Technical and Sentimental Analysis for Stock Price Prediction Using LSTM and GRU

Author: Muneerah Alhajri

Supervisor: Dr. Yuhua Li

A dissertation submitted in partial fulfilment of the requirements for the degree of:

MSc Artificial Intelligence

School of Computer Science & Informatics

Cardiff University

October 2022

Acknowledgments

Glory be to Allah (S.W.A), Who only has the power to honour whom He please, and to abase whom He please. Verily no one can do anything without His will. There is nothing which can payback for His bounties throughout my research period to complete it successfully.

Words cannot express my gratitude to my supervisor Dr. Yuhua Li for his consistent support and guidance, and for the thoughtful comments and recommendations on this dissertation. The weekly meetings and conversations were a source of inspiration and motivation to think critically about my dissertation and improve it.

Lastly, I would be remiss in not mentioning my family, especially my spouse. Their belief in me has kept my spirits and motivation high during this process. I would also like to thank my son for all the entertainment and emotional support.

Muneerah Alhajri

Contents

List of Figures

List of Tables

Abstract

Any thriving and competitive economy depends heavily on the stock market. By making shares of a company publicly available, it aids in its financial growth. Additionally, it enables people to invest in those businesses and earn from doing so. However, due to the volatility of the stock market, stock trading entails a certain risk. This study aims to minimize this risk by providing a solution for stock price prediction. The solution is based on predicting the stock price using a combination of technical and sentimental indicators. The used technical indicators are SMA, EMA, MACD, RSI, and Momentum. To measure their influence on the stock price, sentiment analysis is applied to the stock-related tweets. The BERT model is used for sentiment analysis with a classification accuracy of 82.88%. The sentimental indicators are derived from the sentiment score, and they are: Averaged Weighted Sentiment, Sentiment Moving Average, Tweets Volume, and Tweets Moving Average. The technical and the sentimental data are then fed to the machine learning models i.e. LSTM, and GRU to predict the stock price. During the experiment, the RMSE is reduced from 13.83 to 0.33. The first RMSE results from using LSTM with only technical indicators. The second RMSE results from using GRU with a set of technical and sentimental indicators. These numbers show how the sentimental analysis contributes to better stock price prediction. This study also demonstrates the superiority of GRU over LSTM in the field of stock market prediction.

Chapter 1

Introduction

The stock market is a primary part of any robust and competitive economy. It helps companies grow financially by offering shares to the public. It also allows individuals to invest in those companies and make profits. However, stock trading involves some risk attributed to the unpredictability of the stock market. Two popular theories i.e. the random-walk theory (RWT) [2], and the efficient market hypothesis (EMH) [1] claim that the stock market is random and not predictable, in their short. Nevertheless, the research in the field of stock market prediction, and successful trading strategies show how one can beat the market with efficient analysis of the available information and timed actions. In this study, I support the latter view by exploring efficient solutions to predict the stock market.

1.1 Problem Statement and Motivation

The stock market is so volatile and is affected by many factors some are quantitative and some are not. The quantitative factors come in the form of past prices of the stock, yearly financial reports, and so on. The qualitative factors are unmeasurable such as the reputation of the company, the economic state of the industry, the customers' satisfaction with the products or services of a company, the management, etc...

Technical analysis is a method of predicting the stock price based on historical price data. Thus, it handles a significant portion of the quantitative factors. Fundamental analysis can also assist in this case by studying the fundamentals of the company to derive its intrinsic value, and thus, the share value. However, there should be a way to

consider the qualitative factors and measure their influence on the daily prices of the stock market.

Traditionally, these aspects are assessed by stock market experts by analyzing economic events and changes. This information used to be in the economic news and articles. Nowadays, with the dominance of social media apps in our life, they became the main source of news, as well as, platforms for individuals to express their opinions. Somehow, social media content influences many aspects of our lives. This includes the stock market prices. For example, negative reviews about a certain product might cause the stock price of the product's company to decline. This makes social media content a valuable source of information to analyze when predicting the stock price.

However, the question that arises is to what extent a negative/positive review affects the stock price. A brute-force approach suggests manually going through the endless content and trying to weigh the thoughts about a certain stock, and analyze them, to finally come up to a conclusion of whether that will increase or decrease the stock price. With the advancements in the field of artificial intelligence and machine learning, this solution is no longer effective and must be replaced by an intelligent solution. The use of machine learning in analyzing the text (sentiment analysis) is a growing field, and some might think it is not suitable for such a sensitive and critical application of stock price prediction. Thus, another challenge is to find a reliable way for analyzing the sentiment of social media content.

