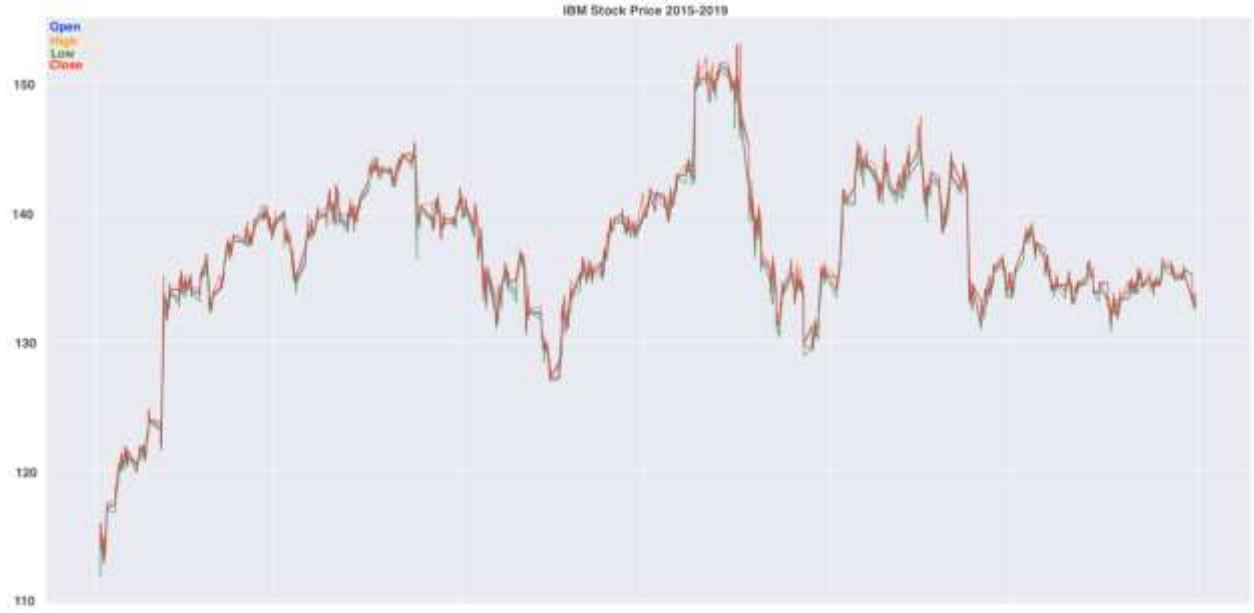

Figure 8. IBM Stock Price History 2015-2019

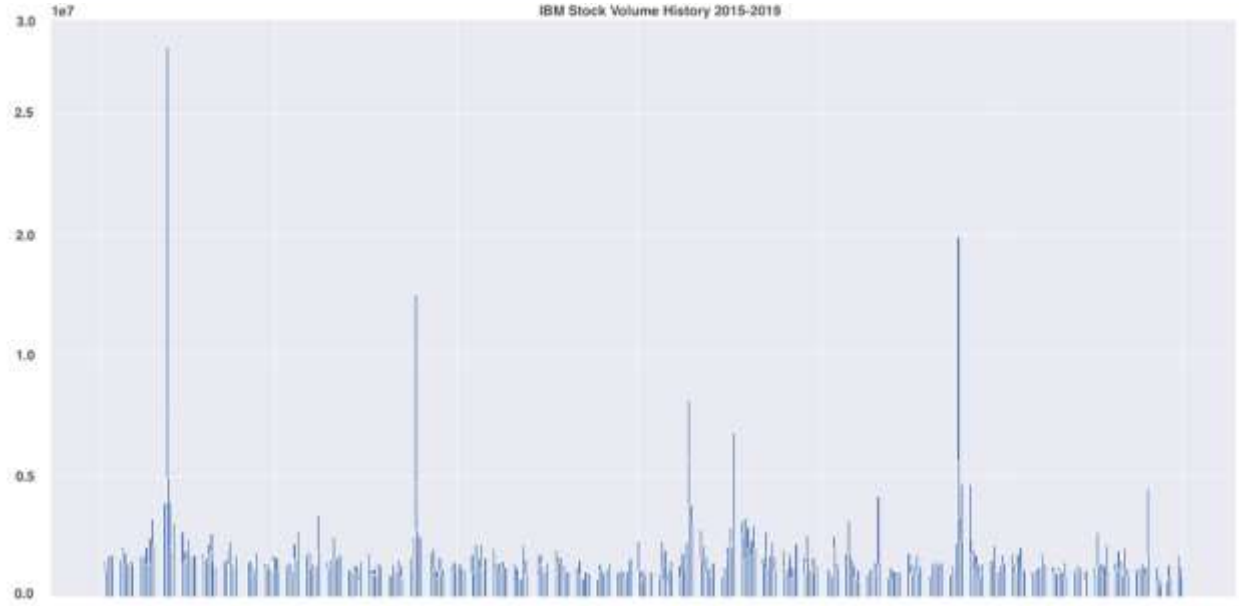

Figure 9. IBM Stock Volume History 2015-2019

The plots for $MSFT stock are shown below with a stock price history (Figure 10) and a volume stock history (Figure 11) summary.

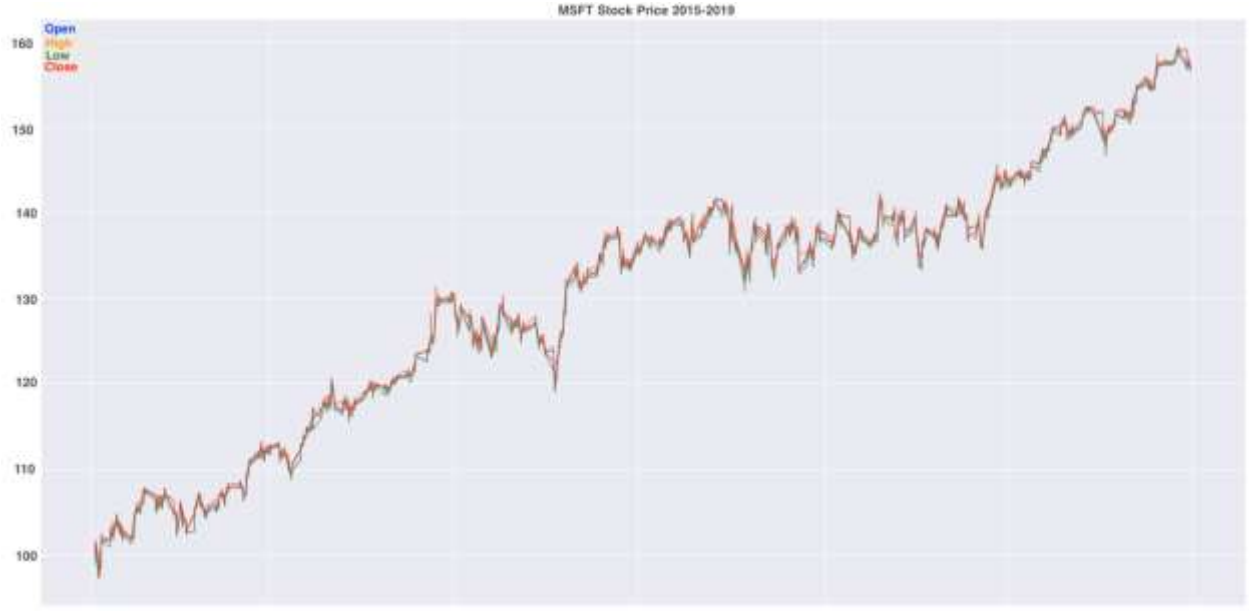

Figure 10. MSFT Stock Price History 2015-2019

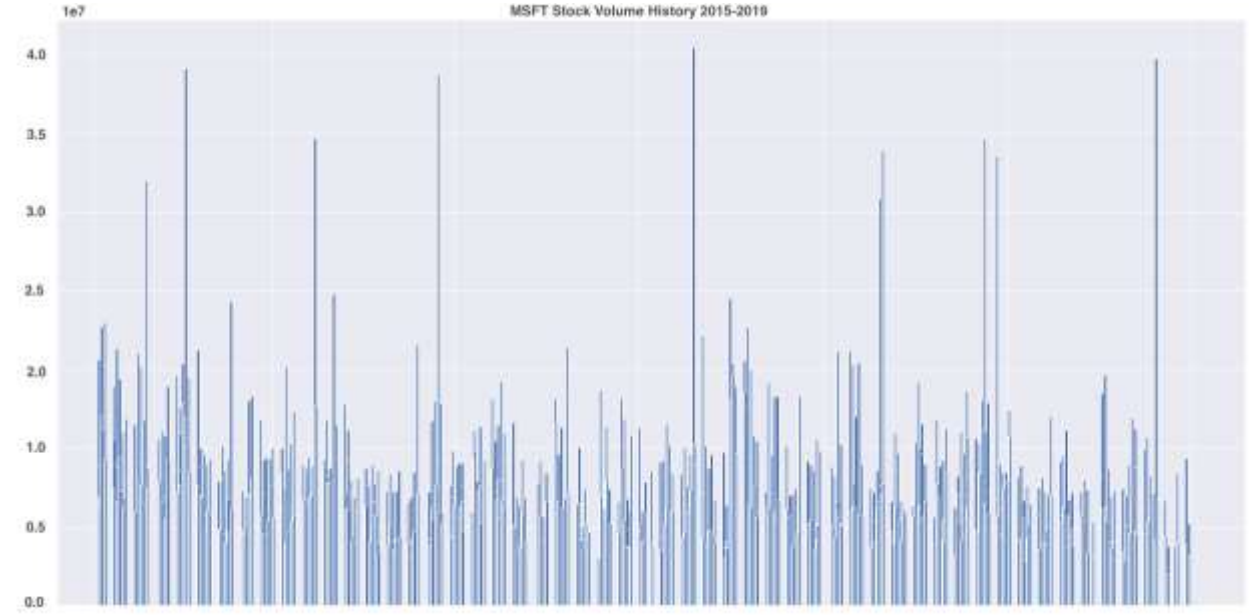

Figure 11. MSFT Stock Volume History 2015-2019

The trends in the graphs above vary significantly, which shows that there are many events that influence stock trends, making stock trend prediction a difficult task.

3.4.2 FS Dataset Preparation

FS data usually consists of features such as: Open, High, Low, Close, Volume, and Date. These features are not enough to do a proper financial analysis. As a result, further feature extraction is required to perform an accurate analysis (Moukalled & Mohamad, 2019). These additional features will help determine why and when prices for certain stocks are high or low. For example, the Bollinger bands allow us to determine if the stock price is high or low and identify if it is over purchased or oversold (Moukalled & Mohamad, 2019). In combination with moving average and the time of day, Bollinger bands can help identify what trends are occurring with a stock (Thampi, Trajkovic, Das, Wozniak, & Berretti, 2019). Stock market indicators such as time (real-time integer-encoded results), weekday (integer-encoded result depending on day of the week), true range (current high minus the current low), Bollinger Indicator (technical indicator used to measure a market's volatility and identify if the stock is “overbought” or “oversold”), stock closing price minus upper Bollinger indicator band, and stock closing price minus lower Bollinger indicator band, simple moving average indicator (integer-encoded result from the simple moving average stock mathematical indicator), are all valuable. They are generally more effective when used together to provide a complete view of the stock price behavior. These features are widely used in chart indicators which gives investors clear visuals (Thampi, Trajkovic, Das, Wozniak, & Berretti, 2019).

The features need to be normalized using the transformation function defined in Equation (2). Variables that are measured at different scales do not contribute equally to the model fitting and model learned function, and might end up creating a bias. To solve this potential problem, feature normalization, such as MinMaxScaler, defined in Equation (2), is used prior to the model fitting.

$$MinMaxScaler(y_i) = \frac{y_i - MIN(Y)}{MAX(Y) - MIN(Y)} \qquad (2)$$

where y_i is the original value, Y is the desired random variable, $MAX(Y)$ is the maximum value, and $MIN(Y)$ is the minimum value.

3.4.3 FS Dataset Cross Validation Technique

The FS dataset is considered a time-series data. The use of a typical train-test split and K-fold cross-validation would not be effective with time-series data, which is why we use time-series cross-validation for the FS dataset. Time series cross validation is also called cross validation on a rolling basis, which provides a train and test indices and allows the time series data samples that are observed at fixed time intervals to be split. Shuffling in time series cross validation would not work since in each split the test indices must be higher than the previous set. This method starts with a small subset of data for training, later data points are forecasted, and the accuracy is then checked for the forecasted data points. The data points that are forecasted are then included as part of the next training dataset and subsequent data points are forecasted. Time series cross validation is a variation of K-Fold cross validation. In the Kth split, the first K-Fold is returned as a train and the Kth + 1-fold is the test set. In time series cross validation, successive training sets are supersets of the sets that came before them. For the FS dataset, we will be using 10 folds.

Chapter 4: Model Construction and Evaluation

This chapter presents the details of the proposed model. As mentioned in Chapter 3, two data types are required as input for the model: financial stock and social media data. Therefore, in addition to assessing financial stock data, the proposed model achieves prediction through the analysis of social media messages to determine the sentiment of retail investors. To obtain the sentiment score from the social media data, we will use an ensembled-based model that uses Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN) member models. The ensemble contains two MLP models which we call the MLP Feature Driven and MLP Simple Word Embedding models. Additionally, the proposed solution employs a financial stock data (FSD) prediction model which consists of two LSTM layers. The proposed solution was tested on the data of four stock tickers, namely AAPL, CSCO, IBM, and MSFT.

