Machine learning based sentiment analysis for stock movement prediction

Ishmael Grech

Supervisor: Mr Thomas Gatt

June, 2022

A dissertation submitted to the Institute of Information and Communication Technology in partial fulfillment of the requirements for the degree of B.Sc. (Hons.) Software Development

Authorship Statement

This dissertation is based on the results of research carried out by myself, is my own composition, and has not been previously presented for any other certified or uncertified qualification.

The research was carried out under the supervision of Mr Thomas Gatt.

Ishmael Grech

June 5, 2022

Copyright Statement

In submitting this dissertation to the MCAST Institute of Information and Communication Technology I understand that I am giving permission for it to be made available for use in accordance with the regulations of MCAST and the Library and Learning Resource Centre. I accept that my dissertation may be made publicly available at MCAST's discretion.

Ishmael Grech

June 5, 2022

Acknowledgements

To start with, I wish to express my appreciation to my mentor and supervisor, Mr Thomas Gatt, for his patience, support and knowledge throughout this dissertation.

I also wish to thank my family, friends and colleagues for the encouragement, support and motivation they provided me along the way.

Table of Contents

List of Figures

List of Tables

Abstract

An attempt to forecast the future value of a stock sector or an individual stock is known as a stock market prediction. A machine learning based stock price prediction using sentiment analysis is outlined in this study. Sentiment scores are generated by integrating Twitter and financial news headlines with a pre-trained sentiment analyser tool. The sentiment scores are combined with historical data of tickers from the S&P 500 index. This data is used to train two types of models. The models used are Long Short Term Memory (LSTM) and Support Vector Regression (SVR), where the goal is to predict future stock prices. The results show that the proposed models cannot consistently predict stock price movement, with the highest mean directional accuracy (MDA) score of 0.57. The findings highlighted in Chapter 4.5 show that the methods explored in this research produced the lowest MAPE score of 1.05. However, this research also shows that the proposed models that used sentiment scores had less impact when compared to other studies. Additionally, it was determined that sentiment is considered an important factor in stock prediction as it positively affected the investment strategy by producing an additional profit of 1.96% for the LSTM model and 12.63% for the LinearSVR.

Chapter 1

Introduction

The process of attempting to forecast the future value of financial instruments or business stocks that are traded on an exchange is known as a stock market prediction. A successful forecast of future stock prices can result in a significant profit. The methodologies of stock prediction are divided into three main categories: technical analysis, fundamental analysis, and machine learning. In recent years, integration of text mining with stock prediction using machine learning has increased in popularity. The primary data sources of text mining are social media and financial articles. This study aims to forecast stock prices with the use of publicly available data such as social media, financial news articles and historical data.

1.1 Motivation

The main objective of stock market prediction is to predict the future movement of a financial exchange's stock value. Investors will be able to make more profit if the model can accurately predict stock price movement. Central banks worldwide use investment as one of the primary sources of income. Therefore, a robust model that accurately predicts stock prices can provide national economic stability.

1.2 Problem Definition

Macroeconomics is a branch of economics which studies the performance, behaviour, structure and decision making of the whole economy. The stock market plays a primary role in a nation's macroeconomy. Stock price forecasting is complex and challenging due to the unpredictable fluctuations and volatility of the stock market. Theories such as the efficient market hypothesis (EMH) and the random walk theory state that stock prices cannot be predicted with the use of past data, thus making the prediction problematic (Batra & Daudpota 2018). However, throughout the years, algorithmic trading has increased in popularity (*Staff Report on Algorithmic Trading in U.S. Capital Markets* 2020).

1.3 Aims and Objectives

Artificial intelligence can process better mathematical operations than humans, making it a capable approach to stock prediction. The primary data used for stock prediction in relation to machine learning is historical data (Batra & Daudpota 2018, Nelson et al. 2017, Lakshminarayanan & McCrae 2019). The historical data of stocks can be integrated with sentiment analysis to contribute to stock prediction. To achieve the objectives of this study, the following research questions have been set:

1. Which techniques are appropriate for stock price predictions?

2. Can stock price movement be predicted consistently?

3. Is sentiment an important factor?

These research questions aim to study key factors in stock prediction. The primary objective of this research is to forecast stock prices effectively. Additionally, with the use of sentiment analysis, experiments will be conducted to analyse the importance of

sentiment in the stock market. The results of the experiments will be evaluated and compared to other researchers.

1.4 Proposed Solution

The solution proposed in this study is a deep learning based model with the integration of sentiment analysis for stock prediction. The model will be trained to predict the stock prices of several stock tickers based on the dataset given. Sentiment analysis will be integrated into the dataset consisting of tweets and financial news headlines of a set of tickers. Several metrics will be used to evaluate the model performance. Changes in these metrics indicate the performance of the model. Additionally, a model that is not deep learning based will be used to compare and contrast various techniques in stock price predictions.

In a real-life scenario, a model that can consistently predict stock prices can yield more profits to investors. This study aims to contribute to the existing stock prediction methods by improving the methodology's efficacy.

1.5 Document Structure

This document is divided into four sections. The first section of this study outlines studies of several researchers and the techniques employed in the subject area. The second section uses the research gathered from the first section and is applied to achieve the research's objectives. The third section discusses the evaluation metrics and the presentation of the results. Lastly, the fourth section examines the limitations encountered, along with recommendations.

Section 1 – Literature Review

Chapter 2: This chapter covers machine learning techniques that researchers use for stock prediction. An overview of natural language processing and sentiment analysis is also provided.

Section 2 – Research Methodology

Chapter 3: This chapter describes the implementation of the techniques used to develop the stock prediction models. The data-gathering methods used for this study are also included in this chapter.

Section 3 – Analysis of Results and Discussion

Chapter 4: This chapter contains a detailed description of the evaluation metrics used along with the results of the previous methods outlined in Chapter 3. Additionally, an evaluation in contrast to other studies is outlined in this chapter.

Section 4 – Conclusions and Recommendations

Chapter 5: This chapter provides a summary of the study. The limitations encountered and recommendations for future work are also presented in this chapter. Furthermore, a closing statement concludes and answers the research questions defined in previous chapters.

Chapter 2

Literature Review

2.1 Introduction

Due to the volatility and unpredictability of the stock market, researchers struggle to find an adequate solution to forecast price movement. Traditional stock price prediction methods involve a statistical approach such as fundamental and technical analysis. However, machine learning approaches are proven to be more accurate and efficient in predicting price movement than traditional methods (Bhattacharjee & Bhattacharja 2019). Multiple techniques are used in machine learning for stock movement prediction. Deep learning techniques are frequently used due to the ability to maintain a large amount of data. Long short-term memory (LSTM) is a type of artificial recurrent neural network (RNN) that can classify and process information to make a prediction based on time series data. Research on stock prediction is typically based on three primary data sources: financial news, historical stock information and social media. A hybrid approach proves to be more efficient because all these factors affect the volatility of stocks. This section will highlight techniques used by researchers to predict stock prices and market movement.

2.2 Traditional Methods

Determining the future stock value of a business is called stock market prediction. Predicting successfully the future stock value of a company can yield considerable profits. Stock price prediction can be achieved in three standard approaches: machine learning, technical analysis, and fundamental analysis.

2.2.1 Traditional Stock Prediction

Stock markets have a complicated history of stock value predictability. The efficient-market hypothesis (EMH) claims that available information reflects the state of asset prices, implying that future prices are uncertain. Timmermann & Granger (2004) conducted an EMH study and concluded that abnormalities in trading make the stock market difficult to predict. An alternative theory known as the random walk theory states that historical stock data cannot be used for future movement prediction due to the unpredictability of the stock market. According to these theories, a stock price cannot be forecasted accurately. Other scholars, on the other hand, believe that stock prices can be predicted. Academics have tried to examine several ways to predict stock price behaviour in recent years. Despite the fact that certain theories claim price prediction is impossible, automated trading is, nevertheless, widely used around the world due to the advantages of machine arithmetic operations over human calculations. Many businesses were affected negatively by the market crash in March 2020, which furthermore resulted in large stock price fluctuations. Although the majority of businesses lost money during this market crash, companies in the sector of food, health and technology saw favourable results. Industries related to the petroleum, hospitality and entertainment sectors have seen significant declines in their market value (Mazur et al. 2021). According to Mazur et al. (2021), the market crash that caused the collapse of prices in stocks was one of the worst in history.

2.2.2 Algorithmic Trading

According to the *Staff Report on Algorithmic Trading in U.S. Capital Markets* (2020), algorithmic trading has grown at an exponential rate. United States algorithmic trading is estimated to have handled between 60 to 73% of trading orders in 2019. Technical analysis is a frequently utilised method for developing stock price prediction models. This approach forecasts future prices by using historical market data such as value and volume. With these values, several indicators can be produced to assume the stock market movement based on historical data. The simple moving average (SMA) calculates the mean of a range of values divided by the period numbers within that span. This indicator makes it simpler to recognise price trends since it smooths the stock's volatility. Research by Chong et al. (2017) found that deep neural networks (DNN) perform better than linear regression models. As a result, a DNN model is able to extract more information, resulting in improved prediction.

2.3 Machine Learning

Humans and machines have similarities: both use memory and electrical signals to transmit and retrieve data, and to reach conclusions based on data given. Machine learning is the study of how intelligent human behaviour can be imitated with the use of machines. Machine learning which is a sub field of AI tries to achieve human actions such as understanding written text, recognition of visual scenes and performing actions based on sentiment. Just like humans before classifying visual objects, or text analysing, the machine needs to learn by being supplied a data set on that specific scenario.

2.3.1 Neural Networks

A neural network is a set of algorithms that mimics the human brain's behaviour. The human brain is said to be made up of 10 billion neurons (Haykin & Network 2004). A logic gate can be compared to a neuron since both of them process input and produce output. The main benefit of neural networks is the ability to learn from given data and produce an output that is not seen in the previous input data, thus making it suitable for prediction tasks (Nygren 2004). In a study by Nygren (2004), daily stock predictions were implemented using neural networks. The model is trained using a back-propagation algorithm that uses gradient descent for supervised learning of neural networks. This technique guides the artificial intelligence in order to learn from error to predict the target variable, which in this case is the stock price. A standard artificial neural network architecture is shown in Figure 2.1.

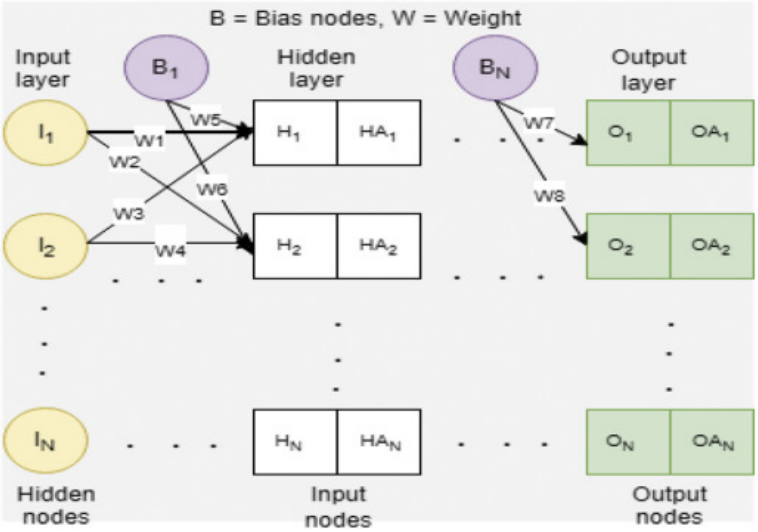

Figure 2.1: A typical neural network architecture
Source: Gupta et al. (2013)

2.3.2 Deep Neural Network

A deep neural network (DNN) is a sub-type of a neural network that has numerous hidden layers of neurons. A DNN ‘s numerous layers allow it to process data at a high level of abstraction. Using a DNN model to forecast stock prices can be difficult since it is inefficient when dealing with long sequences of inputs.

In a comparative study by Shah et al. (2018), long short-term memory (LSTM) was compared with DNN for stock prediction. The LSTM model used Python's Keras module imported from Tensorflow. The model contained four layers: two were hidden, one was an input layer and one was an output layer. An ADAM optimiser was used, which minimises the loss of validation data. Similarly, the DNN model used Keras and also contained the same layer structure as LSTM. The activation used for the hidden layers was ReLU, while sigmoid was used to activate the output layer. When compared to LSTM, the DNN under-performed in stock price prediction on a weekly basis (Shah et al. 2018). With the use of directional accuracy (DA) as a metric, LSTM scored 0.606 while the DNN scored 0.526. Hence, an LSTM model has a better chance of producing accurate predictions.

2.3.3 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of neural network that is meant to train from time-varying and sequential patterns. Since the stock market is non-linear, RNN achieves an optimal solution in retaining past data to provide an accurate prediction (Samarawickrama & Fernando 2017).