To brief, I seek to predict the stock price considering both quantitative and qualitative factors using reliable methods based on machine learning.

1.2 Proposed Solution

The best-known method of analyzing the stock market is to combine technical and fundamental analysis. Technical analysis involves the use of some mathematical measures called technical indicators. Technical indicators are computed with past price data to predict the future price of a stock. In this study, I propose to use five of the most commonly used indicators in the literature [19]: Simple Moving Average (SMA); Exponential Moving Average (EMA); Moving Average Convergence Divergence (MACD);

Relative Strength Index (RSI); and Momentum indicator. This combination includes lagging and leading indicators to provide a better sight for future price movement. The first three are lagging indicators whereas the last two are leading indicators.

Another issue I am attempting to tackle in this study is the analysis of social media content and how it influences the stock price. The social media platform adopted for this study is Twitter. Sentiment analysis is a Natural Language Processing (NLP) task used to classify the text based on its sentiment i.e. postive or negative. The state-of-the-art sentiment analysis model BERT is used to calculate the sentiment score of the stock-related tweets. BERT provides a pure sentiment score, which I believe is not representative enough. Therefore, four sentimental indicators are extracted from the original sentiment score including some statistical indicators. The indicators are: Averaged Weighted Sentiment, Sentiment Moving Average, Tweets Volumn, and Tweets Moving average. The first two indicators provide more information about the overall sentiment movement and trends. The last two are descriptive indicators that measure the data traffic about a certain stock.

The technical and the sentimental indicators are then combined and used to predict the next day's price with a look-back period of 60 days. Therefore, a data sequence of length 60 forms the input and a single price value represents the output. To process this data sequence, a machine learning model that is capable of reading and relating the whole sequence is needed. Recurrent Neural Networks (RNN) are capable of doing so by the means of looping the output of the current step as an input of the next step. This allows the network to keep a memory of previous data. This makes RNN a good option for processing time series data. However, in the case of long sequences, RNN might suffer from the so-called vanishing gradient problem [4]. LSTM and GRU are two sophisticated RNN models that can handle long sequences without running into the vanishing gradient problem by the means of selective memory. These models can remember relatively important information, and forget unnecessary details with the help of some gates.

The two models LSTM, and GRU are used in this study to predict the stock price. Then, the results are compared and analyzed.

1.3 Contributions

The major contributions of this work are summarized as follows.

First, on the technical analysis side, it is common to use technical indicators to predict the stock price. However, when combined with sentiment analysis, pure historical price data are used i.e. open, high, low, close, and volume.[19] Thus, incorporating five of the most practically effective technical indicators with the sentiment analysis is a significant input.

Second, the use of the state of the art model in sentiment analysis BERT to calculate the tweets' sentiments with a testing accuracy of 82.88%. [21] used TextBlob to predict the sentiment of the tweets which performed with 62.67% accuracy when tested on the labeled tweets dataset. [15] used Naïve Bayes voting classifier which performed with 66.78% accuracy when tested on the same dataset of labeled tweets. [20] used the LSTM model to predict the text sentiment which can have some advantages in predicting long sequences. However, BERT has the advantage over LSTM with its bidirectional ability as well as the attention mechanism built into the model.[14]

Besides the high-accuracy prediction of the sentiment, additional data were used to give more meaning to the sentiments like the number of followers and the number of re-tweets. Therefore, the tweets from users with higher follower counts are weighted more heavily. Also, the tweets with more re-tweets are assigned heavier weights. Some statistical indicators are also used such as the number of tweets per day and the tweets moving average.

1.4 Aim and Objectives

This study aims to efficiently and effectively combine technical and sentimental analysis to predict the stock market with state-of-the-art machine learning methods and tools. With this general aim in mind, the following objectives are defined :

1. Review the literature to find the most effective technical indicators.
2. Test and compare different sentiment analysis tools for Twitter sentiment analysis. The tools are VADER; TextBlob, Naïve Bayes, and BERT.

3. Apply sentiment analysis to the stock tweets.
4. Use the sentiment score to derive more sightful indicators.
5. Use descriptive statistics to extract more features about the tweets' traffic about a certain stock.
6. Combine the technical and sentimental analysis data.
7. Use RNN-based models (LSTM, and GRU) to predict the next-day price with 60 days time steps.
8. Test the models i.e. LSTM, and GRU, and compare their performance.
9. Test the influence of the sentimental analysis on prediction accuracy by muting the sentimental indicators and observing how this affects the performance of the models.