4.1 Ensemble Sentiment Estimation (ESE) Model Selection

We employ an ensemble approach to predict the sentiment. The ensemble comprises four members models: Multi-Layer Perceptron Feature Driven (MLP FD), Multi-Layer Perceptron Simple Word Embedding (MLP SWE), Convolutional Neural Network (CNN), and Long Short-Term Memory Neural Network (LSTM). Contrary to supervised learning, Neural Network models specifically Deep Learning models, perform better in sentiment analysis tasks since they do not require manually tuned features based on expert knowledge and available linguistic features (Rojas-Barahona, 2016). Deep Learning techniques learn through the use of multiple layers of representation to generate state of the art predictive results through the use of automatic feature extraction which can yield better performance than traditional feature based techniques in

sentiment analysis (Rojas-Barahona, 2016). In traditional machine learning models, we identify and extract features manually or through the use of feature selection methods (Jain, 2020). However, with the use of deep learning, features are learned and extracted automatically which can lead to higher accuracy and performance (Jain, 2020). To combine the results of the four models, we use a Multi-Layer Perceptron Stacking as the blending model. Stacking is an ensemble machine learning algorithm that uses a meta-learning algorithm to learn how to optimally combine the predictions of the ensemble members (Nti, Adekoya, & Weyori, 2020. The benefit of using stacking is that it harnesses the capabilities of models that have performed moderately well in classification and regression tasks (otherwise known as weak learners) to produce better predictions than any member model of the ensemble (Nti, Adekoya, & Weyori, 2020). Model averaging which is another ensemble technique that requires each member of the ensemble to contribute equally, regardless of their individual performance (Nti, Adekoya, & Weyori, 2020). Instead, we have chosen the stacking approach for our ensemble solution to benefit from the flexibility it affords to combining the predictions of the member models. Figure 12 shows the proposed ensemble-based scheme.

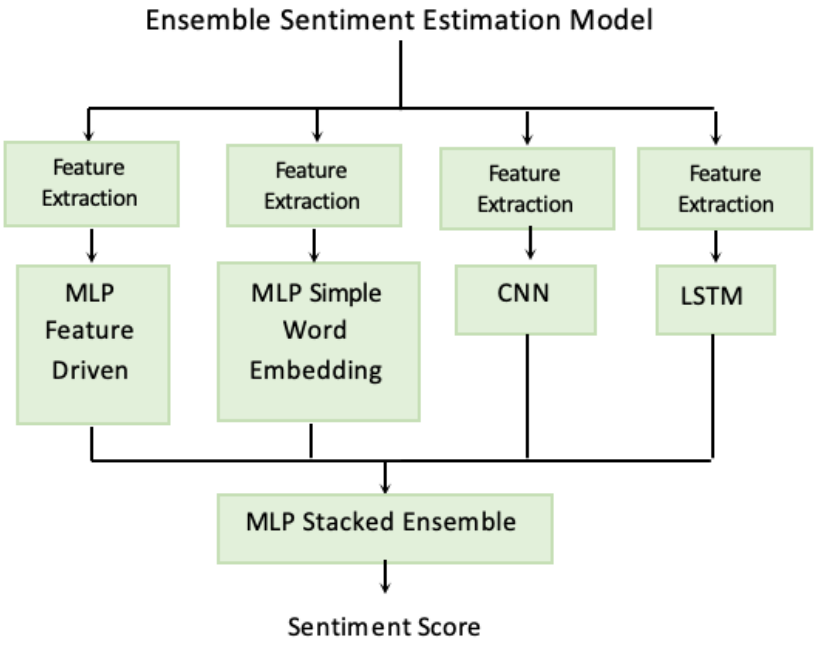

Figure 12. Ensemble Sentiment Estimation Model

In the following sections, we will specify the hyperparameters that we selected and present the performance results for each ensemble member of the ESE model. Each ensemble member model serves a unique purpose. By combining the outputs of all four-member models using the MLP Stacked Ensemble model, we obtain the final sentiment score. We found the best performing hyperparameters for all the models through a trial-and-error process.

4.1.1 Multi-Layer Perceptron Feature Driven (MLP FD)

Table 3. MLP FD Hyperparameters

Architecture Details	**MLP Feature Driven**
Hidden Layers	3
Nodes	50, 30, 15
Activation Function	Rectified Linear Activation Function (ReLU) for the

	hidden layer and Hyperbolic Tangent Function (TanH) for the output layer
Dropout Layer	0.5
Input Dropout	0.25
Hidden Dropout	0.2
L2 Regularization	0.01
Epochs	250
Batch Size	32
Learning Rate	0.0005
Optimizer	Adam
Loss Function	Mean Square Error (MSE)

The MLP FD model estimates the sentiment score using manually chosen features examined in chapter 3 (section 3.1.1). The MLP FD takes three of the manually chosen feature sets as the input to the model. These feature sets are the stock, linguistic, and the sentiment lexicon features. This corresponds to a total of 52 features (for more information, see Appendix B). The MLP FD has three hidden layers with an activation function for the hidden and output layers. Activation functions are an important part of the design of a neural network. The activation function in the hidden layer controls how well the model learns in the training dataset (Nwankpa, Ijomah, Gachagan, & Marshall, 2018). The type of predictions the model makes is also controlled by the choice of the activation function in the output layer (Nwankpa, Ijomah, Gachagan, & Marshall, 2018). An activation function is a mapping of the summed weighted input to the output of the

neuron (Nwankpa, Ijomah, Gachagan, & Marshall, 2018). The activation function also governs the threshold at which the neuron is activated and strength of the output signal (Nwankpa, Ijomah, Gachagan, & Marshall, 2018). We use the Rectified Linear Activation Function (ReLU). ReLU is a non-linear activation function used in multi-layer neural networks and deep neural networks (Nwankpa, Ijomah, Gachagan, & Marshall, 2018). ReLU specifically overcomes the vanishing gradient problem which allows models to learn faster and perform better (Nwankpa, Ijomah, Gachagan, & Marshall, 2018). The main advantage of the ReLU function is that it does not activate all the neurons at the same time which is more computationally efficient when compared to other activation functions (Nwankpa, Ijomah, Gachagan, & Marshall, 2018).

Hidden layers are located between the input and the output of the algorithm in which the weights are then applied to the input and then directed to the activation function as the output. The number of nodes in the hidden layers are 50, 30, and 15 in the first, second, and third hidden layer respectively. The MLP FD employs dropout. Dropout is a regularization technique that reduce the probability of overfitting during training. Dropout forces some of the nodes in various layers of the network (except the output) to be dropped with a certain predefined probability. This forces a greater number of model nodes to contribute to the predictions, which in turn often reduces overfitting. The hyperparameter values are summarized in Table 3. We assess the MLP FD model using the Cosine Similarity assessment metric. Cosine Similarity is a measure used to compare documents based on their similarities. In figure 13 , we show the average of the Stratified k-fold cross validation of all the results for the MLP FD model using 10-folds on the SET5 Training Data 1 and the SET5 Validation Data 1 as explained in chapter 3 (section 3.3).

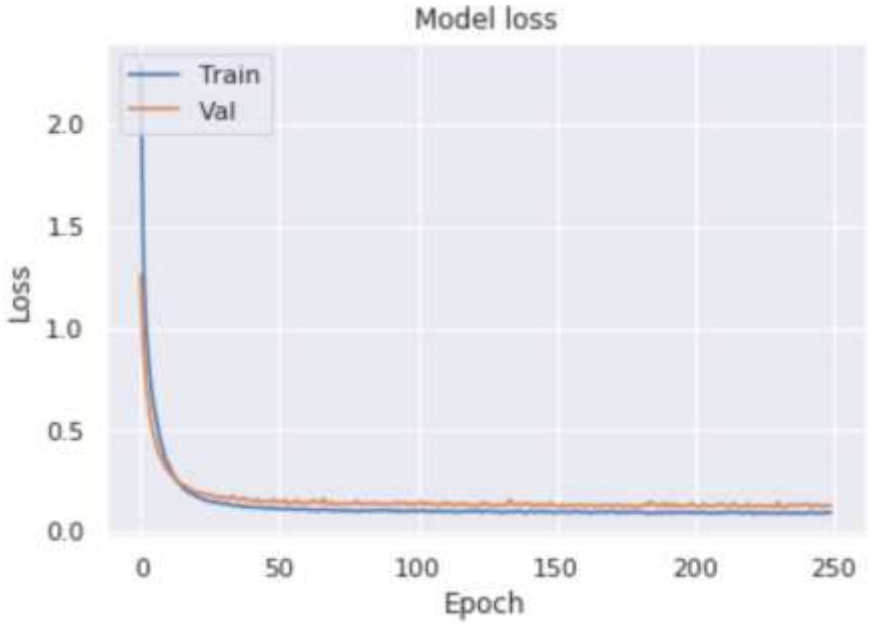

Figure 13. MLP FD SET5 Training Data 1 & SET5 Validation Data 1 Model Loss Values

SET5 Training Data 1 Cosine Similarity: 0.9095829

SET5 Validation Data 1 Cosine Similarity: 0.5147709

After running the Stratified 10-Fold cross validation, our Stratified 10-Fold Cosine Similarity average score for the MLP FD model is 0.514 (51.4 percent) and the standard deviation score is 0.076. The model performs well with an average Cosine Similarity above 0.5. Regardless of its individual performance, the MLP FD will serve a greater purpose when used in the MLP Stacked Ensemble model.

4.1.2 Multi-Layer Perceptron Simple Word Embedding (MLP SWE)

Table 4. MLP SWE Hyperparameters.

Architecture Details	MLP Simple Word Embedding
Hidden Layers	3

Nodes	30, 30, 30
Activation Function	Rectified Linear (ReLU) Activation Function for the hidden layer and Hyperbolic Tangent Function (TanH) for the output layer
Dropout Layer	0.5
Input Dropout	0.25
Hidden Dropout	0.25
L2 Regularization	0.004
Epochs	200
Batch Size	32
Learning Rate	0.0001
Optimizer	Adam
Loss Function	Mean Square Error (MSE)

The MLP SWE model estimates the sentiment score using the vector representation of texts obtained from Google Word2Vec[7]. The MLP SWE takes the cleaned text explained in chapter 3 (see section 3.1.2) as the input to the model which is then transformed into its word-embedding vector representation using the Google Word2Vec pre-trained model. The objective of Word2Vec

[7] https://code.google.com/archive/p/word2vec/

is to generate vector representations of words that carry semantic meanings for further NLP tasks. Each word vector is typically several hundred dimensions and each unique word in the corpus is assigned a vector in the space. With the help of the pretrained Google News Word2Vec, each word in the cleaned text is transformed into a 300-dimensional vector. The MLP SWE has three hidden layers with an activation function for the hidden and output layers. Activation functions are an important part of the design of a neural network. The MLP SWE consists of a ReLU activation function for the hidden layer and TanH for the output layer similar to the MLP FD (see chapter 4 section 4.1.1 for a more detailed explanation of the activation functions). Similar to the MLP FD model, the hidden layers of the MLP SWE model are located between the input and the output where the weights are applied to the input of each node and then directed to the activation function as the output of the node. The number of nodes in the hidden layers is 30 for all three hidden layers. The MLP SWE model also applies a dropout for regularization. The hyperparameter values are summarized in Table 4. We assess the MLP SWE model using the Cosine Similarity assessment metric. In figure 14, we show the average of the stratified k-fold cross validation of all the results for the MLP SWE model using 10-folds on the SET5 Training Data 1 and the SET5 Validation Data 1 as explained in chapter 3 (section 3.3).

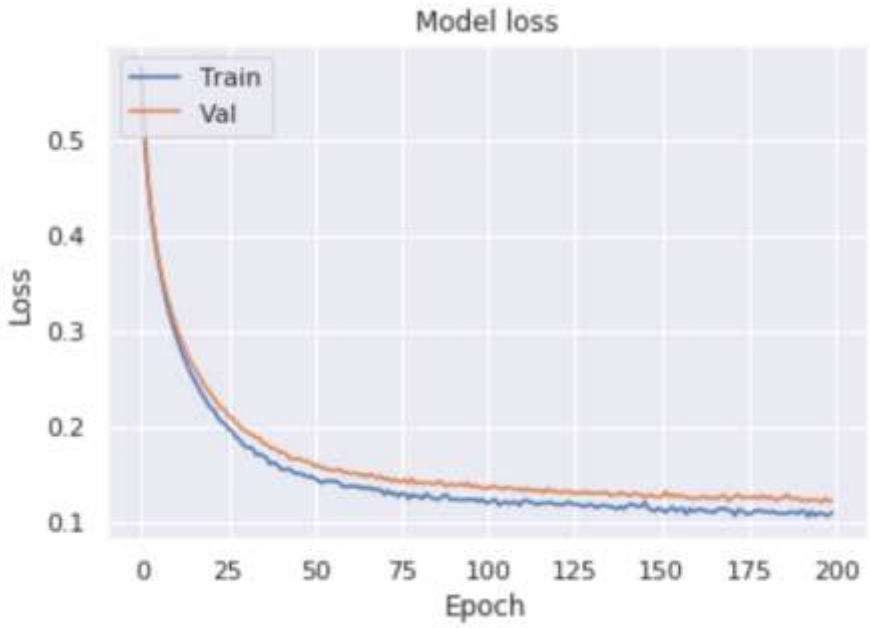

Figure 14. MLP SWE SET5 Training Data 1 & SET5 Validation Data 1 Model Loss Values

SET5 Training Data 1 Cosine Similarity: 0.7208512

SET5 Validation Data 1 Cosine Similarity: 0.5686155

After running the Stratified 10-Fold cross validation, our Stratified 10-Fold Cosine Similarity Average score for the MLP SWE model is 0.568 (56.8 percent) and the standard deviation score is 0.086. The model performs well with an average cosine similarity above 0.5. Regardless of its individual performance, the MLP SWE will serve a greater purpose when used in the MLP Stacked Ensemble model.