Samarawickrama & Fernando (2017) designed an RNN model to predict daily stock prices. The simple recurrent neural network (SRNN) contained 6 input neurons in the input layer and a hidden layer with hidden neurons ranging from 2 to 11. The output layer had included only 1 output neuron. Root mean squared propagation (RMSProp)

is used as a gradient descent where it normalises gradients with the use of the moving average of squared gradients.

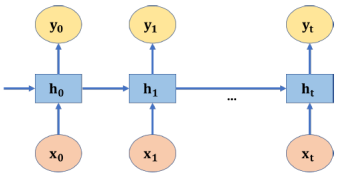

Figure 2.2: A generic recurrent neural network architecture
Ho & Huang (2021)

The output of the RNN depends on the main elements of the sequence. For instance, if a word needs to be predicted in a sentence, the predicted word must have a semantic association in the sentence with the rest of the words. Output is generated on previous and current inputs. Figure 2.2 displays a generic recurrent neural network architecture in which y_t, h_t, x_t represent the output hidden and input layers where t is referred to as time during that state (Ho & Huang 2021).

2.3.4 Long Short-Term Memory

An LSTM model can learn and process historical information. LSTM is an improvement over RNN since it utilizes special units in addition to standard units, which helps to maintain information for a prolonged period of time (Hochreiter & Schmidhuber 1997). Due to the intricacy of the data structure seen in the stock market, this special unit, also known as a memory cell, provides an appropriate solution.

Figure 2.3 displays an LSTM unit with forget (f), output (o) and input (i) gates. The memory cell (c) holds the stock's historical data. The peephole connections from the memory cell allow access to the constant error carousel, preventing vanishing gradients. The LSTM unit has the ability to store previous processing results selectively using the gates in the unit, making it practical for time-series predictions (Wang et al. 2020). It

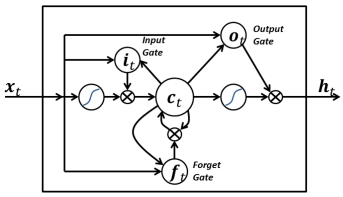

Figure 2.3: A Peephole Long Short-Term Memory Unit
Nelson et al. (2017)

is critical to predict stock prices in a timely and accurate manner in order to ensure national economic stability. However, due to market volatility, stock price prediction has problematic characteristics, as highlighted by Wang et al. (2012).

In a study carried out by Jin et al. (2020) Stocktwits was used as a dataset to perform sentiment analysis on users' tweets regarding a particular stock. The study used a variety of LSTM models as a comparison to predict stock closing prices. Their findings indicate that an LSTM model combined with sentiment scores performs better than a base LSTM model, which produced a MAPE of 4.58, while the LSTM model with sentiment scores produced a MAPE of 2.23.

Nelson et al. (2017) investigated how LSTM networks, in conjunction with technical analysis indicators, might be used to forecast future trends in stock prices based on past data. The open, close, high, low, and volume of a set of stocks in the Brazilian market were gathered and categorized. The model made use of Google's TensorFlow to create an LSTM layer that was utilized to input indicators and pricing data and produced an output using sigmoid activation. The model developed resulted in 0.559 as the highest accuracy score for the BBDC4 index while 0.530 as the lowest accuracy score for the CIEL3 index.

2.3.5 Support Vector Machines

Support vector machines (SVMs) are a collection of supervised learning models which are used for regression and classification. SVMs are efficient in spaces with higher dimensions, which can be optimal for sentiment analysis considering the complexity of the structure of data. In a study carried out by Ahmad et al. (2017), sentiment analysis was performed on two pre-classified datasets gathered from Twitter. They were split into classes depending on sentiment ranging between positive, negative and neutral. Datasets were pre-processed before handling it to the model. This was done by normalising the data for smoother algorithm run times and effective outcomes in less time. This paper concluded that the SVM model has impacted the performance positively on the dataset.

When compared to SVM, LSTM performs more efficiently (Lakshminarayanan & McCrae 2019). During a comparative study by Lakshminarayanan & McCrae (2019) LSTM was compared to SVM for stock price prediction. Their study included multiple models for both machine learning techniques. The results indicated that LSTM performed better than SVM when comparing the mean absolute percentage error (MAPE). The SVM-based model led to a MAPE of 2.47, while the base LSTM model showed a score of 1.41. This proves that LSTM is an ideal approach when using complex datasets such as the stock market.

2.4 Natural Language Processing

Natural language processing (NLP) is an approach in the area of computer science more precisely in the area of artificial intelligence, where computers are given the ability to interpret spoken and textual words, similar to the way humans do (Liddy 2001). Sentiment analysis is a sub-type of NLP and it is the study of human emotions, opinions, and attitudes towards something. Sentiment analysis is considered to be a classification process

where it can be classified into three main levels: document, sentence and aspect level (Medhat et al. 2014). The sentence level's goal is to categorize sentiment expressed in every sentence while a document-level sentiment analysis categorises sentiment on the whole data usually expressed as a positive or negative sentiment. The aspect level aims to categorise sentiment with reference to specific parts of the data.

2.4.1 Text Pre-Processing

Text pre-processing involves preparing data for classification and text cleaning. Generally, online tweets contain a large amount of insignificant data. The text extracted from tweets usually contains advertisements, spam, scripts and HTML tags. Choosing to keep these unwanted words increases the problem dimensionality, thus classification will be harder (Haddi et al. 2013). In NLP every word is considered a dimension, thus it is important to keep only valuable text. Reducing text noise helps to improve the classification speed process and also the classifier's performance, which will consequentially result in a better sentiment analysis. Text pre-processing contains multiple steps such as white-space removal, online text cleaning, abbreviation expansion, removal of stop words and handling of negation words. These steps are also known as transformations. In a study by Haddi et al. (2013), data was transformed to be used for sentiment analysis of online movie reviews. Expression techniques and pattern recognition were used for abbreviation expansion, while for stop words a pre-constructed list was used. Negation words were tagged, but results showed no significant difference in results when removing negation words. This research concluded that pre-processing text greatly affects the performance of the classifier.

2.5 Sentiment Analysis

Sentiment analysis is a field of study that systematically uses computational linguistics, natural language processing, and text analytics to extract and study subjective information and affective states. Chen & Lazer (2013) indicated that in sentiment analysis there are six mood states which are important in the prediction of sentiment. In their study, a list of pre-generated words was used to associate sentiments such as "sad" or "happy" with the given dataset. The model was built for day-to-day movement prediction in the stock market using Twitter as a dataset predating the market by three days. This study concluded that a correlation exists between Twitter and stock market movement when the dataset is predated three days before.

2.5.1 Behavioural Economics

Behavioural economics is the economic decision-making process that institutions and individuals make, and which is related to the study of psychology. Making an optimal decision that provides the greatest satisfaction and benefit is difficult, thus it affects the stock market, making it more unpredictable. The rational choice theory states that persons use calculations on choices linked with their own individual objectives. This theory presumes that, depending on constraints and preferences, people can make rational decisions by evaluating the benefits of the various options available. Blasco et al. (2012) state that herding, in finance, is a terminology which states that investors choose to follow and imitate the decisions of other knowledgeable investors In their study herding behaviour was explored among investors to prove its emotional and rational factors. Causality tests were applied to evaluate the impact of market sentiment depending on herding intensity. Data from the Spanish Stock Exchange Association was collected, specifically the Ibex-35 index which is the main stock exchange of Spain. The daily trading volume is also gathered from the Ibex-35, an increase in the average volume can indicate herding

is being applied. Herding intensity is calculated on selling and buying sequences. This research concluded that there is an emotional correlation with herding, and this can help when responding to large market fluctuations.

2.5.2 Pre-trained Sentiment Analysis Tools

Pre-trained models are a type of model already trained previously on a sizeable dataset related to the field of study. There are several pre-trained tools for sentiment analysis, such as VADER and TextBlob. TextBlob is a python library used for handling textual data to perform sentiment analysis. Similarly to other tools, TextBlob produces a result within the range of -1 to 1. The limitation of TextBlob is the lack of advanced metric scores, which can be an issue for academic studies.

Valence Aware Dictionary and sEntiment Reasoner (VADER) is a rule-based, lexicon tool used for sentiment analysis. This tool is mainly used to define sentiments expressed in social media posts and can classify data in polarities being positive or negative. An advantage of VADER is that it provides a value for a sentiment's positive or negative polarity. In a study by Elbagir & Yang (2019) VADER was applied as a sentiment analyser on Twitter data. The study used scores which depended on the polarity of the sentiment to determine if the tweet was positive, neutral or negative. The scores were set at -2 to +2 where the smaller the number, the more negative the text is represented. Tweets with a score of 0 where assigned a neutral tag. The study concluded that VADER can be used to analyse sentiment in tweets, since the classification provided great results.

2.5.3 Performance Metrics

In a study by Ho & Huang (2021), sentiment analysis was used for stock price movement prediction with the use of LSTM. The Twitter dataset was processed into the VADER tool, and the polarity results were classified into five sentiments: more positive, positive,

neutral, negative and more negative. Sentiment score was calculated on one tweet and the overall score was calculated for the given day. A higher sentiment score meant more positive stock values for that particular day. The sentiment score was integrated with historical trading data to output a stock price movement prediction.

To assess the model's performance, various statistical metrics where used. The results were placed in a table, also known as the confusion matrix, which contained two dimensions: the predicted and actual data. This table included a summary of results often used for classification problems. The confusion matrix had the following key values:

- True Positive (TP) – both prediction and actual data are true
- True Negative (TN) – both prediction and actual data are false
- False Positive (FP) - model predicts true but actual result is false
- False Negatives (FN) – model predicts false but actual result is true

With these key values, statistical metrics such as accuracy, recall, precision and F-score can be extracted. These metrics determine the model's efficiency and can be used to compare models. Accuracy was calculated by dividing the number of correct predictions by the total number of predictions with the formula below:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision is a metric that shows how much is the model precise. It is a ratio of true positives against total values where the model predicted true. The formula can be calculated as shown below:

$$Precision = \frac{TP}{TP + FP}$$

Recall shows the sensitivity of the model. It calculates the ratio of all true positives against actual positives. It can be calculated as follows:

$$Recall = \frac{TP}{TP + FN}$$

The F-score is the average of recall and precision and can be calculated as shown below:

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

2.6 Conclusion

A summary of machine learning methods associated with sentiment analysis and stock price prediction was presented in this chapter. From the research reviewed, stock price prediction is difficult to implement since the market prices fluctuate constantly. The optimal technique researched is a hybrid approach of historical stock information, financial news and social media. Further research showed that LSTM provided better performance when compared to other machine learning techniques such as SVM and DNN. The next chapter focuses on the research methodology and the implementation of the model used for stock movement prediction.

Chapter 3

Research Methodology

3.1 Introduction

This chapter details the methodology and implementation of machine learning techniques for stock prediction using sentiment analysis. The creation of the models, dataset gathering and generation of results are discussed in this chapter, which is divided into two parts. In the first part, the proposed method for stock prediction is presented. The second part features the proposed prototype's implementation and the experiment's generation.

3.2 Proposed Method

This study focuses on machine learning techniques for stock prediction using sentiment analysis. As outlined in Chapter 2.2.1, stock predictions have several complex characteristics. Since external factors have an effect on stock movement, several techniques are combined in the creation of the models. As described in previous chapters, behavioural economics have an impact on price fluctuations thus, the use of social media and news headlines are included in this research. Stock prediction studies used features from his-

torical data such as: open, high, low, volume and close (Nelson et al. 2017, Chen et al. 2015). This study will use the same features but with the use of adjusted close instead of close since it considers factors such as dividends in addition to the closing price.

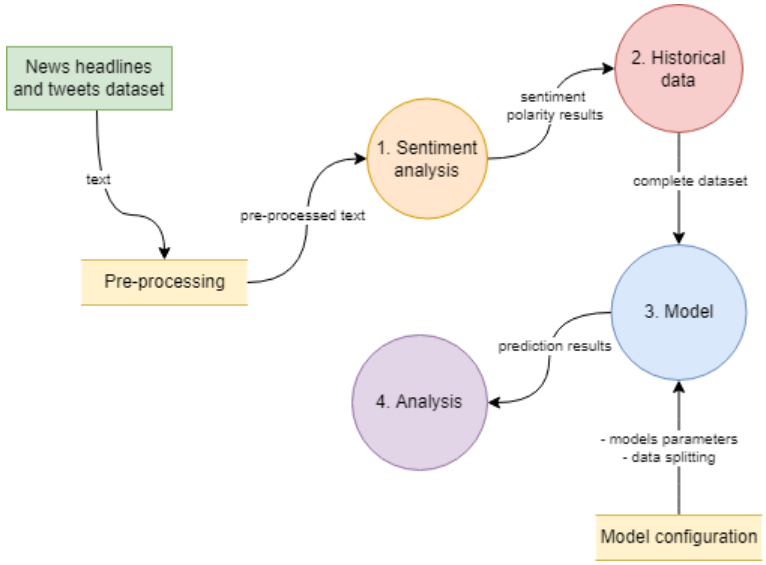

Figure 3.1: Data Flow Diagram: Overview

As seen in Figure 3.1, a general prototype overview is illustrated. The first part focuses on data gathering and producing a sentiment polarity score on the gathered data. From the research compiled in Chapter 2.5.2, VADER will be used as a sentiment analysis tool since it provides a better option than other tools for social media data. The sentiment polarity scores are classified into sentiment scores by assigning a value between 0 to 4 depending on the sentiment polarity score, as inspired by Ho & Huang (2021). As outlined in a comparative study by Lakshminarayanan & McCrae (2019), there is a lack of studies evaluating the performance of LSTM and SVM models. Therefore as an experiment, LSTM will be compared to a linear version of SVM with the addition of sentiment data. The rest of the chapter will focus on the implementation of the methodology, as illustrated in Figure 3.1.