1.5 Road Map

Besides the introduction, this document comprises five more chapters: literature review, methodology, implementation, results and discussion, and the conclusion.

Chapter two is mainly designed to provide the necessary background knowledge that the reader needs to understand the remaining parts of the dissertation. It also presents a literature review to demonstrate the significance and the need for addressing the problem at hand.

Chapter three briefly describes the overall approach to solving the problem of forecasting the stock price using a combination of technical and sentimental indicators with RNN-based machine learning algorithms.

Chapter four covers more technical details about the implementation of the proposed solution including the data collection, data pre-processing, and the machine learning models design.

Chapter five reveals and interprets the evaluation results. The models are tested and compared under different conditions. The setting of the different experiments is explained and the results are discussed.

The last chapter concludes this document by summarising the overall contribution of this work, followed by noting the shortcomings of this study with some future work recommendations. Finally, I recall all the lessons learned and skills gained throughout the journey of my dissertation.

Chapter 2

Literature Review

2.1 Overview

In this chapter, I start by providing some background on the underlying concepts for a better understanding of the research problem. Then, I provide a review of the literature in the field of stock market prediction using machine learning with the major contributions of this work in the area of combining technical and sentiment analysis to predict stock prices using RNN-based models.

2.2 Background

This work incorporates knowledge from different fields including stock market analysis, Natural Language Processing (sentiment analysis), and machine learning models for time series data. In this section, I explain some concepts from these fields that are closely related to our study. In the first part of this section, I explain the stock market analysis techniques and indicators. In the second part, I describe some sentiment analysis tools and models used in the literature. Finally, I describe two machine learning models namely LSTM and GRU that are used in this study.

2.2.1 Stock Market Analysis

In the stock market world, one should have strategic approaches for trading or investment to achieve maximum profits and minimum losses. In general, there are two main methods of analyzing the stock market namely fundamental and technical analysis. Fun-

damental analysis is usually used for long-term investment whereas technical analysis is best suited for short-term trading. In the following subsections, I will provide brief descriptions of the fundamental, and technical analysis, and some of their indicators.

Fundamental Analysis

Fundamental analysis is a way to calculate the true intrinsic value of a company or a share using some economic factors named fundamentals. The study of these fundamentals should follow a top-down approach also known as EIC (Economy-Industry-Company analysis) framework.[3] Through the EIC framework, one needs to first consider the whole economy when doing the fundamental analysis. Based on the results of the first analysis, the industry in which the company operates can be studied. After that, the company's financial status is evaluated keeping in mind the economic and industrial conditions.

I will only focus on the company-level analysis because it relates more to this study. The company-level analysis includes financial/quantitative analysis and non-financial/qualitative analysis.

1. Quantitative analysis: This is about studying the financial aspects of a company such as sales, revenues, profits, assets, expenses, liabilities, and losses. Financial ratios are quantitative measures to assess these aspects. The income statement of the company as well as the balance sheet are the most used financial report to calculate such ratios. Table 2.1 lists the most common financial ratio along with the financial document used to compute them. Besides these documents, the analysts use the current price of the share in the stock market to calculate some financial ratios. The last column of the table shows some common examples of such ratios.[19]

Income Statement	**Balance sheet**	**Both**	**Stock Price**
Gross profit %	Current ratio	Return on equity	Earnings per share
Operating margin %	Working capital ratio	Asset turnover	Price-earnings Ratio
Return on sale %	Dept to equity	Receivable turnover	Price-sales Ratio
	Equity-total %	Inventory turnover	Price-book ratio

Table 2.1: Financial Ratio According to their Source(s)

2. Qualitative analysis: This is concerned with the study of fundamentals that can not be quantified such as the management, the image of the company, and the product's quality. Economic news, product reviews, and social media content can give good insights into such fundamentals. Investors used to check the daily economic news and look for investment opportunities. With the massive amount of daily content, especially with social media becoming a major source of news, opinions, and discussions, and the busy lifestyle of many of us, it became inconvenient to follow up with all the latest news. The tremendous advancement in machine learning in general and Natural Language Processing NLP, in particular, made it possible to automate this task. The text can be input into an NLP model to extract some meaningful information. A particular type of NLP task is sentiment analysis which is a method of classifying a text as either positive, negative, or neutral. In this study, sentiment analysis is used to predict the stock market. I will be referring to it as a sentimental analysis. Throughout this document, sentiment analysis refers to the NLP task of classifying the text into either positive or negative, whereas sentimental analysis refers to the use of sentiment analysis in predicting the stock market.