4.1.3 Convolutional Neural Network (CNN)

Table 5. CNN Hyperparameters.

Architecture Details	CNN
Hidden Layers	2
Nodes	15, 15

Activation Function	Hyperbolic Tangent Function (TanH)
Dropout Layer	0.45 (Hidden Layer)
L2 Regularization	0.006 (Hidden Layer)
Epochs	75
Batch Size	32
Learning Rate	0.0005
Optimizer	Adam
Loss Function	Mean Square Error (MSE)

The CNN aims to retrieve the local behavior of the text. The architecture of the CNN model is composed of two parts, where the first is responsible for feature extraction, while the second is responsible for the fully connected network. CNN takes the cleaned text explained in chapter 3 (see section 3.1.2) as the input to the model, with a maximum of 50 words in each text. Padding is performed on the text with length less than 50 words. Figure 15 depicts the feature extraction module with a 300-dimensional sentence embedding vector as the input. We use Google's Word2Vec[8] pre-trained model to convert each text word into its 300-dimensional vector representation. This vector is then passed to the CNN model. The first layer in the CNN is the Gaussian noise layer. Introducing noise to the input is a regularization strategy that reduces

[8] https://code.google.com/archive/p/word2vec/

overfitting. Therefore, the noise layer is only active during training. The vector is then passed into the Gaussian Noise layer and four layers of convolutions are applied. Twenty-five convolutions are applied for each type resulting in a total of 100 convolutions. Each convolution layer has a different size of output vector, such that, layer 1 convolution produces a 300-dimensional vector, layer 2 produces a 299-dimensional vector, layer 3 produces a 298-dimensional vector, and layer 4 produces a 297-dimensional vector. Once, the convolution layers are complete, the Tanh activation layer is applied on the output. The TanH function converts any real value input into an output ranging between -1 and 1. The TanH function typically improves training performance and is widely used in NLP (Nwankpa, Ijomah, Gachagan, & Marshall, 2018). Moreover, a 1-max pooling is also applied on the output, resulting in 100 univariate vectors. The vectors are then concatenated to produce a single 100-dimensional feature vector. This vector is then inputted into a fully connected network. Max pooling is a down-sampling strategy that selects the maximum element from the region of the feature map covered by the filter. The output of the max-pooling layer is a feature map containing the most important features of the previous map. The CNN feature extraction is summarized in Figure 15. The details of the hyperparameters are shown in Table 5.

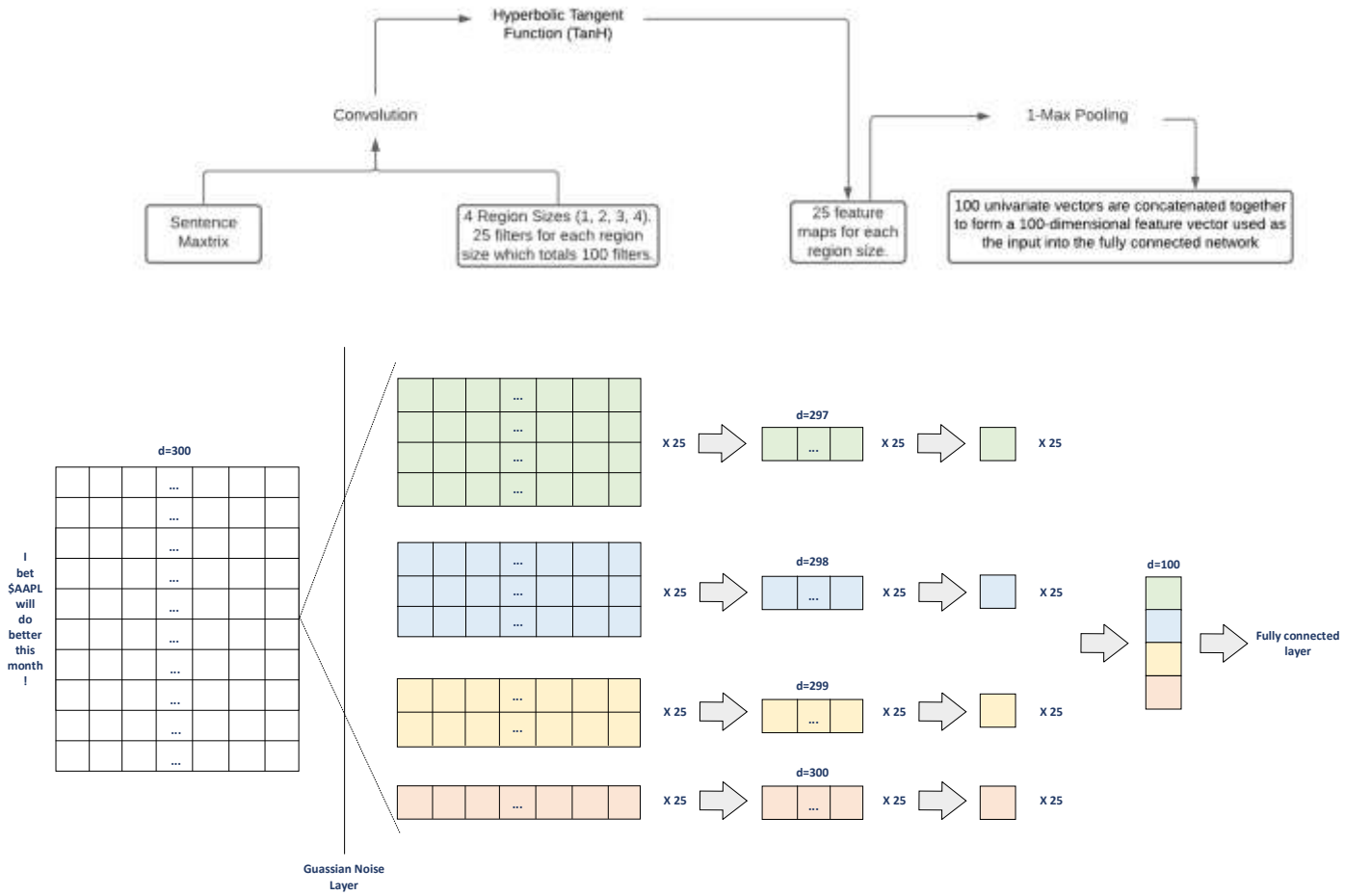

Figure 15. CNN Architecture and Feature Extraction Network for the Social Media Data

This CNN model is assessed using the Cosine Similarity measure. In figure 16, we show the average of the stratified k-fold cross validation of all the results for the CNN model using 10-folds on the SET5 Training Data 1 and the SET5 Validation Data 1 as explained in chapter 3 (section 3.3).

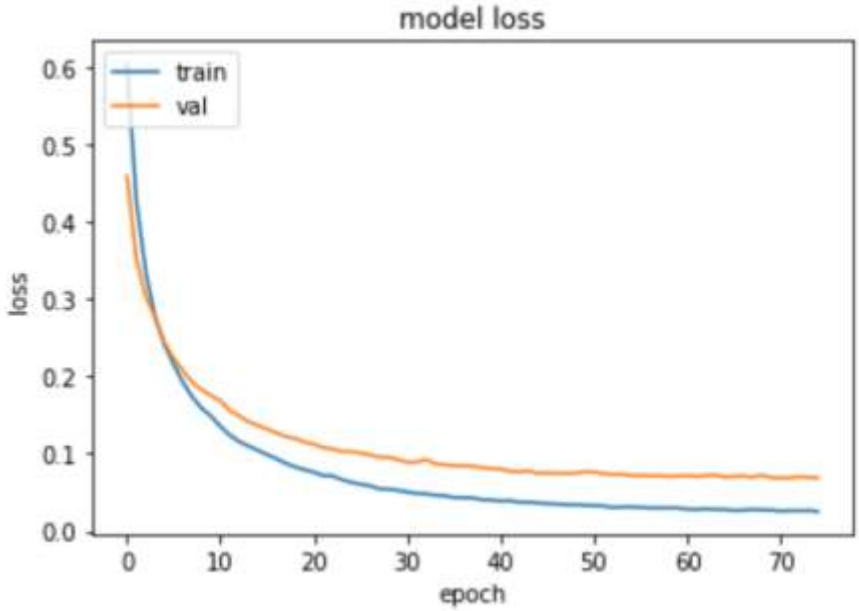

Figure 16. CNN SET5 Training Data 1 & SET5 Validation Data 1 Model Loss Values

SET5 Training Data 1 Cosine Similarity: 0.9151488

SET5 Validation Data 1 Cosine Similarity: 0.5723806

The Stratified 10-Fold Cosine Similarity Average score for the CNN model is 0.572 (57.2 percent) and the standard deviation score is 0.082. The model performs well with an average cosine similarity above 0.5. Similar to the MLP FD and MLP SWE, regardless of the CNN's individual performance, it will serve a greater purpose when used in the MLP Stacked Ensemble model.

4.1.4 Long Short-Term Memory Model (LSTM)

Table 6. LSTM Hyperparameters

Architecture Details	LSTM
Hidden Layers	2
Nodes	50, 10

Activation Function	Rectified Linear Activation Function
Dropout Layer	0.3 (Hidden Layer)
L2 Regularization	0.05 (Hidden Layer)
Epochs	350
Batch Size	32
Learning Rate	0.0001
Optimizer	Adam
Loss Function	Mean Square Error (MSE)

The LSTM captures the global behavior of the text. LSTM models are considered an important type of RNN since they have the capability of learning long-term dependencies. Similar to the CNN model, The LSTM model takes the cleaned text explained in chapter 3 (see section 3.1.2) as the input to the model, with a maximum of 50 words in each text. Padding is performed on the text with length less than 50 words. We use Google's Word2Vec[9] pre-trained model to convert each text word into its 300-dimensional vector representation. This vector is then passed to the LSTM model. The LSTM models uses two layers on top of each other which is then followed by two dense layers and an output layer. Each LSTM layer has a fixed number of neurons of 100, while

[9] https://code.google.com/archive/p/word2vec/

for the dense layers, 50 neurons and 10 neurons are used in the hidden layers respectively. The first layer in the LSTM model is the Gaussian noise layer. Introducing noise to the input is a regularization strategy that reduces overfitting. Since the Gaussian Noise Layer is considered a regularization layer, then it is only active during training and not during the evaluation of the model. The Gaussian Noise Layer is conducted after the input embedding layer and before the two LSTM layers, followed by the Dropout Layer for each of the two LSTM layers as seen in the LSTM architecture shown in figure 17. The embedding layer enables the conversion of each word into a fixed length vector with a fixed size which allows the resultant vector to have real values instead of 0 and 1's. This allows us to represent words in a much better way with its dimensions reduced. The dropout layer is a technique where input and recurrent connections to the LSTM units are probabilistically not activated and the weights are not updated while in the training phase. Once completed, it can then be used as an input.

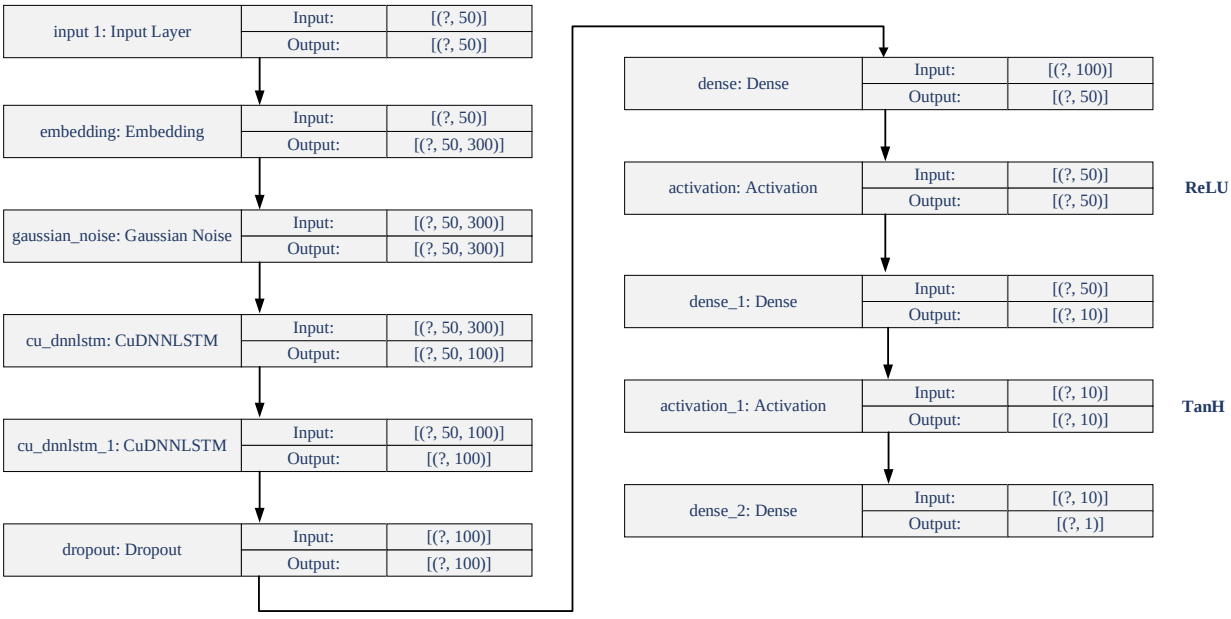

Figure 17. LSTM Architecture and Flow

The details of the hyperparameters are shown in Table 6. This model is assessed using the Cosine Similarity measure. In figure 18, we show the average stratified k-fold cross validation of all the results for the LSTM model using 10-folds on the SET5 Training Data 1 and the SET5 Validation Data 1 as explained in chapter 3 (section 3.3).