3.3 Sentiment Analysis

The process in this phase focused on two datasets from Twitter and financial news websites. The top four companies in the Standard and Poor's 500 (S&P 500) stock market index were chosen. The stock tickers chosen were Alphabet Inc. Class A (GOOGL), Microsoft (MSFT), Meta Platforms Inc. (FB) and Apple Inc. (AAPL). The most popular companies were chosen in order to provide an adequate number of tweets and financial headlines, which were used to conduct sentiment analysis.

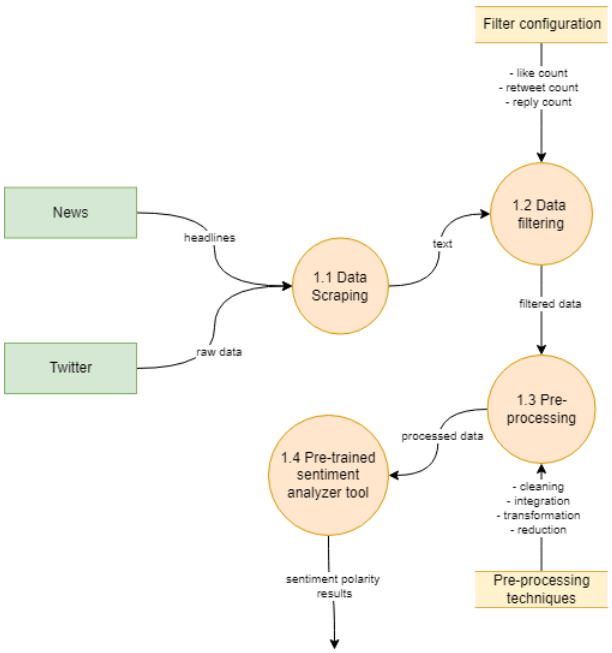

Figure 3.2: Data Flow Diagram: Phase 1 - Sentiment Analysis

3.3.1 Twitter Data

For the first dataset, numerous tweets from Twitter were required, thus the official Twitter API was considered an option. After several attempts to get tweets, Twitter API proved to have limiting factors that could have affected the research. The official Twitter API has request limits, rate limitations and gathering of tweets of up to 7-days only, which is inadequate for this research. An alternative solution was found in a Python project

named Twitter Intelligence Tool (TWINT). TWINT provided several benefits, such as no limitations and access to historical Twitter data. Tweets were scraped starting from the beginning of 2018 to the end of 2021. During this process, it was noted that several scraped tweets contained spam, and some were in a language other than English. Therefore, criteria such as the minimum amount of likes, minimum number of replies, and minimum number of retweets were set to reduce spam tweets. A query was created such that a tweet must contain a minimum number of replies, retweets or likes. The minimum number was set to 2 for any of the three categories. TWINT also provides a query search function that can provide tweets in English only. The data gathered was stored in a CSV file as displayed in Figure 3.3. Datasets were split into files based on their ticker name.

created_at	username	tweet	language	replies_count	retweets_count	likes_count
2020-02-29	tomvideo2	#Azure Exam AZ-500! Three courses have be	en	1	3	10
2020-02-29	iamkirsten	A5 Free collaborative #EdTech help empow	en	0	4	7
2020-02-29	netcorecor	ðŸ"CTRACK 1ðŸ"c @fernandoescolar nos h	en	0	6	4
2020-02-29	iprashanth	Great presentation by @automationnext or	en	0	1	4
2020-02-29	erikaehrli1	Our team charter is growing and I am hiring	en	5	18	59
2020-02-28	rihardjarc	There are more and more #startups with #A	en	0	1	4
2020-02-28	advaiyasoli	Meet your organizational requirements aro	en	0	2	2
2020-02-28	schestowit	#microsoft is running #github at a loss just t	en	1	4	6
2020-02-28	minhazav	Got an automated security pull request rela	en	0	1	3
2020-02-28	kenhorens	Another Venture Out Founders Forum in thr	en	1	3	20
2020-02-27	chainlinkju	so many $link rumours being thrown aroun	en	16	28	180
2020-02-27	sgsliv	XR, MR, VR, AR.... So many realities. Looking	en	0	1	3
2020-02-27	lootsaga	Microsoft and Unity pulling out of GDC due	en	0	2	5
2020-02-27	basevision	One week from today @mirkocolemberg wi	en	0	3	5
2020-02-27	eitbiz	#Nodejs #MobileBusinessApps Global comp	en	0	1	2
2020-02-26	langmanbe	Just four companies â€",#Microsoft, #Apple	en	0	2	2
2020-02-26	mikemclea	Thanks to everyone who joined @meetdux	en	1	2	6
2020-02-26	mrtopstep	#Microsoft just pre announced	en	0	1	2
2020-02-26	fabianwilli	A4: IDK, refer to my #CollabTalk A2 really, I	en	0	1	2
2020-02-26	stautistic	Story time: My friend Paul was having troub	en	1	1	6
2020-02-26	chainlinkju	Pending $link anouncements ðŸ'¥ crypto: $	en	1	10	64
2020-02-26	apswin	'Design Ideas' feature in Powerpoint is prob	en	0	1	3
2020-02-26	thedevbc	I'm attending the @kcdnug tonight and	en	0	2	3
2020-02-25	advaiyasoli	Microsoft Dynamics 365 is the next generati	en	0	2	3
2020-02-25	visualive3d	We're here at @industrial_VRAR in Houstor	en	0	1	3
2020-02-25	parvumpri	#Microsoft is showing the power of #XboxSe	en	3	1	4
2020-02-25	flyntrok	'The change in #culture has been the key fo	en	1	1	2
2020-02-25	pragatioga	With 17 #developers #udaan became the fa	en	0	1	3
2020-02-25	pragatioga	Listening about #opensource by Adrian Lee	en	0	1	2
2020-02-25	azurecris	On my way to Braunschweig. #Microsoft #A	en	0	2	3
2020-02-24	nalimr	ðŸ'½åœ"ï¸#datatalk #AI #ML #BigData #Dat	en	0	13	2
2020-02-24	quareds	Great lineup of exhibitors and interesting ta	en	0	1	2
2020-02-24	angelsinth	So it begins... #Coronavirus fears are startin	en	0	1	2

Figure 3.3: Microsoft sample data from Twitter using TWINT

3.3.2 Financial Headlines Data

The second dataset generated was financial headlines scraped from various financial news websites. This data will help to balance out social media misinformation. A Python library named finnhub [1] was used to scrape financial headlines relevant to the tickers

[1]Finnhub official documentation: https://finnhub.io/docs/api/introduction

used for the Twitter dataset. After further research, a limitation of financial headlines of up to one year was revealed, which does not coincide with the range of dates used in the Twitter dataset; thus, a custom scraper had to be developed. The scraper uses a Selenium[2] library to interact with the dynamic parts of the website, such as scrolling and button clicks.

In contrast, another package named BeautifulSoup[3] was used to extract the page source code and the required information. Scraping was done on three financial news websites that contained enough data to balance with the Twitter dataset. Bloomberg, Forbes, and Financial times were used since they are reputable across the financial industry and have published thousands of articles over several years. Since their respective websites differ, the scraper had to be split into two versions.

For Bloomberg scraping, a pre-defined URL with a search query of a specified ticker was loaded into selenium's web driver, which uses Google Chrome's driver executable. The current page source contained limited data thus, with the use of selenium's web driver, a button that loaded more results was located using XPath and regular expression (regex). This button was clicked until displayed news reached the start of 2018, and then the page source was stored in a variable. After several attempts, Bloomberg identified the scraper as a bot and blocked access to the website. A workaround for this issue was to use a Python library named undetected chrome driver, which worked the same as the selenium web driver but did not trigger CAPTCHA security measures. In addition to this library, after every button click, the program was instructed to wait for several seconds to mimic human behaviour and prevent bot detection. Once data has been successfully collected from the page source, the date and financial headline were stored in a dataframe. With the use of the BeautfulSoup library, elements were searched using a lambda expression to retrieve the headline and publish date. The data frame was saved as a CSV

[2]Selenium library: https://pypi.org/project/selenium/

[3]BeautifulSoup documentation: https://beautiful-soup-4.readthedocs.io/en/latest/

file named with the ticker used and the source from which data was retrieved. Forbes contained a similar layout to that of Bloomberg; thus, the same scraper model was used. The changes in the Forbes version were the paths to the button and data elements. An Excel file containing all the recovered data was saved as seen in Figure 3.4.

date	text
5-Nov-21	Apple Stock Holds Up Despite Supply Issues. Is It A Buy?
19-Dec-21	Stocks This Week: Buy Apple And Sell Short Monster Beverage
6-Dec-21	Apple Stock Hits All Time Highs. Should You Buy, Sell Or Hold?
9-Nov-21	Apple Supplier Stocks Are Underperforming Due To Chip Shortage. Should You B
29-Dec-21	What Does 2022 Have In Store For Apple Stock?
29-Oct-21	Microsoft Is Now The Worldâ€™s Most Valuable Company After Apple Falls On E
16-Nov-21	Stocks That Move The Stock Market
28-Dec-21	Buy Match To Exploit Appleâ€™s Outdated Payment System
10-Oct-21	Stocks This Week: Buy Costco, Apple, And NettApp
8-Oct-21	Apple Stock Looks Like A Buy At $142
26-Oct-21	What To Expect From Appleâ€™s Q4 Earnings?
12-Oct-21	Tesla Vs. Apple: Which Megacap Stock Should You Pick?
27-Oct-21	Should You Pick This At-Home Fitness Stock Over Apple?

Figure 3.4: Apple financial headlines from Forbes sample

Scraping data from Financial Times required several alterations since the website contained several limitations. The first limitation encountered was that Financial Times did not support queries that contained more than 1000 results thus, scraping had to be done monthly. Another issue encountered was that some headlines contained no dates and thus were invalid.

date	text
27-Jan-21	Techâ€™s roaring 2020
27-Jan-21	EU spat with AstraZeneca throws spotlight on vaccine supplies
27-Jan-21	Big Tech is trying to take governmentsâ€™ policy role
27-Jan-21	FirstFT: Todayâ€™s top stories
27-Jan-21	Aim-listed Scapa bought by US peer for Â£400m
27-Jan-21	Microsoft shares hit record on work-from-home revenue boost
26-Jan-21	FirstFT: Todayâ€™s top stories
26-Jan-21	North Korea hackers use social media to target security researchers
26-Jan-21	â€˜Deepfakeâ€™ videos: to believe or not believe?
25-Jan-21	Global chip shortage puts car supply chain under the microscope
25-Jan-21	US stocks rise as tech earnings loom
25-Jan-21	Streaming rivalry gets animated
25-Jan-21	US stock rally drives â€˜ludicrous indexâ€™ towards dotcom era heights

Figure 3.5: Microsoft financial headlines from Financial Times sample

The critical difference between Bloomberg and Financial Times is that on Bloomberg, the page source contains all the financial headlines and publish dates from the range of

date provided, while Financial Times only contains 25 news articles per page source. This issue was solved by scraping every page and appending the data to a dataframe until the beginning of 2018. After scraping is finished, the data was stored in a separate CSV file containing the article name and date.

3.3.3 Twitter and Financial Headlines Pre-processing

Once data has been collected, the next step was to process and clean data. Throughout data gathering for tweets and financial headlines, several problems with data were noticed. Some of the data contained symbols and invalid characters, which had to be removed. Another issue was the date formatting, where Bloomberg and Financial Times data contained differing data formats.

As seen in Figure 3.2, several pre-processing techniques were applied to the dataset before passing the data to the pre-trained sentiment analyser tool. Twitter and financial headlines were loaded separately to check for invalid data. Several headlines without a date were discovered in the Financial Times CSV file. Data without a date were removed since the text needs to be allocated to a particular day. The format of news dates was changed to match the format of the Twitter dataset. Non-essential columns such as username, language, reply count, retweet count and like count from the Twitter dataset were removed, only the date and text were kept. This dataset will be merged with the historical dataset; thus the same date format is essential to avoid misinformation or loss of data. The three CSV files containing tweets and news headlines from Twitter, Bloomberg, and Financial Times were merged into one single file and saved as a CSV for further reference.