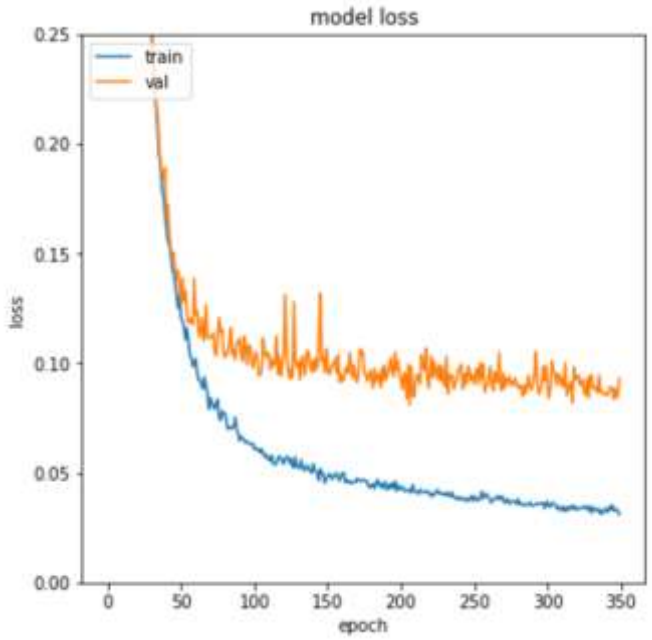

Figure 18. LSTM SET5 Training Data 1 & SET5 Validation Data 1 Model Loss Values

SET5 Training Data 1 Cosine Similarity: 0.9033100

SET5 Validation Data 1 Cosine Similarity: 0.5429877

The Stratified 10-Fold Cosine Similarity Average score for the LSTM model is 0.542 (54.2 percent) and the standard deviation score is 0.082. The model performs well with an average cosine similarity above 0.5. Similarly to the MLP FD, MLP SWE, and the CNN Model, the LSTM model performs well with an average cosine similarity above 0.5. Regardless of its individual performance, the CNN model will serve a greater purpose when used in the MLP Stacked Ensemble model.

4.1.5 MLP Stacking Ensemble Model

Table 7. MLP Stacking Ensemble.

Architecture Details	Stacked Ensemble Model
Hidden Layers	1
Nodes	4
Activation Function	ReLU for hidden layer, TanH for output layer
Dropout Layer	0.05
L2 Regularization	0.02
Epochs	300
Batch Size	32
Learning Rate	0.00075
Optimizer	Adam
Loss Function	Mean Square Error (MSE)

We apply a Multi-Layer Perceptron Stacking Ensemble model to blend the sentiment estimates of the four ensemble member models, namely MLP FD, MLP SWE, CNN, and LSTM. Stacking is an ensemble machine learning algorithm that uses a meta-learning algorithm to learn how to optimally combine the predictions of the ensemble members (Nti, Adekoya, & Weyori, 2020. The benefit of using stacking is that it harnesses the capabilities of models that have performed moderately well in classification and regression tasks which in return will make a prediction that

has better performance than any model used in the ensemble (see section 4.1 for more details regarding the ensemble model) (Nti, Adekoya, & Weyori, 2020). The details of the hyperparameters for the MLP Stacked Ensemble model are shown in Table 7. In figure 19, we show the cosine similarity average score for the MLP Stacked Ensemble model on the SET5 Training Data 2 and the SET5 Validation Data 2 as explained in chapter 3 (section 3.3).

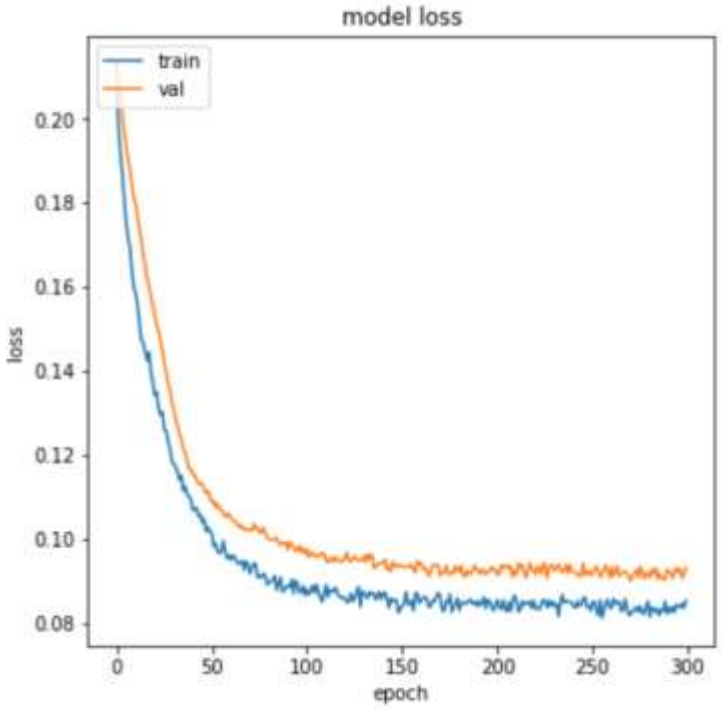

Figure 19. MLP Stacked Ensemble Plot training & validation loss values

SET5 Training Data 2 Cosine Similarity: 0.9162778
SET5 Validation Data 2 Cosine Similarity: 0.6749153

Each of the four models, MLP FD, MLP SWE, CNN, and LSTM, achieved a cosine similarity average score around 0.5. When combined using the MLP Stacked Ensemble model, we achieved a cosine similarity of 0.674 (67.4%).

4.2 Financial Stock Data (FSD) Model Selection

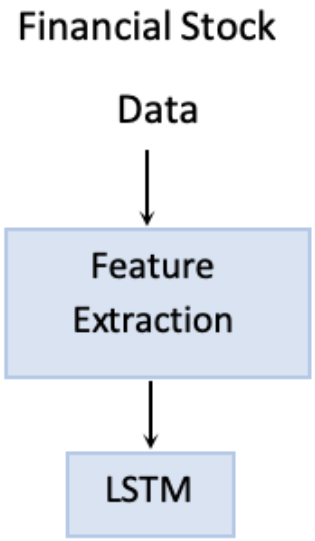

Figure 20. Financial Stock Data Model

The FSD model seen in Figure 20, consists of two Long Short-Term Memory layers (LSTM). LSTM models are ideal for processing time-series inputs. They learn the inter-dependencies between the observations of the input series. Given the sequential nature of financial stock data, LSTM models can be powerful predictors of future behavior based on current and past trends.

Time Series refers to a series of observations collected at constant time intervals whether it is daily, monthly, quarterly, or yearly. A time series problem where the model predicts one or more future numerical values is a regression type predictive modeling problem. In the following sections, we present the hyperparameters we use in the LSTM model to predict the financial stock price.

We use the Mean Absolute Percentage Error (MAPE) (Equation 3) to assess the performance of the FSD model. This metric has been widely employed to evaluate stock prediction models

(Kim & Kim, 2016). It is especially popular given how simple it is to interpret its significance. For instance, MAPE value of 10% signifies that the difference between the actual and the forecasted value is 10%. The models' performance is computed using MAPE by calculating the absolute percentage error.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right| * (100) \quad (3)$$

Where *n* is the number of samples, x_i is the stock price, and $\hat{x}_i$ is the predicted stock price.

We will also be using a confusion matrix to summarize the prediction results. A confusion matrix is an N x N matrix used to evaluate the performance of a model, where N is the number of target classes (Castoe, 2020). A confusion matrix calculates the target value with the values predicted by the model which will allow us to see how well our model is performing and where the errors are being made (Castoe, 2020). Confusion Matrix is the visual representation of the actual vs predicted results which will allow us to measure the performance of our machine learning model (Deepanshi, 2021). Using the confusion matrix, we will be able to calculate the Accuracy (Equation 3). Accuracy (Equation 4) is a ratio of correctly predicted observations to the total observations.

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+FP+TN+FN}} * (100) \quad (4)$$

Where,

TP + TN = Number of correct Predictions,

TP + FP + TN + FN = Total number of predictions,

True Positive (TP) – The predicted value matches the actual value,

True Negative (TN) – The predicted value matches the actual value,

False Positive (FP) – The predicted value was falsely predicted,

False Negative (FN) – The predicted value was falsely predicted.

When conducting our experiments, we use the predicted stock price to compare it to the actual stock price to calculate our MAPE value. Our predicted stock price is then used to calculate the price fluctuation on the forecasted stock price. If the predicted stock price increased, then the output is 1, and if the predicted stock price decreased, then the output is 0. These results will allow us to calculate the confusion matrix and Accuracy metric.

We first use the flatten() function from the NumPy library to collapse our array into a one dimensional array. We then convert the stock price to a range of real data for both the predicted stock price and the actual stock price in the range of 0 and 1. We then check difference of the forecasted stock price of the current day minus the previous day. If the forecasted stock price increased then the output is 1, and if the forecasted stock price decreased, then the output is 0 (lambda x:1 if x > 0 else 0). This is done on both the predicted stock price and the actual stock price. The confusion matrix is then applied which allows us to calculate the Accuracy measurement.

4.2.1 FS Dataset LSTM Model

Table 8. FS Dataset LSTM Hyperparameters.

Architecture Details	LSTM Layers	
Nodes	256	128
Activation Function	Linear activation function	
L2 Regularization	0.005	0.005
Epochs	350	350
Batch Size	25	25
Learning Rate	0.0005	0.0005
Optimizer	Adam	Adam
Loss Function	Mean Square Error (MSE)	

The input to the LSTM model will consist of the FS dataset which was described in chapter 3 (section 3.4.2). The FS dataset pertains to four companies: AAPL, CSCO, IBM, and MSFT. The scraped data covers the period from 2015 to 2019. We assess this model using a 10-fold MAPE average score and 10-fold validation accuracy average score. The FS dataset is considered a time-series data. The use of a typical train-test split and K-fold cross-validation would not be effective with the time-series data, which is why we use time-series cross-validation for the FS dataset. Time series cross validation is also called cross validation on a rolling basis. This method starts with a small subset of data for training, later data points are forecasted and the accuracy is then checked for the forecasted data points. The data points that are forecasted are then included as part of the next training dataset and subsequent data points are forecasted (see chapter 3 section 3.4.3 for more

details). The details of the hyperparameters are shown in Table 8. In figures 21 to 28 we show the model loss and the 'prediction vs real stock price' for each of the $AAPL, $CSCO, $IBM, and $MSFT stocks using the FS dataset as explained in chapter 3 (see section 3.4 for more details).