The next step was text processing, where several NLP techniques are used to prepare the dataset for sentiment analysis. Unlike other sentiment analysis tools, VADER uses emojis to classify the polarity of text. In this research, emojis will be removed since

most of the financial emojis, such as rocket and moon emojis, are not recognised by the VADER tool. Symbols and flags are also removed from text since they do not provide polarity change from the sentiment analyser tool. Stop words are a set of words that are often removed in sentiment analysis studies (Haddi et al. 2013, Ho & Huang 2021, Batra & Daudpota 2018). A Python library named Natural Language Toolkit [4] (NLTK) was used to remove stop words from the text. This technique did not work as expected, since stop words containing an apostrophe were not removed correctly. The workaround for this issue was to use contractions, where this library expands words in the text by removing the apostrophe and displaying the complete word. As depicted in Figure 3, the text gathered from the scraper contained non-ASCII characters. These characters are removed with the use of lambda and regular expressions.

Further evaluation of the data showed that some text contained hyperlinks that needed to be removed since they would not affect the sentiment result. With regex and string manipulation, links and punctuation were removed from the text. Results were saved in an Excel file containing the date, plus old, and new text with pre-processing techniques (see Figure 3.6).

date	text	new_text
10/10/2021	MacBook Pro 16â€ laptops now pushed back to a 6-8 week shipping wait timeâ€¦ Also still unavailable to pick up in any store in the UK Come on #Apple .. release these damn M1X machines already â€¦ do not you realise some of us have deep seeded addictions to your products?!!	macbook pro 16 laptops pushed back 6 8 week shipping wait time also still unavailable pick store uk come #apple .. release damn m1x machines already realise us deep seeded addictions products !!
2/4/2020	Remember playing ball with WIRED headphones? Could not be me. #sports #airpods #apple #backstreetball #recess	remember playing ball wired headphones could me. #sports #airpods #apple #backstreetball #recess
6/8/2019	Fucking Apple products. #Apple #BMovieManiacs	fucking apple products. #apple #bmoviemaniacs
2/4/2020	Investors Look To Extend Stock Rally On Optimism About Potential Chinese Stimulus	investors look extend stock rally optimism potential chinese stimulus

Figure 3.6: Random sample from processed data

3.3.4 Pre-Trained Sentiment Analyser Tool

The pre-processed data is loaded into a dataframe and divided into four parts, with each part containing information relevant to one year. This process is done to improve the

[4]NLTK documentation: https://www.nltk.org

runtime of the sentiment analyser tool since the appending function of a dataframe takes longer when iterating through a large number of rows.

$$\text{Sentiment} = \begin{cases} \text{most positive} & \text{if compound } > 0.6 \\ \text{positive} & \text{if compound } \in [0.5, 0.1) \\ \text{neutral} & \text{if compound } \in [0.1, -0.1) \\ \text{negative} & \text{if compound } \in (-0.1, -0.6] \\ \text{most negative} & \text{if compound } < -0.6 \end{cases} \tag{3.1}$$

The following process was to execute the VADER tool for the four parts of the dataset. This tool returns four scores: positive, neutral, negative and compound. The positive, neutral, and negative values range from 0 to 1, where a higher value means better categorising that sentiment. Contrarily, the compound score contains a value from -1 to 1, where this value is the normalised aggregated total of the other combined scores.

$$\text{score} = \begin{cases} 4 & \text{if tweet is most positive} \\ 3 & \text{if tweet is positive} \\ 2 & \text{if tweet is neutral} \\ 1 & \text{if tweet is negative} \\ 0 & \text{if tweet is most negative} \end{cases} \tag{3.2}$$

The score is assigned depending on the sentiment as shown in equation 3.1, and it is set to a single tweet. The four parts are saved in an Excel file containing text, date and score (see Figure 3.7).

date	text	new_text	positive	neutral	negative	compound	score
12/31/2018	So that chain message in #facebook that s	chain message #facebook say	0	0.833	0.167	-0.1531	1
12/31/2018	I will Speak LIVE At 5p..CST.. TODAY... FACE	speak live 5p..cst.. today... fa	0.136	0.864	0	0.296	3
12/31/2018	It is almost 2019 and people still think cutt	almost 2019 people still think	0.112	0.683	0.205	-0.2263	1
12/31/2018	Just realized that next year those festive p	realized next year festive par	0.416	0.584	0	0.6908	4
12/31/2018	$FUSZ Happy Healthy New Year @RoryCut	$fusz happy healthy new yea	0.241	0.759	0	0.8356	4
12/31/2018	Things I am not doing in 2019: #Facebook,	things 2019 #facebook suffer	0	0.609	0.391	-0.8402	0
12/31/2018	Bet #Facebook is an absolute delight today	bet #facebook absolute delig	0.328	0.672	0	0.5994	3
12/31/2018	I have taken #Facebook off my phone and	taken #facebook phone use	0	0.769	0.231	-0.4404	1
12/31/2018	Mission accomplished. Good bye #Faceboo	mission accomplished. good	0.222	0.575	0.203	0.128	3

Figure 3.7: Facebook sentiment score sample

3.4 Historical Data

The second phase of this research involved retrieving historical stock data and merging sentiment scores. The third data set produced historical stock prices and volume depending on the ticker used. To retrieve this data, a Python library called Yahoo_fin [5] was used. This library scraped data from Yahoo Finance and was used to gather information related to the tickers used. The collected data is allocated to these columns:

- **Open** : stock price on market open
- **High** : highest stock price reached that day
- **Low** : lowest stock price reached that day
- **Volume** : the quantity of shares traded
- **Adjusted close** : stock closing price

3.4.1 Historical Data Transform

The historical dataset was loaded into a dataframe and sliced to get data from 2018 to 2021. Upon displaying data, the date was shown as an index of the dataframe. This index was copied and stored inside a column named ”date”. Another column called

[5] Yahoo Finance Scraper: http://theautomatic.net/yahoo_fin-documentation/

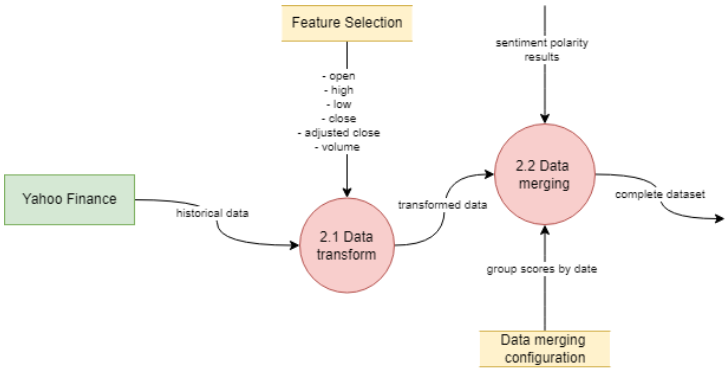

Figure 3.8: Data Flow Diagram: Phase 2 - Historical Data

”future” was created, and it contains the next day’s adjusted closing price where this will be used as a target variable. Additionally, another future column with the price of the next 5 days’ closing price was created to conduct an experiment on how models process future prices. The adjusted close was used instead of the regular close price since the adjusted close considers factors such as stock offerings, dividends, and stock splits thus, it is considered to be a more accurate value representative. With Pandas in-built function shift, this column was appended rows with next day future prices. The future price of this row will be the target variable where the model tries to predict it.

An additional column was appended using equation 3.3, where P is the movement percentage, x_1 is the future price, and x_2 is the adjacent close. This column will contain the future price of that specific day but in percentages. This value indicated the stock price movement and could contain negative and positive values.

$$P = (x_1/x_2 - 1) \times 100 \tag{3.3}$$

3.4.2 Historical Data Merging

As shown in Figure 3.8, the transformed historical data was merged with the polarity scores to finalise the dataset. The sentiment scores produced in Chapter 3.3.4 were

loaded and stored into a dataframe. This dataframe is grouped by date as key, and frequency is set to days. The mean of the grouped data is calculated and appended to the historical dataset using a column named ”senitment_score”. The complete dataset was saved in an Excel file and contained both historical data and sentiment scores (see Figure 3.9).

	open	high	low	close	adjclose	volume	ticker	future	movement_percentage	sentiment_score
4/12/2018	43.3525	43.75	43.26	43.535	41.75425	91557200	AAPL	41.89571	0.338802033	3.181818182
4/16/2021	134.3	134.67	133.28	134.16	133.3673	84922400	AAPL	134.0433	0.506855437	3.038461538
7/31/2020	102.885	106.415	100.825	106.26	105.1034	374336800	AAPL	107.7518	2.519767368	2.745098039
2/7/2019	43.1	43.485	42.585	42.735	41.43338	126966800	AAPL	41.48207	0.117515856	2.620689655
5/8/2019	50.475	51.335	50.4375	50.725	49.39095	105358000	AAPL	48.86029	-1.074404868	2.571428571
10/11/2019	58.2375	59.41	58.0775	59.0525	57.94018	166795600	AAPL	57.85678	-0.143949395	2.20212766
12/7/2018	43.3725	43.6225	42.075	42.1225	40.83955	169126400	AAPL	41.10859	0.658771164	1.842105263
5/18/2020	78.2925	79.125	77.58	78.74	77.88293	135178400	AAPL	77.43291	-0.577815717	1.818181818
9/9/2021	155.49	156.11	153.95	154.07	153.6497	57305700	AAPL	148.5636	-3.310195015	1.714285714

Figure 3.9: Random sample from the complete dataset

3.5 Implementation of Models

For this research, two models will be developed to predict stock prediction. The first model will use an LSTM architecture. In contrast, the second model will use SVM to contribute to the research gap stated by Lakshminarayanan & McCrae (2019), where there is a lack of comparison between LSTM and SVM for stock prediction. As demonstrated in figure 3.10, the workflow of the two models follows three steps.

3.5.1 Data Splitting and Preparation

Initially, a random seed of 56 is set to get the same results after re-running, making this model reproducible by other researchers. The dataset is loaded and stored in a dataframe, and the features and target are initialised. The volume, open, high, low, adjacent close and sentiment scores are used as features while the future price is set as a target. This implies that the model will learn from the features' data and try to predict the target

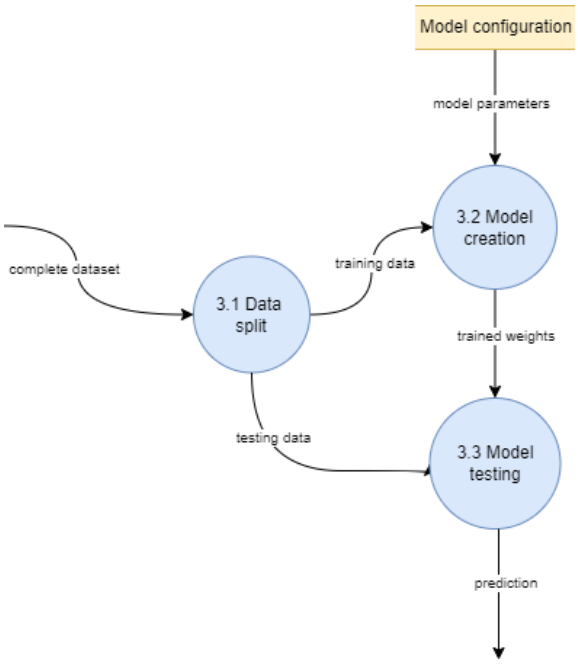

Figure 3.10: Data Flow Diagram: Phase 3 - Model

variable.

Data Splitting and Preparation for LSTM

Data splitting for the two models had to be done separately since the LSTM required a 3-d input shape. The first dimension in the 3-d shape will represent the batch size. The time steps fed into a sequence represent the second dimension, while the unit number in one sequence represents the third dimension. For this research, the first dimension's value is the dataset's length, the second dimension's value is set to 50, and the third dimension is set depending on the number of features. The sequence length outlined in the second dimension is also known as the window size. Multiple values were tested to find the optimal window size, with a value of 50 being the best value. The dataset is split using a sklearn train_test_split [6] python library. Data is split at a percentage of 70% assigned to training and 30% allocated to testing. Additionally, data is scaled using a

[6]Sklearn library: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

min-max scaler which normalises data by setting values between 0 and 1.