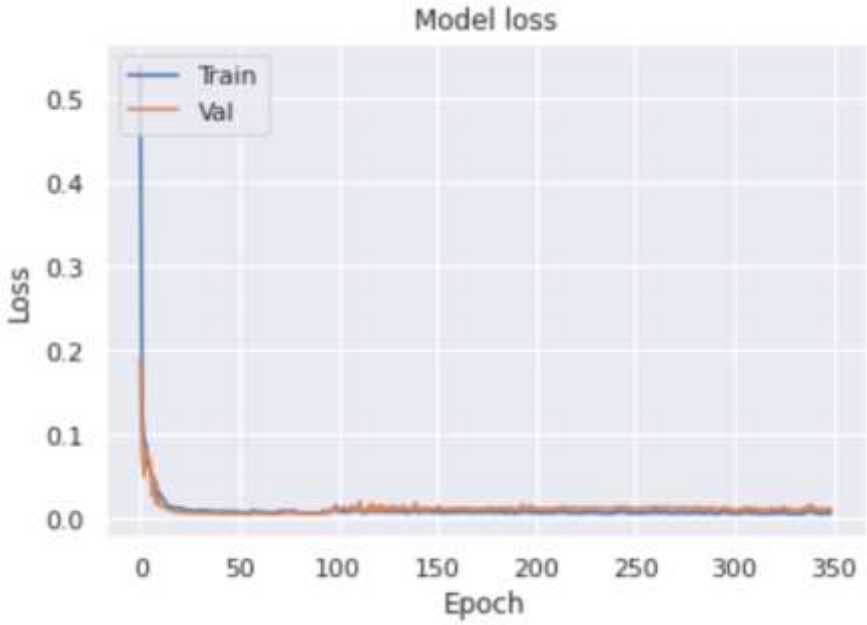

Figure 21. $AAPL FSD Model Loss

Figure 22. $AAPL FSD Model Prediction Vs Real Stock Price

$AAPL MAPE Accuracy: 1.8123452

$AAPL Prediction vs Real Stock Price Performance: 0.69156087

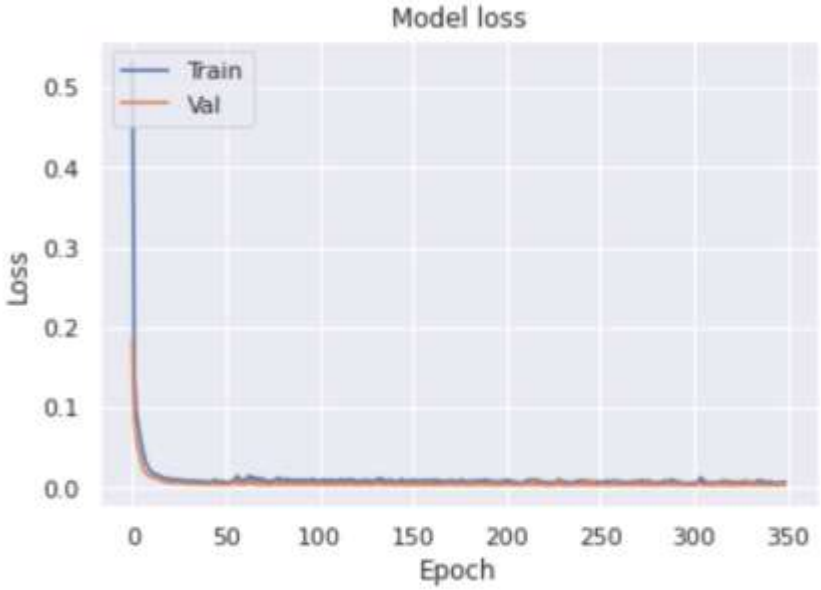

Figure 23. $CSCO FSD Model Loss

Figure 24. $CSCO FSD Model Prediction Vs Real Stock Price

$CSCO MAPE Accuracy: 1.5036873

$CSCO Prediction vs Real Stock Price Performance: 0.6839154

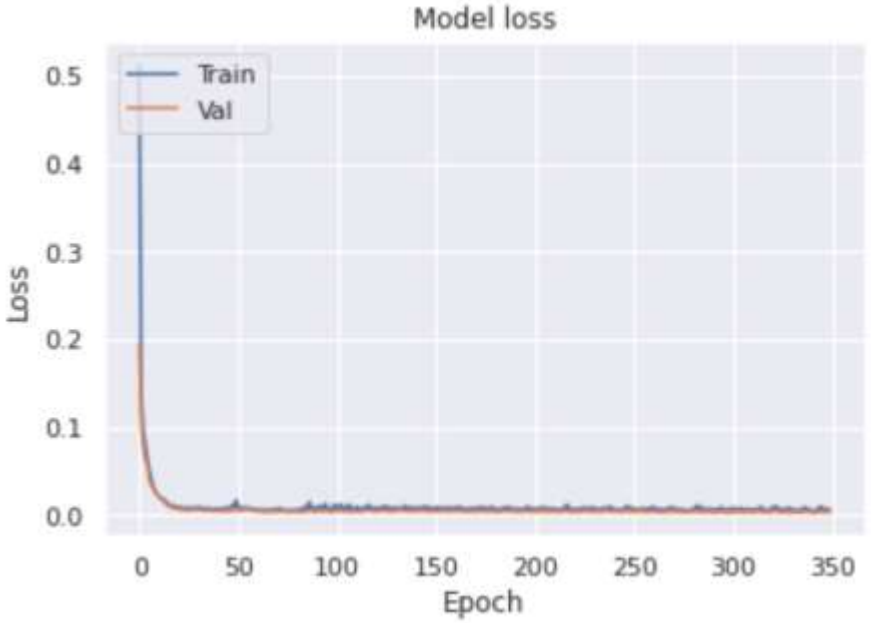

Figure 25. $IBM FSD Model Loss

Figure 26. $IBM FSD Model Prediction Vs Real Stock Price

$IBM MAPE Accuracy: 1.4181419

$IBM Prediction vs Real Stock Price Performance: 0.6804351

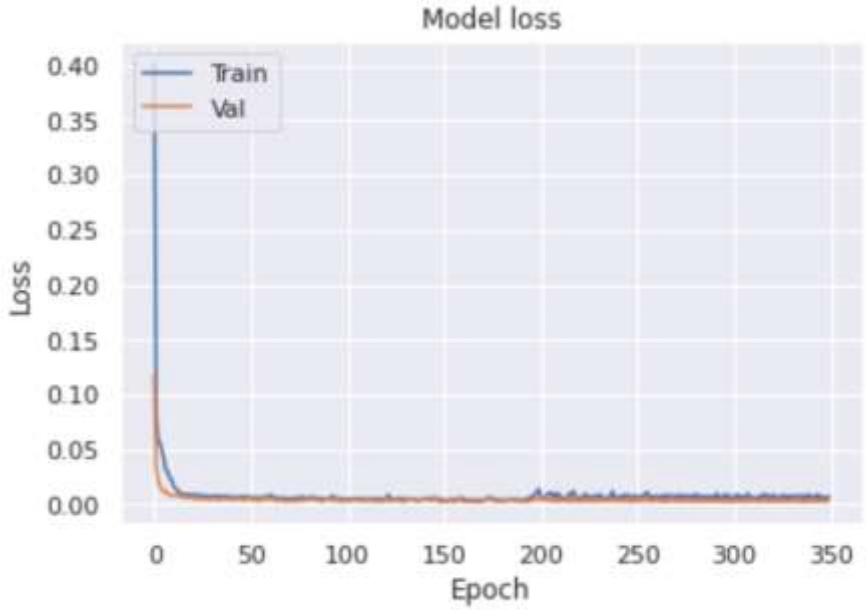

Figure 27. $MSFT FSD Model Loss

Figure 28. $MSFT FSD Model Prediction Vs Real Stock Price

$MSFT MAPE Accuracy: 1.4399486

$MSFT Prediction vs Real Stock Price Performance: 0.6899093

After running the 10-Fold MAPE Accuracy and 10-Fold prediction vs real stock price performance, we achieved an average MAPE of 1.543 and an average 10-Fold prediction vs real stock price performance of 0.685 (68.5%). The MAPE standard deviation score is 0.56.

4.3 Sentiment and Financial Data (SFD) model: Combination of The ESE Model and the FSD Model

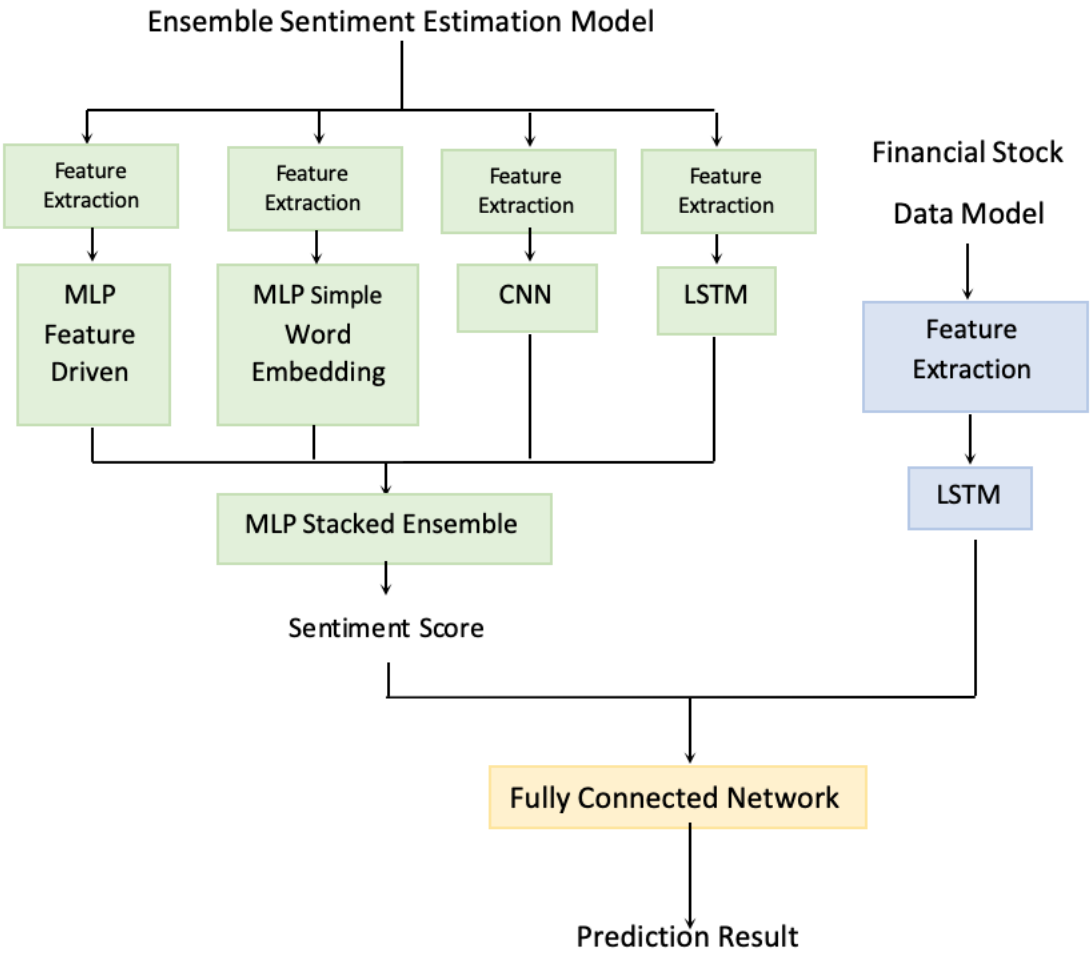

Figure 29. Sentiment and Financial Data (SFD) Model

The objective of this thesis is to combine the financial stock and sentiment information to predict stock market prices. Hence, we propose fusing the outputs of the ESE model and the FSD model to generate a prediction that incorporates the strength of both models. Figure 29 depicts the proposed Sentiment and Financial Data (SFD) model solution. We employ a Fully Connected

Network to combine the sentiment and stock price estimates. The Fully Connected Network produces a final stock prediction.

To assess the performance of the proposed solution, we must combine the FS dataset described in section 3.4 with the ACIM dataset described in section 3.1. The FS dataset contains the financial stock data for four stocks, namely $AAPL, $CSCO, $IBM, and $MSFT for the period between 2015-01-01 and 2019-12-31. The ACIM contains twitter messages that reference these stocks during the same period. Similar to the FSD model, we use the Mean Absolute Percentage Error (MAPE) (Equation 3) and Accuracy (Equation 4) to assess the performance of the SFD model (see Chapter 4 section 4.2 for more details).

The timestamps for the entries in the FS dataset are coarse grained and increase in one-hour increments, without additional information about minutes or seconds. However, the timestamps for the entries in the ACIM dataset are finer and correspond to the exact time when the message was posted on Twitter (down to the second). Therefore, to associate a sentiment information with every entry in the FS dataset that represents an hour, we aggregate all sentiment scores for the relevant tweets during that hour. Moreover, we collect the sentiment scores during closed hours and associate them with the FS dataset entry for the opening hour of the next day. For example, since the stock market is only open between the hours of 9:30 AM and 4:00 PM, sentiment scores retrieved from tweets between the hours of 4:00 PM and 9:30 AM of the next day will be associated with the FS dataset entry for 9:30 AM. Moreover, for weekends, since the stock market is closed, sentiment scores calculated from Friday at 4:00 PM to Monday at 9:30 AM are associated with the FS data for Monday at 9:30 AM. The same concept applies to holidays when the stock market is

closed. If we cannot retrieve any relevant tweets to calculate sentiment information during an hour, then we apply the sentiment information of the previous hour as a simple extrapolation mechanism.

The features used as the input into the fusion Fully Connected Network are the Standard Deviation Sentiment Score, Mean Sentiment Score, Median Sentiment Score, Minimum Sentiment Score, Maximum Sentiment Score, and the FS data features. The Fully Connected Network is appended to the LSTM layers while also taking the five sentiment score outputs from the ESE model. The input to the model is considered a time-series data and hence we need to use time-series cross-validation for the combined data. The details of the hyperparameters for the Fully Connected Network are shown in Table 9. In figures 30 to 37 we show the model loss and the 'prediction vs real stock price' for each of the $AAPL, $CSCO, $IBM, and $MSFT stocks using the combination of the ESE model and FSD model.

Table 9. MLP Fully Connected Network Hyperparameters.