Data splitting and preparation for SVM

The SVM model's data splitting and preparation are done similarly with some minor changes. A standard scaler is used for the SVM model, where this is calculated by subtracting the feature vector x with its mean $\bar{x}$ and it is divided by the standard deviation σ. The standard scaler is used to reduce the ranges between large values. The data is split 70% for training and 30% for evaluating the model.

$$x' = \frac{x - \bar{x}}{\sigma} \tag{3.4}$$

3.5.2 Model Creation

For the task of stock prediction using machine learning techniques, two types of models were chosen. Recent studies in stock prediction have demonstrated that LSTM and SVM models are suitable for this task (Lakshminarayanan & McCrae 2019, Ho & Huang 2021). This part aims to use the dataset created in the previous chapters for the training and testing of the models. Although the goal for both models is identical, the implementation is different for both of them. An sklearn library named RandomizedSearchCV[7] was used to find the best parameters for the models. The RandomizedSearchCV library selects random combinations to find the models best fit. This module was preferred over GridSearchCV since RandomizedSearchCV takes less time to execute than GridSearchCV.

[7] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

LSTM Implementation

The LSTM model implemented used features from a Python library named Tensorflow[8] . Specifically, the module used was Keras[9], where layers, callbacks and models are imported for later use. A method was used to build the model with a set of pre-defined parameters and was passed in a Keras Regressor wrapper. This wrapper is used to execute the RandomizedSearchCV to find the best parameters. The following parameters are used:

- **Dropout** : float that ranges between 0 and 1. Drops units for linear change of the inputs
- **Units** : positive number that refers to the dimensionality of cells
- **Batch** : the amount of training instances used in a single iteration
- **Epochs** : number of cycles for the given dataset
- **Activation** : function used in the learning of complex patterns; introduces non-linearity in the neuron's output
- **Optimiser** : modifies attributes such as learning rate and weights.
- **Loss** : invalid decisions penalties

Callbacks were used to display the model's training and validation progress, as seen in Figure 3.11. Additional callbacks such as model checkpoint and early stopping were used as well. Initial tests on the model have shown that the model was underfitting, thus producing unfavourable outcomes. As stated in Chapter 2.3.3, the stock market is non-linear and complex to process, hence additional layers were added to the LSTM model

[8]https://www.tensorflow.org/learn
[9]https://www.tensorflow.org/api_docs/python/tf/keras

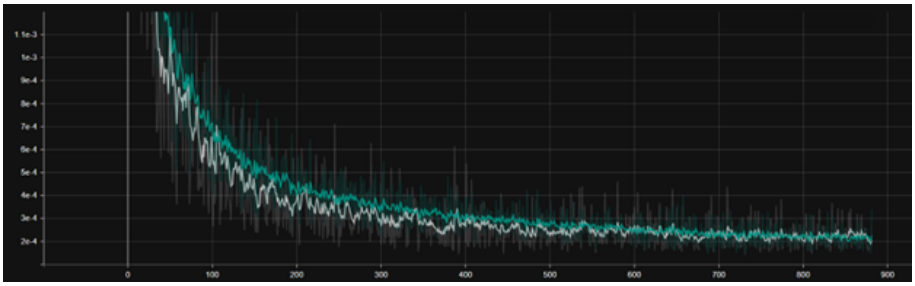

Figure 3.11: Model Training dual-layer LSTM loss

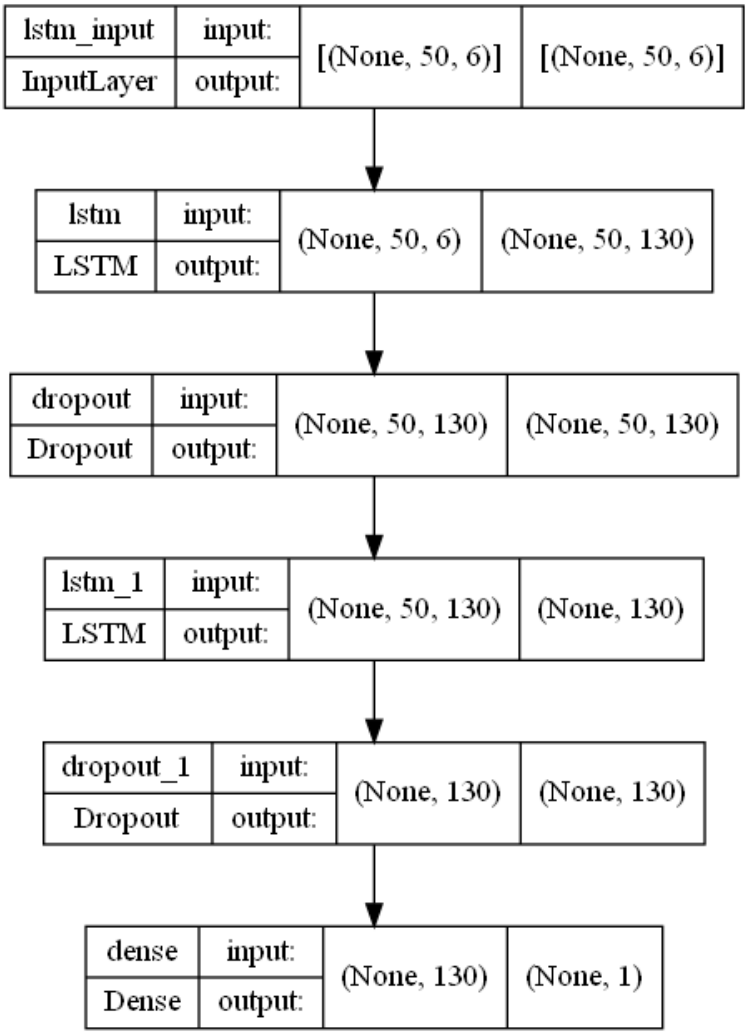

Figure 3.12: LSTM Architecture

to compensate for this problem. When three or more layers were added, the model was overfitting, thus a dual-layer model was the most suitable option. Figure 3.12 shows the final architecture used for the LSTM model. The model was trained with the returned RandomizedSearchCV parameters and are outlined in the next chapter.

SVM Implementation

Contrary to the LSTM implementation, the SVM implementation uses sklearn's python library. The LinearSVR[10] module was used since it scales better with large samples as outlined in the official sklearn's documentation (Pedregosa et al. 2011). Unlike LSTM, this model does not require a 3-d dataset; hence it requires less code. A method containing the initialisation of the LinearSVR model was created. The parameters passed were:

- **C** : regularisation parameter which reduces overfitting
- **epsilon** : tolerance margin in which errors are not penalised
- **loss** : the training goals to be minimised
- **max_iter** : maximum iterations per run
- **tol** : stopping criteria's tolerance

The method was passed to the RandomizedSearchCV with cross-validation set to 5. Cross-validation is set to rotate the tested sample in order to provide a better statistical analysis. Since the hyperparameter tuning process occurs within a brief time frame, the model's weight was not saved. The hyperparameter tuning's best parameters are saved for further reference.

[10]https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html

3.6 Analysis

Once predictions are made, data is inverse transformed utilising the scaler which was used before data splitting. This data is inversely transformed to calculate several metrics using the predicted and actual data. The predicted and actual data are appended to the original dataset. With the predicted data merged with the original dataset, several graphs were created using the matplotlib [11] Python library.

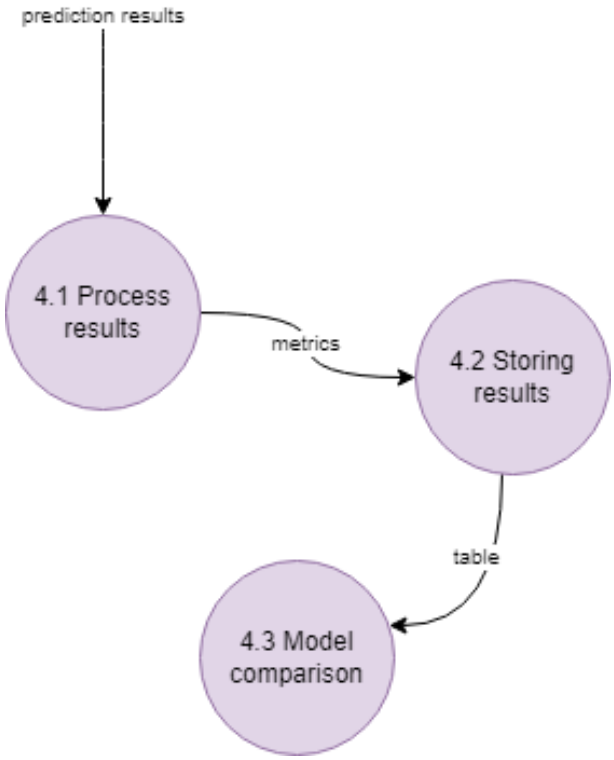

Figure 3.13: Data Flow Diagram: Phase 4 - Analysis

As displayed in Figure 3.13, metrics are stored in a table which will subsequently be used to compare the model results. The table is created using Excel, and it contains the metrics of all the experiments.

[11] https://matplotlib.org/stable/index.html

3.6.1 Investment Strategy

An investment strategy was implemented to test the prediction of daily stock prices. Two separate strategies were implemented in a study by Chen & Lazer (2013). The first strategy involved classification data where only the direction of stock movement was considered. The second strategy consisted of regression data where the invested amount depended on the predicted percentage change. For this study, investment in regression data was opted for since classification data results in data loss. The following investment strategy was implemented centred on observations of the data:

$$\text{investment } = \begin{cases} 75\% \text{ if prediction \% change } \geq 2\% \\ 50\% \text{ if } 2\% > \text{ prediction \% change } \geq 1\% \\ 25\% \text{ if } 1\% > \text{ prediction \% change } \geq 0.5\% \\ 5\% \text{ if } 0.05\% > \text{ prediction \% change } \geq 0.05\% \end{cases} \tag{3.5}$$

In formula 3.5, investment depicts the percentage of funds used for investing. The amount of investment depends on the confidence of the prediction percentage change of the model. If none of the above criteria is met, a sell trigger is initiated, and the result is added to the investment variable. This process is repeated for all of the test data. A graph is outputted containing buy and sell indicators across the test data. An Excel file is saved, with results formatted as invested amount, profit/loss amount, buy indicators and sell indicators.

3.7 Conclusion

This chapter focused on the techniques used to achieve the objectives of this research. The first phase of this study focused on the sentiment analysis of Twitter and financial headlines data. The process is illustrated in a data flow diagram where all the main steps

are described in further detail. The second phase represents the gathering of historical data and the merging with sentiment scores. The third phase focused on how the models were created and prepared for the experiments. Lastly, the fourth phase described how the predictions were processed.

The results of the performed experiments will be displayed and discussed in the next chapter. The next chapter will also include an in-depth evaluation of the metrics used along with the results acquired.

Chapter 4

Analysis of Results and Discussion

4.1 Introduction

This chapter presents the results and evaluation of the methods described in the previous chapter. Additionally, a detailed assessment of the metrics used will be presented in this chapter.

The results and evaluation chapter is split into five sections. The first part describes the metrics used to produce the results in detail. The second, third and fourth parts describe the results of the methodology outlined in the previous chapter in relation to the research questions. Lastly, the final section compares the results acquired to related studies.

4.2 Metrics

Most studies in relation to stock prediction are divided into two categories: regression and classification. Regression models are used to find and predict the actual values, usually using the future price as a target variable. Contrarily, classification is used to measure the upward or downward movement of the stock prices.

$$MDA = \frac{1}{n}\sum_{t} \mathbf{1}_{\text{sign}(A_t - A_{t-1}) == \text{sign}(F_t - A_{t-1})} \tag{4.1}$$

Metrics such as the mean directional accuracy (MDA) are used in regression to measure the prediction direction compared to the actual data. As illustrated in the formula above, the actual data (A_t) at time (t) and predicted data (F_t) at time (t) are checked for equality using the sign function. The sign function returns 1 for positive numbers, 0 for neutral, and -1 for negative values.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| * 100 \tag{4.2}$$

The mean absolute percentage error (MAPE) is calculated to measure the error percentage of the model's results. As described in the above formula, the MAPE is calculated by subtracting the actual data from the predicted data and dividing it by the actual data. The result is multiplied by 100 to get a real percentage required for comparison with other studies.

$$RMSE = \sqrt{\sum_{t=1}^{n} \frac{(F_t - A_t)^2}{N}} \tag{4.3}$$

The root mean square error (RMSE) is the prediction errors' standard deviation. This metric shows how far the results are from the actual regression line. The RMSE is measured by subtracting the predictions from the actual data and then squaring them. The average is calculated by dividing the result by the total number of data (N). Lastly, the RMSE is calculated by determining the square root of the average.

$$R^2 = 1 - \frac{\sum_{t=1}^{n} (A_t - F_t)^2}{\sum_{t=1}^{n} \left(A_t - \bar{A}\right)^2} \tag{4.4}$$

In the above formula, the coefficient of determination, also known as R-squared (R^2),

is used to measure the goodness of fit of the models. The goodness of fit defines the relationship between the observed data and the model. A high R^2 indicates that the model has a strong relationship with the data and thus can fit more effectively with the model. The R^2 is calculated by subtracting the divided sum of square residuals by the total sum of squares by 1. The sum of square residuals is calculated by squaring the deducted amount between the actual and predicted data. The total sum of squares is determined by subtracting the actual data from the mean data ($\bar{A}$).

$$MAE = \sum_{t=1}^{n} \frac{|F_t - A_t|}{N} \tag{4.5}$$

Lastly, the mean absolute error measures the predicted and actual data errors. This metric is calculated by summing the absolute errors and dividing them by the sample size.