Architecture Details	**MLP Fully Connected Network**	
Hidden Layers	2	
Nodes	128	64
Activation Function	Rectified Linear Activation Function	
Input Dropout Layer	0	
Dropout Layer	0.175	
L2 Regularization	0.0025	

Epochs	275
Batch Size	25
Learning Rate	0.0004
Optimizer	Adam
Loss Function	Mean Square Error (MSE)

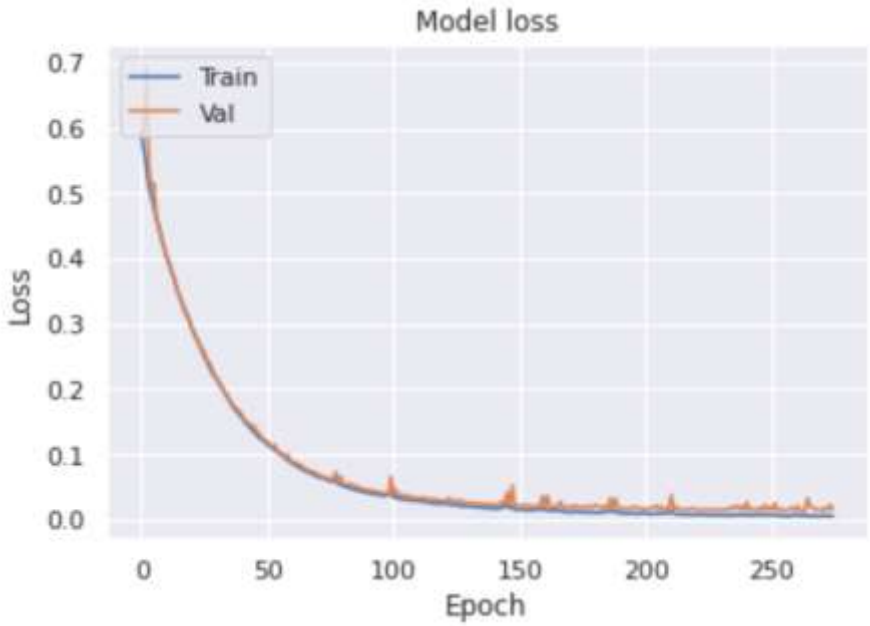

Figure 30. $AAPL Fully Connected Network Model Loss

Figure 31. $AAPL Fully Connected Network Prediction vs Real Stock Price

$AAPL MAPE Accuracy: 1.0717613

$AAPL Prediction vs Real Stock Price Performance: 0.7453473

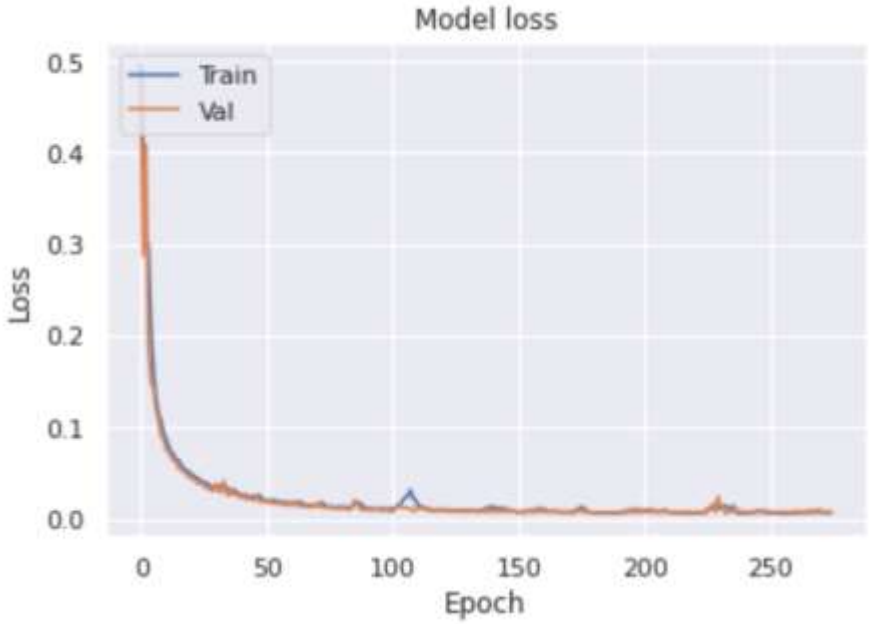

Figure 32. $CSCO Fully Connected Network Model Loss

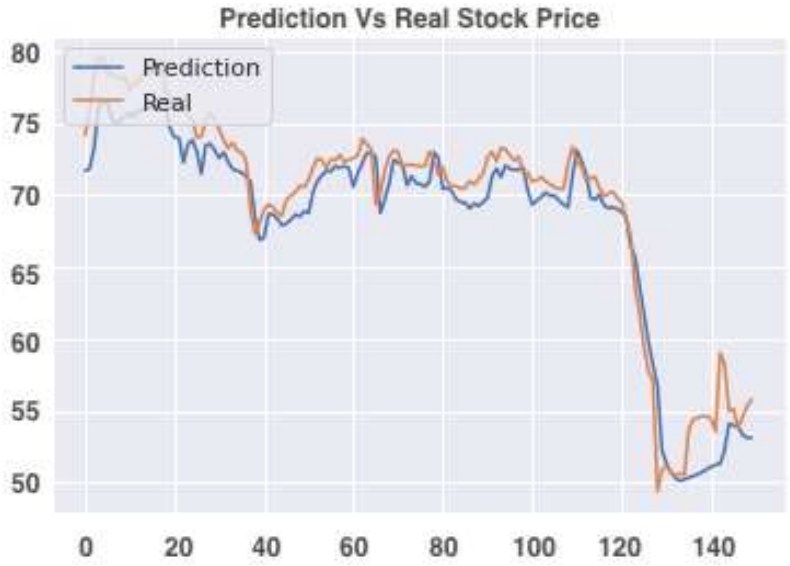

Figure 33. $CSCO Fully Connected Network Prediction vs Real Stock Price

$CSCO MAPE Accuracy: 1.0075246

$CSCO Prediction vs Real Stock Price Performance: 0.7445013

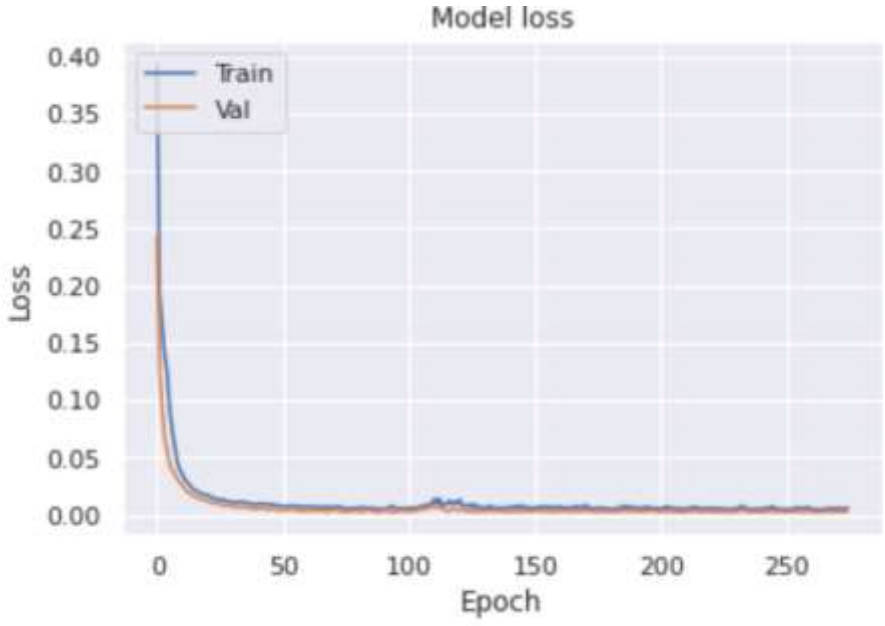

Figure 34. $IBM Fully Connected Network Model Loss

Figure 35. $IBM Fully Connected Network Prediction vs Real Stock Price

$IBM MAPE Accuracy: 1.1634728

$IBM Prediction vs Real Stock Price Performance: 0.7417577

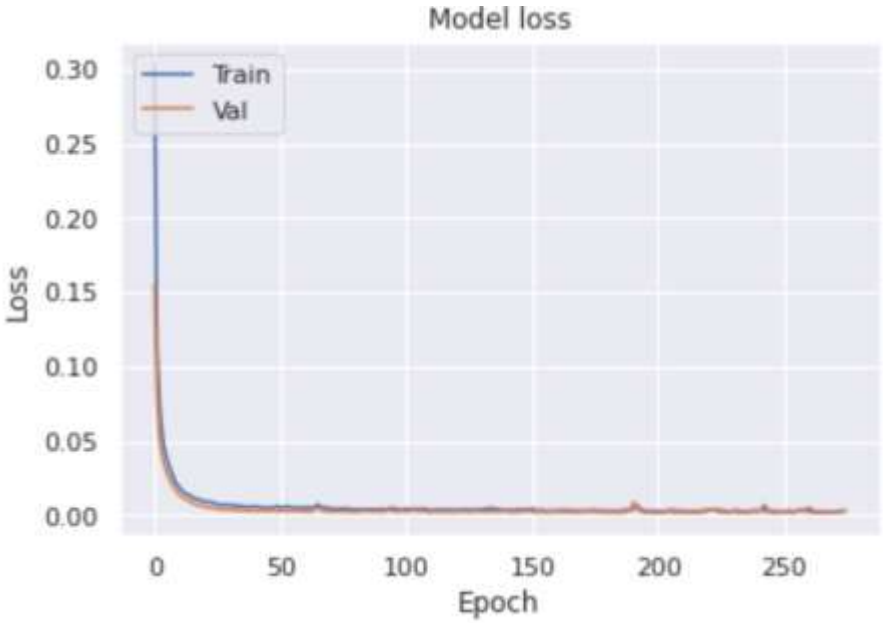

Figure 36. $MSFT Fully Connected Network Model Loss

Figure 37. $MSFT Fully Connected Network Prediction vs Real Stock Price

$MSFT MAPE Accuracy: 1.0750703

$MSFT Prediction vs Real Stock Price Performance: 0.7429589

Table 10. Final Average Result of The Fully Connected Network

	Accuracy Score
$APPL	**0.745**
$CSCO	**0.744**
$IBM	**0.741**
$MSFT	**0.742**
Final Average Accuracy	**0.743 (74.3 percent)**
Standard Deviation	**0.0434**
MAPE	**1.079**

After running the 10-fold MAPE accuracy and 10-Fold prediction vs real stock price performance, we achieved a MAPE average accuracy of 1.079 and a 10-Fold prediction vs real stock price performance of 0.743 (74.3 percent) as seen in Table 10. The standard deviation score was also 0.0434 as seen in Table 10. We can clearly observe that there is an increase in the accuracy when using the combination of both the ESE model and the FSD model. The FSD model achieved an average accuracy of 68.5%, while the combination of the ESE model and the FSD model achieved an average accuracy of 74.3%. We also observe that there is a reduction in the MAPE standard deviation which indicates the prediction error has decreased. The increased accuracy and the reduction of the MAPE standard deviation clearly show that exploiting the sentiment score with the combination of the FS data increased the stock market prediction performance.

4.4 Model Evaluation

We evaluate the proposed SFD model solution against existing state-of-the-art models that have been presented in the literature. To ensure a fair comparison, we apply our solution to the same dataset employed by the work we compare with. We summarize our results in Table 11.

Table 11. Model Evaluation Against Other Models.

Author	Model	Dataset	Accuracy	Our Proposed SFD Model Accuracy	Notes
Xiang et al.	(LR, SVM, CART, RF, BN, and XGB) and six DNN models (MLP, DBN, SAE, RNN, LSTM, and GRU)	S&P 500 Index Fund	50%	67%	Reducing the timeframe to be between the years 2015-2019 in our model achieves an accuracy of 81%.

Dey et al.	XGBoost (eXtreme Gradient Boosting)	Yahoo and Apple stocks. Dates were not specified but used timeframes of 28 days, 60 days, and 90 days.	85% - 99%	Repeating similar time frames, the proposed model achieved 94% on the Yahoo and Apple stocks.
Pagolu et al.	Random Forest	Microsoft, between the dates of August 2015 and August 2016.	70.1%	84.7%

The results shown in Table 11 prove that the proposed model achieves superior results by benefiting from the sentiment score from social media and the FS data through the features and models used.

4.5 Research Answers

Through the results from chapter 4 (section 4.3), we were able to answer our questions related to stock price predictions that we announced in chapter 1 (section 1.4) as follows:

1. **Can we improve stock price predictions using sentiment analysis of social media messages compared to estimates solely based on financial data?**

   Our results from chapter 4 (section 4.3) have successfully shown that combining sentiment analysis from social media messages with the FS data modeling improved our prediction performance. Through the use of the FSD model alone we were able to achieve a 68.5% prediction vs real price performance. However, when we combined the ESE model with the FSD model, we were able to achieve a 74.3% prediction vs real price performance. Hence, stock price estimation using sentiment analysis from social media messages improves the stock price estimation.

2. **Can we achieve superior performance compared to the state-of-the-art methods in stock price estimation using deep learning techniques?**

   Our results from chapter 4 (section 4.4) have successfully shown that employing a combination of deep learning techniques that are trained to estimate stock prices from a variety of text (sentiment) and financial stock features can improve prediction performance. We evaluated our proposed SFD model against existing state-of-the-art models that have been presented in the literature. Our SFD model results achieved an increase of up to 17% in price prediction performance against existing state-of-the-art models.

4.6 Chapter Summary

This chapter examined the proposed models for sentiment estimation and the stock prediction. The proposed model predicted the sentiment score from retail investors extracted using Twitter data. To extract the sentiment score from Twitter data, we used an ensembled-based model that uses the power of MLP, LSTM, and CNN. The MLP consisted of two different models, the MLP Feature Driven and MLP Simple Word Embedding models. For the stock prediction, the model consisted of an LSTM model. The ESE model and the FSD model are then combined into a fusion Fully Connect Network which provided the final prediction. The optimal hyperparameters that were determined for all the models were found through trial and error. Through the results achieved and evaluation against existing state-of-the-art models in this field, the proposed model achieves far superior results.

Chapter 5: Conclusion and Future Works

5.1 Research Summary

Predicting the prices of stocks and identifying the financial variables that can influence them can be a challenging task, especially due to the volatile nature of the stock market (Kiersz, 2015). Studies have shown that the EMH (Efficient Market Hypothesis) theory and the random walk hypothesis of the stock market may be incorrect (Kiersz, 2015). The EMH theory states that the stock market does not follow a linear trend but can be very random, which impedes its prediction (Cao & Tay, 2000). As a result, researchers are constantly trying to find an accurate model that can predict the stock market (Fama, 1965). Developing a model that can predict the market and result in higher returns than expected will disprove the EMH theory and increase profits on financial investments. The EMH theory does not explain the constant fluctuations in stock market pricing during short-term periods because the EMH theory assumes that all information regarding a stock are factored into the price and no amount of analysis can allow an investor to predict the stock market (Thune, 2020). These fluctuations may be the result of investors buying or selling a certain stock due to news articles and social media reactions. Stock market prices do not follow constant trends but are rather dynamic and can be affected by news, social media, politics, investor emotions and the general economy (Bezerra & Albuquerque, 2017).