4.3 Stock Movement Prediction

The stock prediction techniques are based on the methodology described in Chapter 3.5. For the LSTM and SVM models; hyperparameter tuning was used as outlined in Chapter 3.5.2. The following results of the hyperparameter optimisation were utilised for the LSTM model:

Configuration	**Value**
dropout	0.11662
units	130
batch size	170
epochs	882
activation	linear
optimiser	adam
loss	huber_loss

Table 4.1: LSTM Hyperparameter Results

The same procedure was conducted for the SVM models, and the results are presented in Table 4.2

Configuration	Value
C	10
epsilon	0.00001
loss	squared_epsilon_insensitive
max_iter	2370
tol	0.01

Table 4.2: LinearSVR Hyperparameter Results

These values are used as the best parameters for the models created in the previous methodology. After further evaluation of the models using the above parameters, it was noted that the LSTM model would overfit after a certain number of epochs. Consequently, early stopping was used to prevent overfitting of the model when training. With the models built, the following question is meant to be answered:

Can stock price movement be predicted consistently?

In connection with this question, the subsequent hypothesis is set:

The stock price movement can be predicted consistently with the use of historical data and sentiment scores.

Several experiments were conducted to attempt to answer the above research question. As outlined in Chapter 3.4.1, two types of targets were created. Each model predicted the future price of the next day and five days into the future. The comparison of each model with their future targets was represented into various tables.

In Table 4.3, the MDA for the LSTM model is shown, sorted from highest to lowest accuracy. The results show that the LSTM with sentiment scores for the GOOGL ticker was the best performant model. This can be explained by the low amount of volatility of the ticker GOOGL throughout the years between 2018 and 2021. Therefore, the LSTM

Model	Experiment	Ticker	Steps	MDA
LSTM	Sentiment	GOOGL	1	0.56446
LSTM	Base	GOOGL	1	0.554007
LSTM	Base	AAPL	1	0.54007
LSTM	Sentiment	AAPL	1	0.522648
LSTM	Sentiment	MSFT	1	0.508711
LSTM	Base	FB	1	0.505226
LSTM	Sentiment	FB	1	0.491289
LSTM	Base	MSFT	1	0.491289

Table 4.3: LSTM Results 1-Day Future

was able to predict the movement of the next-day future price more efficiently than the other results.

Model	Experiment	Ticker	Steps	MDA
LSTM	Base	FB	5	0.561404
LSTM	Sentiment	FB	5	0.554386
LSTM	Sentiment	GOOGL	5	0.540351
LSTM	Sentiment	AAPL	5	0.526316
LSTM	Base	GOOGL	5	0.526316
LSTM	Base	AAPL	5	0.519298
LSTM	Base	MSFT	5	0.501754
LSTM	Sentiment	MSFT	5	0.498246

Table 4.4: LSTM Results 5-Day Future

The best performant model for the 5-day future step was the LSTM without sentiment scores for the ticker FB as shown in Table 4.4. It is noted that sentiment scores from Table 4.4 have a more positive outcome than those presented in Table 4.3. This could be due to the fact that sentiment scores affect the stock prices at a later step rather than the next day.

In Table 4.5, the results of the LinearSVR model for the 1-day future target are shown. The best model in this table is the LinearSVR with sentiment scores for the AAPL ticker. When compared to the LSTM model (see Table 4.3), LinearSVR is underperforming. It is also noticed that both models cannot process the sentiment scores efficiently since the change in MDA is minimal.

Model	Experiment	Ticker	Steps	MDA
LinearSVR	Sentiment	AAPL	1	0.523179
LinearSVR	Base	MSFT	1	0.523179
LinearSVR	Sentiment	FB	1	0.519868
LinearSVR	Base	AAPL	1	0.519868
LinearSVR	Sentiment	MSFT	1	0.513245
LinearSVR	Base	FB	1	0.509934
LinearSVR	Base	GOOGL	1	0.480132
LinearSVR	Sentiment	GOOGL	1	0.476821

Table 4.5: LinearSVR Results 1-Day Future

Model	Experiment	Ticker	Steps	MDA
LinearSVR	Base	FB	5	0.566667
LinearSVR	Sentiment	FB	5	0.556667
LinearSVR	Base	AAPL	5	0.546667
LinearSVR	Sentiment	AAPL	5	0.546667
LinearSVR	Sentiment	GOOGL	5	0.5
LinearSVR	Base	GOOGL	5	0.493333
LinearSVR	Sentiment	MSFT	5	0.49
LinearSVR	Base	MSFT	5	0.486667

Table 4.6: LinearSVR Results 5-Day Future

The LinearSVR model for the 5-day future target shows improvement over the 1-day results. The same observation is made where the sentiment score feature is not being processed correctly and thus, the MDA leads to a slight change.

4.4 Investment Strategy

As outlined in Chapter 3.6.1, an investment strategy was conducted to observe each model's performance. Each model will invest a total of $4000 across four tickers. The investment is made on the predictions made on the test data. The data is represented in a table containing the model, total profit, and number of indicators. Every time a buy or sell is executed the number of indicators is incremented by one. These results are based on the LSTM and LinearSVR models with differing features and targets. Additionally,

this sub-chapter intends to answer the following question:

Is sentiment an important factor in stock prediction?

The following hypothesis is set in relation to the above research question:

Tweets and financial news deliver enough information to affect stock prices

Model	Experiment	Steps	Total Profit	Total Indicators	Total Buy	Total Sell
LSTM	Sentiment	1	1348.230023	824	655	169
LSTM	Base	1	1308.672954	840	659	181
LSTM	Sentiment	5	1264.744393	675	596	79
LSTM	Base	5	1253.490935	622	532	90
LinearSVR	Sentiment	5	929.6259552	632	597	35
LinearSVR	Base	5	902.6163811	634	595	39
LinearSVR	Sentiment	1	413.614744	649	475	174
LinearSVR	Base	1	281.056098	674	485	189

Table 4.7: Investment Results with all the combinations

Table 4.7 displays the investment results from the experiments made for every ticker. This experiment aims to answer to answer the question of whether sentiment is an essential factor in stock prediction. The best model from Table 4.7 is the LSTM combined with sentiment scores for the next day's future forecast. This configuration had a 28.8446% profit from the starting amount throughout the test dataset. The LSTM model made a considerable amount of profit when compared to LinearSVR. The sentiment factor has positively impacted the profit throughout all the models. It is also noted that the LSTM with sentiment and without a sentiment for future step 1 had a substantial increase in indicators, thus indicating the model is more confident in investing daily.

In Table 4.8, a more detailed breakdown of the investment profits is displayed. The most profitable tickers were FB and AAPL. This can be explained by analysing the volatility of stock tickers. As illustrated in Figure 4.2, Facebook and Apple had the highest yearly volatility rate for the years ranging between 2018 and 2021. In investment,

Ticker	Steps	Investment Start	Investment End	Profit/Loss	Total Indicators
AAPL	1	1000	1463.080867	463.08087	221
FB	1	1000	1415.042901	415.0429	214
GOOGL	1	1000	1252.651441	252.65144	234
MSFT	1	1000	1217.454814	217.45481	155

Table 4.8: Investment results for LSTM 1-Day Future with Sentiment Scores

higher volatility yields more profit, but it is riskier since it can also lead to more loss. As displayed in Figure 4.1, Google and Microsoft have less volatility than the other tickers.

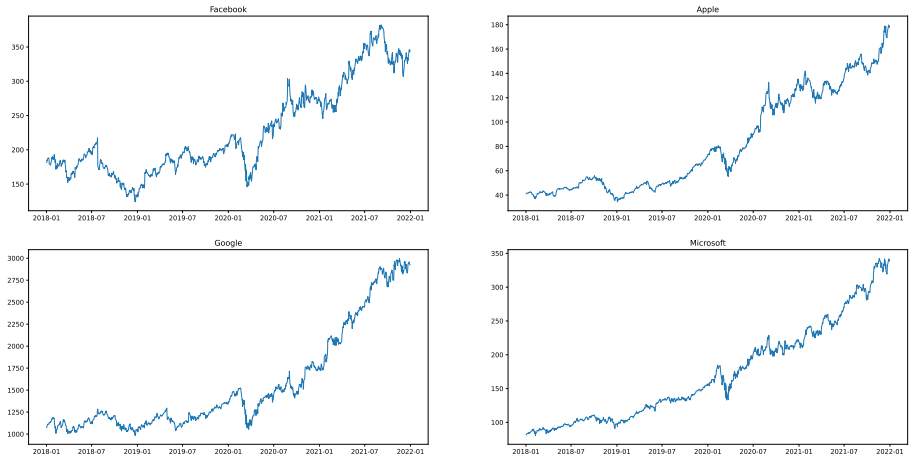

Figure 4.1: Historical stock data from the beginning of 2018 to the end of 2021

Ticker	Steps	Investment Start	Investment End	Profit/Loss	Total Indicators
GOOGL	5	1000	1423.990622	423.99062	116
MSFT	5	1000	1397.696021	397.69602	208
AAPL	5	1000	1299.887241	299.88724	182
FB	5	1000	1143.170509	143.17051	169

Table 4.9: Investment results for LSTM 5-Day Future with Sentiment Scores

The investment results, as depicted in Table 4.9, have different outcomes than those in Table 4.8. The most profitable tickers were GOOGL and MSFT. This could be explained by the increase in percentage change for the 5-day future target. Therefore, the LSTM model was able to generate more profit from stocks with low volatility.

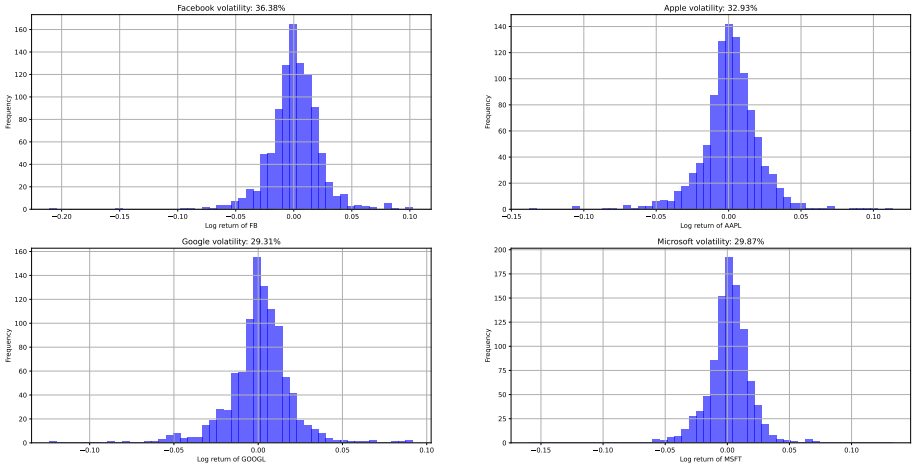

Figure 4.2: Volatility of stocks from the beginning of 2018 to the end of 2021

4.5 Techniques Appropriate for Stock Prediction

As outlined in Chapter 2.3, there are several methods of stock prediction. This section aims to answer the following research question:

Which techniques are appropriate for stock price predictions?

In relation to the research question, the following hypothesis is set:

A machine learning technique can be used to predict stock prices.

Two models were implemented, one using deep learning and the other using support vector machines, as discussed in Chapter 3.5. This aims to find the better technique for stock prediction by comparing both models. The data used in the previous chapter was set at a random seed to ensure that the models train and test on the same data, thus making a valid comparison. The results are presented in a range of metrics, including RMSE, MAE, MAPE, and R2.

From the results gathered in table 4.10, the model with the lowest RMSE is the AAPL ticker without sentiment scores. Although this result had the lowest RMSE, it

Model	Experiment	Ticker	Steps	RMSE	MAPE	MAE	R2
LSTM	Base	AAPL	1	2.917581	1.579043	2.194982	0.963071
LSTM	Sentiment	AAPL	1	3.201025	1.745471	2.407714	0.955547
LSTM	Base	MSFT	1	5.418142	1.597186	4.209763	0.981863
LSTM	Base	FB	1	5.764707	1.437076	4.482922	0.974175
LSTM	Sentiment	FB	1	5.970785	1.507607	4.709045	0.972295
LSTM	Sentiment	MSFT	1	6.421971	1.823619	4.918358	0.97452
LSTM	**Sentiment**	**GOOGL**	**1**	**35.49714**	**1.109474**	**26.08547**	**0.992768**
LSTM	Base	GOOGL	1	35.70565	1.133558	26.60475	0.992683

Table 4.10: LSTM Results 1-Day step

doesn't mean it is the best model, hence the importance of using multiple metrics. The best results obtained were the LSTM combined with sentiment scores for the GOOGL ticker. This result had the lowest percentage of error and the highest R-squared. This can be attributed to the GOOGL ticker being one of the lowest risk stocks, as seen in Figure 4.1.

Model	Experiment	Ticker	Steps	RMSE	MAPE	MAE	R2
LSTM	Base	AAPL	5	5.550039	3.097424	4.298305	0.863033
LSTM	Sentiment	AAPL	5	5.876701	3.26824	4.543261	0.846435
LSTM	Base	MSFT	5	9.884375	2.977966	7.931348	0.93899
LSTM	Sentiment	MSFT	5	10.8493	3.433169	8.772854	0.926497
LSTM	Base	FB	5	12.20874	3.123478	9.780183	0.884255
LSTM	Sentiment	FB	5	12.45317	3.158607	9.904179	0.879574
LSTM	**Base**	**GOOGL**	**5**	**76.44414**	**2.552726**	**60.2237**	**0.966234**
LSTM	Sentiment	GOOGL	5	82.79433	2.777282	65.67375	0.960391

Table 4.11: LSTM Results 5-Day step

When compared to Table 4.11, the error increases as expected. The best result for the 5-day future is the GOOGL without sentiment scores. From the results of Table 4.11, it was noted that sentiment had a negative impact on the price predictions. This observation means that the sentiment score is ineffective in predicting stock prices for long steps. As observed in the above tables, an increase in steps caused a decrease in the R-squared value. This remark can be explained by how well the data fits throughout the model. A greater number of future steps means the model will have issues fitting the

given data.

Model	Experiment	Ticker	Steps	RMSE	MAPE	MAE	R2
LinearSVR	Base	AAPL	1	2.219791	1.279761	1.732809	0.980126
LinearSVR	Sentiment	AAPL	1	2.223839	1.28618	1.7407	0.980053
LinearSVR	**Sentiment**	**MSFT**	**1**	**3.63636**	**1.045664**	**2.732173**	**0.9922**
LinearSVR	Base	MSFT	1	3.642693	1.046692	2.73566	0.992173
LinearSVR	Base	FB	1	6.065802	1.502743	4.649816	0.971615
LinearSVR	Sentiment	FB	1	6.10944	1.509934	4.676777	0.971205
LinearSVR	Sentiment	GOOGL	1	38.22473	1.250317	29.06219	0.992428
LinearSVR	Base	GOOGL	1	38.3791	1.255168	29.22851	0.992367

Table 4.12: LinearSVR results 1-day step

Model	Experiment	Ticker	Steps	RMSE	MAPE	MAE	R2
LinearSVR	Base	AAPL	5	4.916651	2.892157	3.90588	0.900423
LinearSVR	Sentiment	AAPL	5	4.947107	2.904094	3.921013	0.899185
LinearSVR	**Sentiment**	**MSFT**	**5**	**7.572446**	**2.281114**	**5.962741**	**0.965877**
LinearSVR	Base	MSFT	5	7.586153	2.282252	5.966532	0.965753
LinearSVR	Base	FB	5	12.63379	3.221035	10.03406	0.876852
LinearSVR	Sentiment	FB	5	12.64974	3.227465	10.0554	0.876541
LinearSVR	Sentiment	GOOGL	5	92.68951	3.07985	72.65496	0.955049
LinearSVR	Base	GOOGL	5	92.74598	3.082054	72.72324	0.954994

Table 4.13: LinearSVR results 5-day step

The results in Table 4.12 indicate that the LinearSVR was able to predict MSFT with the lowest error. The RMSE values indicated that between the sentiment and base experiment, the LinearSVR was not able to process sentiment scores efficiently. Compared to the LSTM model, LinearSVR performed similarly for the 1-day future target. Similarly to the previous results, the LinearSVR performed worse for the 5-day future step. The best result for Table 4.13 was the MSFT ticker with sentiment scores. The RMSE and MAE values observed in Table 4.13 indicate that the sentiment scores have more effect than those in Table 4.12.

As illustrated in figure 4.3, a target correlation heatmap is displayed to analyse how the sentiment score affects the target variable. The sentiment score affects the future price negatively for every ticker examined. This could be justified by the data gathered. Since some of the data for sentiment analysis is collected during a pandemic, it contains a

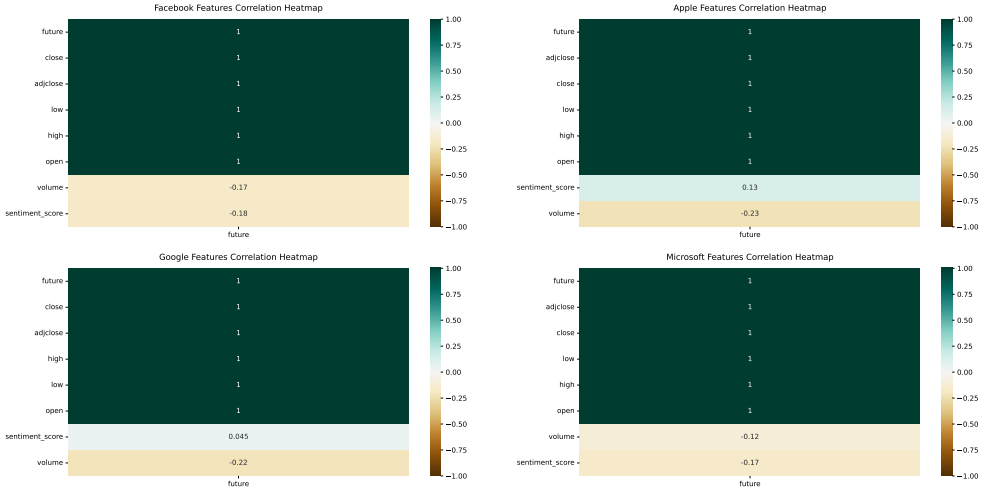

Figure 4.3: Target correlation heatmap

more significant amount of negative sentiment. Therefore, this could explain the negative correlation to the target variable.

4.6 Evaluation in contrast to other studies

In a study by Shah et al. (2018), LSTM was used to predict the Indian stock market. As outlined in Chapter 2.3.2, the LSTM model proposed by the researcher contained one input layer, two hidden layers and one output layer, which is the same layout as described in Figure 3.12. The main differences are the number of hidden nodes, epochs and window length. The result of the study by Shah et al. (2018) recorded an MDA of 0.606 for the LSTM model's weekly predictions. A graph was been plotted to compare the studies, shown in Figure 4.4, which displays the correlation between the S&P BSE SENSEX index and the tickers used. GOOGL and AAPL are the highest correlated tickers with a value of 0.21.

The MDA of the 5-day future, as outlined in Table 4.4 for the tickers of AAPL and

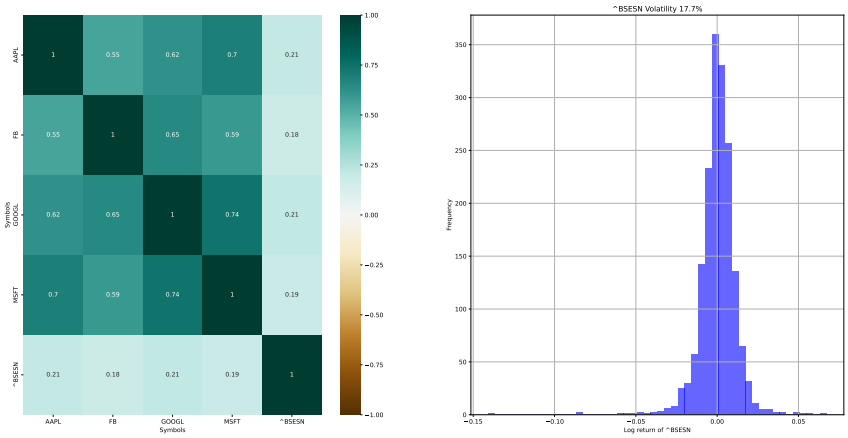

Figure 4.4: S&P BSE Sensex Correlation and Volatility graph

GOOGL, are 0.53 and 0.54, respectively. The change in MDA between the research made by Shah et al. (2018) and the proposed LSTM model could be attributed to the fact that the LSTM model performs better with stocks that have low volatility percentages. As illustrated in figure 4.4, the S&P BSE SENSEX index's volatility percentage is outputted. The S&P BSE SENSEX volatility is 17.7% compared to the 29.31% volatility of the GOOGL ticker, as seen in figure 4.2.

As described in Chapter 3.3.1, Twitter is one of the data gathering websites used for sentiment analysis. In a study by Jin et al. (2020), the LSTM model is implemented along with sentiment analysis gathered from StockTwits. Their approach compares various LSTM model configurations for stock price prediction. The results of their research reveals a MAPE of 2.23 for the LSTM with sentiment scores and a MAPE of 4.58 without sentiment scores. The study attempts a daily stock price prediction with the LSTM models on the ticker AAPL. Compared to this study's results, a MAPE of 1.75 is outputted for the AAPL ticker with sentiment scores while a MAPE of 1.58 for the same ticker without sentiment scores. A substantial improvement is seen in the MAPE scores. It is also observed that in the study by Jin et al. (2020), the sentiment scores had a more

significant and positive impact on the result than in this study. Their research was conducted using the dates 2013 to 2018, while the results presented in Table 4.10 were gathered on data from the years 2018 to 2021. This means that several tweets accumulated were during a pandemic; thus, more negative tweets circulated. StockTwits could also help reduce sentiment data that is out of scope and therefore affect the data negatively since it is specifically used for the stock market.

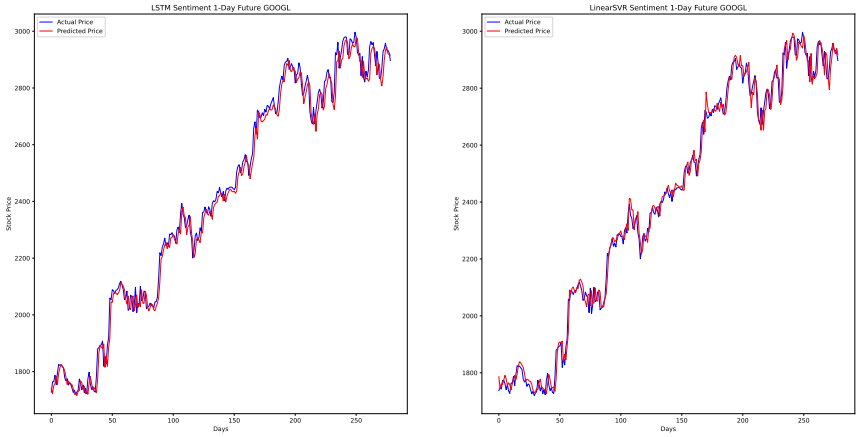

Figure 4.5: LSTM comparison with LinearSVR for GOOGL prediction

In a comparative study by Lakshminarayanan & McCrae (2019), an LSTM and an SVM were compared for stock prediction. Their study used the Dow Jones Index (DJI) as a dataset. MAPE was used as a comparison since the RMSE for the DJI is not comparable with the current stock tickers used. Their SVR model achieved a MAPE of 2.47, while the LSTM model yielded a MAPE of 1.41. The results attained in the current study are a MAPE of 1.11 for the LSTM model with sentiment scores and a MAPE of 1.25 for the LinearSVR model with sentiment scores. These results are based on the GOOGL ticker. For both studies, the LSTM model yielded better MAPE scores than the SVM model. In the study by Lakshminarayanan & McCrae (2019), the difference in MAPE between the models is greater than the difference in the current study. This can

be explained by the SVM models used in both studies. The LinearSVR model used in this study is implemented using a liblinear library instead of libsvm, which is the default library of the SVR. The liblinear library provides better flexibility and choice of loss functions and penalties, thus scaling better to substantial datasets. As seen in Figure 4.5, this graph illustrates the difference in predictions between the results presented in the tables mentioned previously. Subtle differences can be noted from the chart, such as the sudden spike in the forecast for the LinearSVR model between days 150 and 200. An instance where the LinearSVR performed better than the LSTM model was the prediction for the MSFT ticker for the 1-day future. The results obtained are a MAPE of 1.60 for the LSTM model without sentiment and a MAPE of 1.05 for the LinearSVR model without sentiment. This result indicates that LinearSVR can sporadically outperform the LSTM model.

4.7 Conclusion

This chapter provides the results and discussion of techniques assessed in the previous chapter. The results, in conjunction with the discussion of findings, are divided into five sections. The metrics used to evaluate the models' performance were presented in the first section. The metrics were shown to aid in the evaluation of the models, along with a comparison with other researchers' studies.