In the past, predicting the stock market was done using average volume movements, distinct analysis, autoregressive models and correlations (Kumar & Thenmozhi, 2014). Recently, researchers began using artificial intelligence to recognize patterns and random samples of variables that affect the stock market behavior through deep learning and machine learning

algorithms (Wang, Wang, Zhang, & Guo, 2012). Machine learning is used frequently in predictive models to analyze complex patterns. Due to the challenging nature of predicting the stock market, testing different methods of machine learning algorithms can allow us to produce new techniques in predicting and processing the difficult tasks and patterns of the stock market (Hsu, Lessmann, Sung, Ma, & Johnson, 2016).

The two main areas of focus in stock market research are verifying what variables need to be considered and what machine learning algorithm to be used for the prediction. Previous methods focused mostly on historical stock market price and stock specific variables without considering the investors mood on social media platforms (Ren, He, Girshick, & Sun, 2017). Using social media information to conduct sentiment analysis while still considering historical stock market price data, we can potentially improve the performance of machine learning methods that predict the stock market.

There are two well-known methods previously used for the prediction of the stock market which are technical analysis and fundamental analysis. These two methods present completely different approaches to predicting the stock market. Many of the research that has been done previously had a strong focus on either technical analysis or fundamental analysis. Many of them did not take into consideration the power in the use of both types of analysis. The technical analysis method uses the financial data to extract the mathematical indicators and takes the stock price into consideration to predict its future value. However, the technical analysis does not consider any other important factors such as investors' sentiment, politics, news, etc. These important factors are only considered in fundamental analysis.

Deep learning has been applied to numerous applications in the last decade. This has allowed machine learning research to grow tremendously. Deep learning methods such as CNN, and Long LSTM are a few examples of methods that can be used in various fields. CNN is heavily used in image processing and NLP fields. LSTM can be used in the areas of text, voice, and video data.

This research focused on a method to predict the stock market using sentiment analysis in combination with FS data to improve the stock market's prediction performance. To extract the sentiment score, an ensemble-based model that uses the power of MLP, LSTM, and CNN is used. We conducted feature engineering and selection to determine the most important features from the social media data and FS data. The model prediction was conducted on $AAPL, $CSCO, $IBM, and $MSFT, utilizing a combination of the FS data and sentiment score extracted from the social media platform Twitter between the years of 2015 to 2019. From the experiments that we conducted, we observed that the combination of the FS data and sentiment score can improve the stock market prediction performance. The proposed SFD model has achieved a prediction performance of 74.3%. The proposed SFD model achieved far superior results when evaluated against other models in this field.

5.2 Key Concepts of The Research

- We predicted the stock market prices of four companies ($AAPL, $CSCO, $IBM, $MSFT) through a combination of machine learning models.
- We trained two separate models and then combined their outputs to obtain a final prediction.
    - Ensemble Sentiment Estimation (ESE) Model

    - Financial Stock Data (FSD) Model
- We gathered the Twitter data using the official Twitter API which used a cashtag to scrape all data of investors who have Twitter to tweet about stocks.
- We conducted feature engineering and selection on the data.
    - We conducted missing value analysis, constant variable analysis, duplicated variable analysis, correlated variable analysis on various Twitter features selected.
    - We used 52 features as input into the ESE model.
- The FS dataset was gathered using the Yahoo finance package (yfinance).
- We applied feature engineering and selection on the FS dataset.
- We conducted an analysis of the stocks to study the movement of the stocks over the years.
- We used several machine learning models for the prediction of the sentiment score and the stock prices.
    - ESE Model:
        - MLP FD
        - MLP SWE
        - CNN
        - LSTM
        - MLP Stacked Ensemble
    - FSD Model:
        - LSTM

- The ESE model and the FSD model were combined using a fusion machine.
- The performance of the proposed solution improved when we combined both models. We obtained a prediction accuracy of 74.3% using the SFD model.
- We evaluated the proposed SFD model to existing state-of-the-art methods. The proposed SFD model achieved superior results.

5.3 Future Work

The models developed in this research have reflected the power of social media and its impact on stock market predictions. In future work, the addition of highly influential platforms such as Reddit can have a major effect on the accuracy of the stock market prediction. Reddit has a vast community of investors who provide in-depth analysis on stock market performance.

Another area open to further exploration is the use of a wider range of companies in the dataset. During this study, we focused on four companies: $AAPL, $CSCO, $IBM, and $MSFT, but if we were to add additional companies, we could validate the strength of the stock prediction performance.

As fascination with financial trends increases, another growing topic is crypto currency which has grown tremendously in the past few years as well as the sentiment of investors leaning towards certain crypto coins resulting in crypto price changes. Hence, we are interested in exploring the use of techniques that exploit historical crypto-currency data with the sentiment information from social media platforms for price prediction.

References

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. MIS Quarterly, 25(1), 107. https://doi.org/10.2307/3250961

Babu, M. S., Geethanjali, N., & Satyanarayana, P. B. (2012). Clustering Approach to Stock Market Prediction. International Journal Advanced Networking and Applications, 03(04), 1281–1291.

Barak, S., Arjmand, A., & Ortobelli, S. (2017). Fusion of multiple diverse predictors in stock market. Information Fusion, 36, 90–102. https://doi.org/10.1016/j.inffus.2016.11.006

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. The North American Journal of Economics and Finance, 47(0), 552–567. https://doi.org/10.1016/j.najef.2018.06.013

Bezerra, P. C. S., & Albuquerque, P. H. M. (2017). Volatility forecasting via SVR–GARCH with mixture of Gaussian kernels. Computational Management Science, 14(2), 179–196. https://doi.org/10.1007/s10287-016-0267-0

Bernal, A., Fok, S., & Pidaparthi, R. (2012). Financial Market Time Series Prediction with Recurrent Neural Networks. 1–5. https://doi.org/10.1.1.278.3606

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007

Cakra, Y. E., & Distiawan Trisedya, B. (2015). Stock price prediction using linear regression based on sentiment analysis. 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 147–154. https://doi.org/10.1109/ICACSIS.2015.7415179

Cao, L. J., & Tay, F. E. H. (2000). Feature Selection for Support Vector Machines in Financial Time Series Forecasting. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 1983, pp. 268–273). https://doi.org/10.1007/3-540-44491-2_38

Castoe, M. (2020). Predicting Stock Market Price Direction with Uncertainty Using Quantile Regression Forest. (November).

Chourmouziadis, K., & Chatzoglou, P. D. (2016). An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. Expert Systems with Applications, 43, 298–311. https://doi.org/10.1016/j.eswa.2015.07.063

Di Persio, L., & Honchar, O. (2017). Recurrent neural networks approach to the financial forecast of Google assets. International Journal of Mathematics and Computers in Simulation, 11, 7–13. Retrieved from https://iris.univr.it/retrieve/handle/11562/959057/66085/Recurrent neural networks approach to the financial forecast of Google assets Di Persio.pdf

Confusion matrix: What is confusion matrix: Confusion matrix for DS. Analytics Vidhya. (2021, July 1). Retrieved October 8, 2021, from https://www.analyticsvidhya.com/blog/2021/05/in-depth-understanding-of-confusion-matrix/.

Fama, E. F. (1965). The Behavior of Stock-Market Prices. The Journal of Business, 38(1), 34. https://doi.org/10.1086/294743

Hossain, M. A., Karim, R., Thulasiram, R., Bruce, N. D. B., & Wang, Y. (2018). Hybrid Deep Learning Model for Stock Price Prediction. 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 1837–1844. https://doi.org/10.1109/SSCI.2018.8628641

Hsu, M.-W., Lessmann, S., Sung, M.-C., Ma, T., & Johnson, J. E. V. (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. Expert Systems with Applications, 61, 215–234. https://doi.org/10.1016/j.eswa.2016.05.033

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. Computers & Operations Research, 32(10), 2513–2522. https://doi.org/10.1016/j.cor.2004.03.016

Kalyanaraman, V., Kazi, S., Tondulkar, R., & Oswal, S. (2014). Sentiment Analysis on News Articles for Stocks. 2014 8th Asia Modelling Symposium, (September 2014), 10–15. https://doi.org/10.1109/AMS.2014.14

Kara, Y., Acar Boyacioglu, M., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. Expert Systems with Applications, 38(5), 5311–5319. https://doi.org/10.1016/j.eswa.2010.10.027

Kiersz, A. (2015). Warren Buffett Berkshire Hathaway vs SP 500. Retrieved October 28, 2019, from http://www.businessinsider.com/warren-buffett-berkshire-hathaway-vs-sp-500-2015-3?IR=T

Kumar, M., & Thenmozhi, M. (2014). Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models. International Journal of Banking, Accounting and Finance, 5(3), 284. https://doi.org/10.1504/IJBAAF.2014.064307

Laboissiere, L. A., Fernandes, R. A. S., & Lage, G. G. (2015). Maximum and minimum stock price forecasting of Brazilian power distribution companies based on artificial neural networks. Applied Soft Computing, 35, 66–74. https://doi.org/10.1016/j.asoc.2015.06.005

Lee, H., Surdeanu, M., MacCartney, B., & Jurafsky, D. (2014). On the importance of text analysis for stock price prediction. Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014, (2009), 1170–1175.

Lv, D., Yuan, S., Li, M., & Xiang, Y. (2019). An Empirical Study of Machine Learning Algorithms for Stock Daily Trading Strategy. Mathematical Problems in Engineering, 2019(2), 1–30. https://doi.org/10.1155/2019/7816154

Markowska-Kaczmar, U., & Dziedzic, M. (2008). Discovery of Technical Analysis Patterns. 2008 International Multiconference on Computer Science and Information Technology, 3, 195–200. https://doi.org/10.1109/IMCSIT.2008.4747239

Milosevic, N. (2016). Equity forecast: Predicting long term stock price movement using machine learning. https://doi.org/10.1453/jel.v3i2.750

Mittal, a, & Goel, A. (2012). Stock Prediction Using Twitter Sentiment Analysis. Tomx.Inf.Elte.Hu, (June). Retrieved from http://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf

Moat, H. S., Curme, C., Avakian, A., Kenett, D. Y., Stanley, H. E., & Preis, T. (2013). Quantifying Wikipedia Usage Patterns Before Stock Market Moves. Scientific Reports, 3(1), 1801. https://doi.org/10.1038/srep01801

Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. Expert Systems with Applications, 42(24), 9603–9611. https://doi.org/10.1016/j.eswa.2015.07.052

Pagolu, V. S., Reddy, K. N., Panda, G., & Majhi, B. (2016). Sentiment analysis of Twitter data for predicting stock market movements. 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), 1345–1350. https://doi.org/10.1109/SCOPES.2016.7955659

Peachavanish, R. (2016). Stock selection and trading based on cluster analysis of trend and momentum indicators. Lecture Notes in Engineering and Computer Science, 1, 317–326.

Powell, N., Foo, S. Y., & Weatherspoon, M. (2008). Supervised and Unsupervised Methods for Stock Trend Forecasting. 2008 40th Southeastern Symposium on System Theory (SSST), 203–205. https://doi.org/10.1109/SSST.2008.4480220

Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news. ACM Transactions on Information Systems, 27(2), 1–19. https://doi.org/10.1145/1462198.1462204

Shah, D., Isah, H., & Zulkernine, F. (2019). Stock Market Analysis: A Review and Taxonomy of Prediction Techniques. International Journal of Financial Studies, 7(2), 26. https://doi.org/10.3390/ijfs7020026

Tiwari, S., Pandit, R., & Richhariya, V. (2006). Predicting future trends in stock market by decision tree rough-set based hybrid system with HHMM. Expert Systems with Applications, 30(4), 605–611. https://doi.org/10.1016/j.eswa.2005.07.006

Wang, J.-J., Wang, J.-Z., Zhang, Z.-G., & Guo, S.-P. (2012). Stock index forecasting based on a hybrid model. Omega, 40(6), 758–766. https://doi.org/10.1016/j.omega.2011.07.008

Weng, B., Ahmed, M. A., & Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. Expert Systems with Applications, 79, 153–163. https://doi.org/10.1016/j.eswa.2017.02.041

Wu, K. P., Wu, Y. P., & Lee, H. M. (2014). Stock trend prediction by using k-means and aprioriall algorithm for sequential chart pattern mining. Journal of Information Science and Engineering, 30(3), 653–667. https://doi.org/10.6688/JISE.2014.30.3.7

Xiao, Y., Xiao, J., Lu, F., & Wang, S. (2014). Ensemble ANNs-PSO-GA Approach for Day-ahead Stock E-exchange Prices Forecasting. International Journal of Computational Intelligence Systems, 7(2), 272–290. https://doi.org/10.1080/18756891.2013.864472

Xu, Y., & Cohen, S. B. (2018). Stock Movement Prediction from Tweets and Historical Prices. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1, 1970–1979. https://doi.org/10.18653/v1/P18-1183

Chourmouziadis, K., & Chatzoglou, P. D. (2016). An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. Expert Systems with Applications,43, 298-311. doi:10.1016/j.eswa.2015.07.063

Hao, K. (2020, April 02). We analyzed 16,625 papers to figure out where AI is headed next. Retrieved from https://www.technologyreview.com/2019/01/25/1436/we-analyzed-16625-papers-to-figure-out-where-ai-is-headed-next/

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence,39(6), 1137-1149. doi:10.1109/tpami.2016.2577031

Kim, H., & Jeong, Y. (2019). Sentiment Classification Using Convolutional Neural Networks. Applied Sciences, 9(11), 2347. doi:10.3390/app9112347

K. Ravichandran, S. Arulchelvan and K. PeriyaKannan, "Perception of medical awareness of media analyzed by multilayer perceptron," Journal of Media and Communication Studies, vol. 10, no. 10, pp. 118- 127, 2018, doi: 10.5897/jmcs2018.0627.