The second section displayed results in correlation to stock movement prediction. Both models were compared with the MDA metric. The highest MDA resulted in 0.57 for the LinearSVR and 0.56 for the LSTM. The result of the investment strategy proposed in the previous chapter was presented in the third section. The results are based on an experiment conducted to answer a research question about sentiment analysis. The experiment compared both models with a range of features to evaluate the effect of sentiment analysis on the stock market. The highest performing model, which is the LSTM

with sentiment scores, produced a 28.8446% profit. An analysis of the stock stickers was created to compare and evaluate the performance of the models. This included the historical data and the yearly volatility represented by a chart. The results demonstrate that sentiment does have an impact on the stock prices.

The fourth section determined the technique that is more appropriate for stock prediction using several metrics. Results indicate that the LinearSVR model cannot process sentiment as efficiently as the LSTM model. Nevertheless, the best result was achieved by the LinearSVR model with a MAPE of 1.05. The results vary depending on the ticker. It was also observed that due to the negative correlation of the sentiment score, the LSTM was affected more negatively when compared to LinearSVR.

Lastly, the final section outlined the comparison between this study and similar work covered in the previous chapters. The MDA, when compared to other studies, underperformed in terms of accuracy. This was attributed to the fact that outcomes vary depending on the dataset. Therefore a conclusion on which model is better could not be compiled since the studies mentioned use different datasets.

Chapter 5

Conclusions and Recommendations

A technique for stock prediction was presented in this study. The main aim of this research was to predict stocks using sentiment analysis and historical data. Several targets were established in order to reach this goal.

The initial target was to gather data to use for sentiment analysis. The data gathered contained tweets and financial news headlines. A pre-trained sentiment analyser tool produced sentiment scores from the data collected. The scores were combined with historical data depending on the tickers used, as described in Chapter 3.4.2. This data was utilised to train two separate models, one of which was a deep learning architecture, and the other was a support vector machine. Experiments were conducted to answer the research questions mentioned in the previous chapter. The results of the experiments are presented in Chapter 4, along with the metrics used.

5.1 Limitations and Recommendations

During this research, several limitations were encountered. This part of the chapter outlines the main restrictions faced and recommendations for mitigating them.

Social media data gathering – As outlined in the results presented in Chapter 4.6,

the study by Shah et al. (2018), StockTwits was used as a dataset for sentiment analysis. The initial plan for this study was the usage of StockTwits instead of Twitter to gather social media data related to the described tickers. Further investigation revealed that StockTwits was still working on its official API at the time of data collection. Therefore, Twitter was used for social media data gathering. The official Twitter API limited the amount of data gathered, which was another restricting factor in this study. A combination of social media data collected from StockTwits and Twitter can be used as a further improvement to this study.

Financial headlines data gathering – Several limitations were encountered during this data gathering process. The initial restriction was that most financial websites provided no free API for gathering data. Unofficial websites for financial news gathering contained an API, which had a limitation of up to a year. Therefore a web scraper has been built for every financial news source. The scrapers encountered other limitations, for example, some financial websites detecting the scraper as a bot, thus some financial news sources were excluded from this study. This can be improved by making use of websites that provide historical data on financial news websites. This allows gathering financial news headlines from a more extensive range of sources. However, these services are not free and can be expensive for extensive research.

Pre-trained sentiment analyser tool – This study used VADER to retrieve sentiment scores based on Twitter and financial headlines. From the study, it was observed that VADER was unable to process slang related to the financial sector. Training a custom sentiment analyser model based on a dataset related to economic news is recommended.

Hyperparameter optimisation – This technique was used to find the best parameters for the models. As explained in detail in Chapter 3.5.2, hyperparameter optimisation takes parameters as input and finds the best combination between the given parameters.

A hardware limitation was noted during this process since, with the given parameters, GridSearchCV would crash due to insufficient memory when used to find the best parameters for LSTM. Therefore, to counter this issue, RandomisedSearchCV was used. This can be improved with upgraded PC parts allowing the IDE to handle more memory, although this approach will be expensive.

5.2 Closing Statement

This study presented a deep learning based model for stock prediction using sentiment analysis. The method proposed in the previous chapters was based on multiple machine learning models used for comparison. The models were trained and tested using different target and feature combinations. Both models cannot consistently predict stock price movements from the results gathered. The investment strategy was conducted to study the relation between stock prices and sentiment scores. From the tables provided, both models produced greater profits with the inclusion of sentiment scores. Therefore, sentiment is an essential factor for stock prediction. Multiple metrics were used to find an appropriate technique for stock prediction. The results show that LinearSVR produced better scores when compared to the LSTM model.

Nevertheless, the inability to process the sentiment score feature effectively could affect the scalability of this model. Therefore, from the results gathered, the LSTM model proposed is an appropriate technique for stock prediction since it can process volatility better than the LinearSVR model, and it can generate more profit using the investment strategy proposed.

Appendices

Ticker	Steps	Investment Start	Investment End	Profit/Loss	Total Indicators
AAPL	1	1000	1350.186428	350.18643	195
GOOGL	1	1000	1326.249097	326.2491	253
MSFT	1	1000	1322.369393	322.36939	201
FB	1	1000	1309.868037	309.86804	191

Table 1: Investment results for base LSTM 1-Day Future

Ticker	Steps	Investment Start	Investment End	Profit/Loss	Total Indicators
GOOGL	5	1000	1593.95869	593.95869	147
MSFT	5	1000	1314.057576	314.05758	145
AAPL	5	1000	1174.201931	174.20193	167
FB	5	1000	1171.272737	171.27274	163

Table 2: Investment results for base LSTM 5-Day Future

Ticker	Steps	Investment Start	Investment End	Profit/Loss	Total Indicators
AAPL	5	1000	1513.134825	513.13483	297
MSFT	5	1000	1390.476033	390.47603	297
FB	5	1000	1025.324551	25.324551	36
GOOGL	5	1000	1000.690546	0.6905456	2

Table 3: Investment results for LinearSVR 5-Day Future with sentiment scores

Ticker	Steps	Investment Start	Investment End	Profit/Loss	Total Indicators
AAPL	1	1000	1287.710981	287.71098	296
MSFT	1	1000	1076.696278	76.696278	214
FB	1	1000	1036.479385	36.479385	70
GOOGL	1	1000	1012.7281	12.7281	69

Table 4: Investment results for LinearSVR 1-Day Future with sentiment scores

Ticker	Steps	Investment Start	Investment End	Profit/Loss	Total Indicators
AAPL	5	1000	1499.238309	499.23831	297
MSFT	5	1000	1385.894783	385.89478	297
FB	5	1000	1016.792743	16.792743	38
GOOGL	5	1000	1000.690546	0.6905456	2

Table 5: Investment results for base LinearSVR 5-Day Future

Ticker	Steps	Investment Start	Investment End	Profit/Loss	Total Indicators
AAPL	1	1000	1156.357291	156.35729	296
MSFT	1	1000	1071.411118	71.411118	212
FB	1	1000	1035.796295	35.796295	100
GOOGL	1	1000	1017.491394	17.491394	66

Table 6: Investment results for base LinearSVR 1-Day Future

Bibliography

Ahmad, M., Aftab, S. & Ali, I. (2017), 'Sentiment analysis of tweets using svm', *Int. J. Comput. Appl* **177**(5), 25–29.
URL: *http://dx.doi.org/10.5120/ijca2017915758*

Batra, R. & Daudpota, S. M. (2018), Integrating stocktwits with sentiment analysis for better prediction of stock price movement, *in* '2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)', IEEE, pp. 1–5.
URL: *https://doi.org/10.1109/ICOMET.2018.8346382*

Bhattacharjee, I. & Bhattacharja, P. (2019), Stock price prediction: A comparative study between traditional statistical approach and machine learning approach, *in* '2019 4th International Conference on Electrical Information and Communication Technology (EICT)', IEEE, pp. 1–6.
URL: *https://doi.org/10.1109/EICT48899.2019.9068850*

Blasco, N., Corredor, P. & Ferreruela, S. (2012), 'Does herding affect volatility? implications for the spanish stock market', *Quantitative Finance* **12**(2), 311–327.
URL: *https://doi.org/10.1080/14697688.2010.516766*

Chen, K., Zhou, Y. & Dai, F. (2015), A lstm-based method for stock returns prediction: A case study of china stock market, *in* '2015 IEEE international conference on big

data (big data)', IEEE, pp. 2823–2824.
URL: *https://doi.org/10.1109/BigData.2015.7364089*

Chen, R. & Lazer, M. (2013), 'Sentiment analysis of twitter feeds for the prediction of stock market movement', *stanford edu Retrieved January* **25**, 2013.

Chong, E., Han, C. & Park, F. C. (2017), 'Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies', *Expert Systems with Applications* **83**, 187–205.
URL: *https://doi.org/10.1016/j.eswa.2017.04.030*

Elbagir, S. & Yang, J. (2019), Twitter sentiment analysis using natural language toolkit and vader sentiment, *in* 'Proceedings of the international multiconference of engineers and computer scientists', Vol. 122, p. 16.

Gupta, N. et al. (2013), 'Artificial neural network', *Network and Complex Systems* **3**(1), 24–28.
URL: *https://iiste.org/Journals/index.php/NCS/article/view/6063/6019*

Haddi, E., Liu, X. & Shi, Y. (2013), 'The role of text pre-processing in sentiment analysis', *Procedia computer science* **17**, 26–32.
URL: *https://doi.org/10.1016/j.procs.2013.05.005*

Haykin, S. & Network, N. (2004), 'A comprehensive foundation', *Neural networks* **2**(2004), 41.

Ho, T.-T. & Huang, Y. (2021), 'Stock price movement prediction using sentiment analysis and candlestick chart representation', *Sensors* **21**(23), 7957.
URL: *https://doi.org/10.3390/s21237957*

Hochreiter, S. & Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.
URL: *https://doi.org/10.1162/neco.1997.9.8.1735*

Jin, Z., Yang, Y. & Liu, Y. (2020), 'Stock closing price prediction based on sentiment analysis and lstm', *Neural Computing and Applications* **32**(13), 9713–9729.
URL: *https://doi.org/10.1007/s00521-019-04504-2*

Lakshminarayanan, S. K. & McCrae, J. P. (2019), A comparative study of svm and lstm deep learning algorithms for stock market prediction., *in* 'AICS', pp. 446–457.

Liddy, E. D. (2001), Natural language processing., *in* 'Encyclopedia of Library and Information Science', 2nd Ed. NY. Marcel Decker, Inc.

Mazur, M., Dang, M. & Vega, M. (2021), 'Covid-19 and the march 2020 stock market crash. evidence from s&p1500', *Finance Research Letters* **38**, 101690.
URL: *https://doi.org/10.1016/j.frl.2020.101690*

Medhat, W., Hassan, A. & Korashy, H. (2014), 'Sentiment analysis algorithms and applications: A survey', *Ain Shams engineering journal* **5**(4), 1093–1113.
URL: *https://doi.org/10.1016/j.asej.2014.04.011*

Nelson, D. M., Pereira, A. C. & de Oliveira, R. A. (2017), Stock market's price movement prediction with lstm neural networks, *in* '2017 International joint conference on neural networks (IJCNN)', IEEE, pp. 1419–1426.
URL: *https://doi.org/10.1109/IJCNN.2017.7966019*

Nygren, K. (2004), 'Stock prediction–a neural network approach', *Royal Instiute of Technology* pp. 1–34.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau,

D., Brucher, M., Perrot, M. & Duchesnay, E. (2011), 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830.

Samarawickrama, A. & Fernando, T. (2017), A recurrent neural network approach in predicting daily stock prices an application to the sri lankan stock market, *in* '2017 IEEE International Conference on Industrial and Information Systems (ICIIS)', IEEE, pp. 1–6.
URL: *https://doi.org/10.1109/ICIINFS.2017.8300345*

Shah, D., Campbell, W. & Zulkernine, F. H. (2018), A comparative study of lstm and dnn for stock market forecasting, *in* '2018 IEEE International Conference on Big Data (Big Data)', IEEE, pp. 4148–4155.
URL: *https://doi.org/10.1109/BigData.2018.8622462*

Staff Report on Algorithmic Trading in U.S. Capital Markets (2020).
URL: *https://www.sec.gov/files/marketstructure/research/algo_trading_report_2020.pdf*

Timmermann, A. & Granger, C. W. (2004), 'Efficient market hypothesis and forecasting', *International Journal of forecasting* **20**(1), 15–27.
URL: *https://doi.org/10.1016/S0169-2070(03)00012-8*

Wang, B., Huang, H. & Wang, X. (2012), 'A novel text mining approach to financial time series forecasting', *Neurocomputing* **83**, 136–145.
URL: *https://doi.org/10.1016/j.neucom.2011.12.013*

Wang, C., Du, W., Zhu, Z. & Yue, Z. (2020), 'The real-time big data processing method based on lstm or gru for the smart job shop production process', *Journal of Algorithms & Computational Technology* **14**, 1748302620962390.
URL: *https://doi.org/10.1177/1748302620962390*