A. Victor Devadoss and T. Antony Alphonnse Ligori, "Forecasting of Stock Prices Using Multi Layer Perceptron," International Journal of Web Technology, vol. 002, no. 002, pp. 52-58, 2013, doi: 10.20894/ijwt.104.002.002.006.

X. Jiang, K. Xu and T. Sun, "Action Recognition Scheme based on Skeleton Representation with DS-LSTM Network," IEEE Transactions on Circuits and Systems for Video Technology pp. 1-1, 2019, doi: 10.1109/tcsvt.2019.2914137.

F. Audrino, F. Sigrist and D. Ballinari, "The impact of sentiment and attention measures on stock market volatility," International Journal of Forecasting, vol. 36, no. 2, pp. 334-357, 2020, doi: 10.1016/j.ijforecast.2019.05.010.

M. Jiang, M. Lan and Y. Wu, “ECNU at SemEval-2017 Task 5: An Ensemble of Regression Algorithms with Effective Features for Fine- Grained Sentiment Analysis in Financial Domain,” in Proc. of the 11th International Workshop on Semantic Evaluation (SemEval- 2017), 2017, pp. 888-893, doi: 10.18653/v1/S17-2152.

A. Picasso, S. Merello, Y. Ma, L. Oneto and E. Cambria, "Technical analysis and sentiment embeddings for market trend prediction", Expert Systems with Applications, vol. 135, pp. 60-70, 2019, doi: 10.1016/j.eswa.2019.06.014.

M. Shastri, S. Roy and M. Mittal, "Stock Price Prediction using Artificial Neural Model: An Application of Big Data", ICST Transactions on Scalable Information Systems, vol. 0, no. 0, p. 156085, 2018, doi: 10.4108/eai.19-12-2018.156085.

E. Al-Mansouri, “Using Artificial Neural Networks and Sentiment Analysis to Predict Upward Movements in Stock Price,” B.S. thesis, Dept. Computer Science, Worcester Polytechnic Inst., Worcester, Massachusetts, 2016.

GitHub, 2014. [Online]. Available: https://github.com/coastalcph/cs_sst/blob/master/data/res/emnlp_dict.t xt. [Accessed: 16- Sep- 2019].

Qi, P., Dozat, T., Zhang, Y., & Manning, C. D. (2018). Universal Dependency Parsing from Scratch. Proceedings of the. doi:10.18653/v1/k18-2016

Nielsen, F. Å. (2011, March 15). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. Retrieved from https://arxiv.org/abs/1103.2903

Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment Analysis of Short Informal Texts. Journal of Artificial Intelligence Research,50, 723-762. doi:10.1613/jair.4272

Zhu, X., Kiritchenko, S., & Mohammad, S. (2014). NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). doi:10.3115/v1/s14-2077

"Guide to General Inquirer Category Listings", Wjh.harvard.edu, 2000. [Online]. Available: http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm. [Accessed: 01- Oct- 2019].

Figure 9: Positive and negative polarity assigned by a lexical resource, SentiWordNet (Baccianella, Esuli & Sebastiani, 2010), to words within a negative and positive topic models. (n.d.). doi:10.7717/peerj-cs.252/fig-9

"Google Code Archive - Long-term storage for Google Code Project Hosting.", Code.google.com, 2013. [Online]. Available: https://code.google.com/archive/p/word2vec/. [Accessed: 05- Oct- 2019].

"SemEval-2017 Task 5", Alt.qcri.org, 2017. [Online]. Available: http://alt.qcri.org/semeval2017/task5/. [Accessed: 18- Aug- 2019].

Xiang, C., & Fu, W. M. (2006). Predicting the Stock Market using Multiple Models. 2006 9th International Conference on Control, Automation, Robotics and Vision. doi:10.1109/icarcv.2006.345431

Dey, S., Kumar, Y., Saha, S., & Basak, S. (2016). Forecasting to Classification : Predicting the direction of stock market price using Xtreme Gradient Boosting Forecasting to Classification : Predicting the direction of stock market price using Xtreme Gradient Boosting. (October), 1–10.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly: Management Information Systems, 28(1), 75–105. https://doi.org/10.2307/25148625

Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, Samir Chatterjee (2008). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, vol. 24 no. 3, pp. 45–77.

Gholamalinezhad, H., & Khosravi, H. (2020). Pooling Methods in Deep Neural Networks, a Review. Retrieved from http://arxiv.org/abs/2009.07485

Turney, P. D., & Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems, 21(4), 315–346. https://doi.org/10.1145/944012.944013

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation Functions: Comparison of trends in Practice and Research for Deep Learning. 1–20. Retrieved from http://arxiv.org/abs/1811.03378

Rojas-Barahona, L. M. (2016). Deep learning for sentiment analysis. Language and Linguistics Compass, 10(12), 701–719. https://doi.org/10.1111/lnc3.12228

Posted by Kamal Jain on August 7, 2. A. (n.d.). Sentiment Analysis using Deep Learning. Retrieved from https://www.datasciencecentral.com/profiles/blogs/sentiment-analysis-using-deep-learning

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. Journal of Big Data, 7(1), 20. https://doi.org/10.1186/s40537-020-00299-5

Appendix A. The procedure to clean the social media data

The procedure to clean the social media data consists of:

- HTML encodings removal.
    - E.g. &amp, $quot, etc.
- Removal of user ID mentions within the messages.
- Removal of retweet symbols will need to be removed from sentences.
- Conversion of URL to '_url'.
- Correction of abbreviations using the slang dictionary (NLP dictionary) corpus.
- Removal of common word contractions.
- Correction of spelling mistakes.
- Correction of exaggerated/erroneous word elongations.
    - E.g. Victoryyyyy to Victory, callendar to calendar, etc.
- Verify if numeric values are floats or not.
- Stock symbols joined with its word (padding).
    - E.g. $ AAPL to $AAPL
- HTTP joined (padding) to a URL.
- WWW will need to be joined (padding) to URL.
- Removal of unnecessary apostrophes.
- Conversion of Ordinal words into its ordinal number.
    - E.g. first to 1st, second to 2nd, #1 to one, one to first, #1 to first, # first to 1st.
- Punctuation split from last word in a sentence.

Appendix B. Complete List of Social Media Data Features

- Cashtag
- Conversation Parent
- Conversation Replies
- Created at (Time)
- Liked By Self
- Official Account
- Sentiment
- Sentiment Score
- Source
- Spans
- Text
- Total Likes
- Clean Text
- Base Text
- POS-Tag features
    - Part of Speech Tags: ‘VB’, ‘VBD’, ‘VBG’, ‘VBN’, ‘VBP’, ‘VBZ’, ‘VM’
- Binary feature ‘+num’
- Binary feature ‘-num’
- Binary feature ‘num%’
- Binary feature ‘+num%’
- Binary feature ‘-num%’
- Binary feature ‘$num’
- Binary feature mixed number and word
- Binary feature ordinal number
- Binary feature ‘num-num’
- Binary feature ‘num-num%’
- Binary feature ‘ num-num-num’
- Binary feature ‘num/num’
- Binary feature ‘num/num/num’
- Binary feature only numbers (No symbol and characters)
- Binary feature ‘call’ or ‘calls’ or ‘called’ before ‘+num%’
- Binary feature ‘call’ or ‘calls’ or ‘called’ before ‘-num%’
- Binary feature ‘put’ or ‘puts’ before ‘+num%’
- Binary feature ‘put’ or ‘puts’ before ‘-num%’
- Binary feature ‘Bull’ or ‘Bullish’
- Binary feature ‘Bear’ or ‘Bearish’
- Calculate number of ‘!’
- Calculate number of ‘?’

- Calculate number of '$'
- Calculate number of continuous '!'
- Calculate number continues '?'
- Calculate number of capital words
- Calculate number hashtags
- AFINN Sentiment Lexicon
    - Sum Score
    - Max Score
    - Min Score
    - Ratio of positive words
    - Ratio of negative words
- Bing Liu Sentiment Lexicon
    - Ratio of positive words
    - Ratio of negative words
- General Inquirer Sentiment Lexicon
    - Ratio of positive words
    - Ratio of negative words
- NRC Hashtag Sentiment Lexicon
    - Sum Score
    - Max Score
    - Min Score
    - Ratio of positive words
    - Ratio of negative words
- SentiWordNet Sentiment Lexicon
    - Sum Score
    - Max Score
    - Min Score
    - Ratio of positive words
    - Ratio of negative words
- TF-IDF n-gram features
    - Average TF-IDF 1-gram
    - Average TF-IDF 2-gram
    - Average TF-IDF 3-gram
    - Average TF-IDF 4-gram
- Average RF n-gram features
    - Average RF 1-gram
    - Average RF 2-gram
    - Average RF 3-gram
    - Average RF 4-gram
- PMI Score

Appendix C. List of Features with Missing Values

Below is a list with features containing missing values:

- Analysis of `'conversation_parent'`

Table 12. Results from missing value analysis of 'Conversation_Parent'

Attribute	Value
count	191.000000
mean	0.776546
std	0.4376841
min	0.000000
25%	0.000000
50%	1.000000
75%	1.000000
max	1.000000

.

The analysis indicated the Conversation_parent should be removed because there were many missing values.

- Analysis of `'conversation_replies'`

Table 13. Results from missing value analysis of 'Conversation_Replies'

Attributes	Values
count	908.000000
mean	0.011553
std	0.053976

min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

The analysis showed that the ‘Conversation_replies’ missing values will have a value of -1, which is an indication of a very negative sentiment.

- Analysis of `'liked_by_self'`

Table 14. Results from missing value analysis of ‘liked_by_self’

Attribute	Values
count	956.0
mean	0.0
std	0.0
min	0.0
25%	0.0
50%	0.0
75%	0.0
max	0.0

The results of the analysis show that ‘liked_by_self’ should be removed due to many values missing.

- Analysis of `'official_account'`

Table 15. Results from missing value analysis of 'official_account'

Attributes	Values
count	979.000000
mean	0.176253
std	0.359782
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

The analysis showed that the 'official_account' missing values will have a value of -1, which is an indication of a very negative sentiment.

- Analysis of `'sentiment'`

Table 16. Results from missing value analysis of 'sentiment'

Attributes	Values
count	182
unique	2
top	bullish
freq	123

The sentiment feature showed that the bullish will have a value of 1, bearish will have a value of -1, and if not either then it will have a value of 0 because not-a-number values are neutral.

- Analysis of `'total_likes'`

Table 17. Results from missing value analysis of 'total_likes'

Attributes	Values
count	912.000000

mean	0.047871
std	0.125650
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

The analysis showed that the ‘total_likes’ missing values will have a value of -1, which is an indication of a very negative sentiment.

- Analysis of `'SentiWordNet_max_score'`

Table 18. Results from missing value analysis of ‘SentiWordNet_max_score’

Attributes	Values
count	1390.000000
mean	0.256518
std	0.290000
min	-0.750000
25%	0.000000
50%	0.125000
75%	0.375000
max	1.000000

The results above show that the ‘SentiWordNet_max_score’ missing values will have a neutral value of 0 if a word in the sentence is not in the SentiWordNet synsets.

- Analysis of `'SentiWordNet_min_score'`

Table 19. Results from missing value analysis of 'SentiWordNet_min_score'

Attributes	Values
count	1390.000000
mean	-0.184872
std	0.208427
min	-1.000000
25%	-0.375000
50%	0.000000
75%	0.000000
max	0.500000

The results above show that the 'SentiWordNet_min_score' missing values will have a neutral value of 0 if a word in the sentence is not in the SentiWordNet synsets.

- Analysis of `'Avg_TFIDF_1-grams'`

Table 20. Results from missing value analysis of 'Avg_TFIDF_1-grams'

Attributes	Values
count	1378.000000
mean	5.722852
std	0.756193
min	1.132958
25%	4.463252
50%	5.361894
75%	5.251906
max	15.045454

All TF-IDF missing values will have a value of 0 because there aren't any words that pass the n-gram filter. The value 0 is considered a neutral value.

Given that there is an abundant number of missing values, ‘Conversation_parent’ will be removed. Similarly, ‘Conversation_replies’ missing values will have a value of -1 instead. ‘Liked_by_self’ have also been removed, given that there are numerous missing values. Missing values for ‘Official_account’ will have a value of -1 instead. The sentiment feature will have ‘bullish’ with a value of 1, ‘bearish’ with value of -1, and a value of 0 otherwise, given that not-a-number values are neutral. ‘Total_likes’ missing values will have a value of -1. When a word in the sentence is not in the SentiWordNet synsets, a neutral value of 0 is given for ‘SentiWordNet_max_score’ missing values. The same procedure is adapted for ‘SentiWordNet_min_score’ missing values. Finally, all TF-IDF missing values will have a value of 0, given that there are no words that pass the n-gram filter, hence being considered a neutral value.