Figure 2.1 summarise the different stages and types of fundamental analysis. In this figure, sentimental analysis is shown as a form of qualitative analysis to measure public opinions and satisfaction with a certain firm. However, it is worth mentioning that some other textbooks and papers regard sentimental analysis as a third type of analysis besides fundamental and technical analysis. The difference categorization should not be a problem as long as the definition of sentimental analysis in the stock market is one.

Fundamental analysis aims to derive the true intrinsic value of a share. The value is to be compared with the current market price to find selling or buying opportunities. One common use of fundamental analysis is to study the fundamentals of a company to make an investment decision. Therefore, fundamental analysis might be used to decide what stock to buy/sell.

Technical Analysis

Technical analysis is concerned with forecasting the market price of a stock based on the historical price data of the same stock. Technical analysis does not consider any external

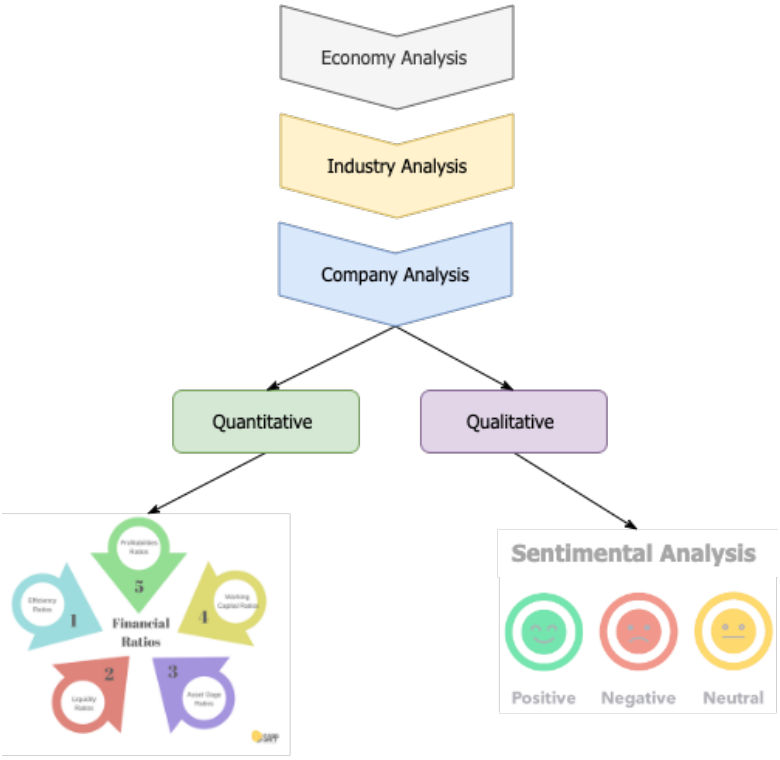

Figure 2.1: Fundamental Analysis Phases and Types

economic factors but the past price movement of the stock. Technical analysts claim that the movement patterns of the price repeat themselves and thus one can utilize that to make a trading decision. Some of the commonly used tools by technical analysts are described below.

Charts

Charts are what traders look at when doing technical analysis for stock prices. There are many types of charts used, the most common ones are:

- Line Chart: This is the simplest used chart for technical analysis. The x-axis corresponds to the time and the y-axis corresponds to the stock price. The chart is plotted by connecting the closing price (or mid-price) points for the given time interval.
- Bar Chart: This chart is a sequence of bars, one bar for every time interval. The bar has four main elements: Open - the opening price of the stock at the beginning of the bar's time interval; High- the highest price reached during the bar's time

interval; Low- the lowest price reached during the bar's time interval; Close- the closing price of the stock at the end of the bar's time interval. Figure 2.2 depict the anatomy of the bar which is also called the OHLC bar

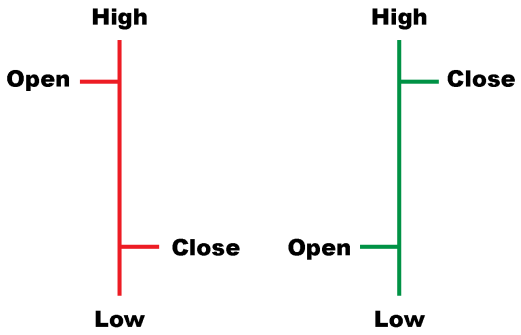

Figure 2.2: OHLC Bar

- Candlestick Chart: This is the most commonly used chart for technical analysis. The candlestick plotting mechanism was originated in Japan. As shown in figure 2.3, the candlestick has a body and a wick also called a shadow. The body represents the price range between Open and Close. The upper and the lower shadows represent the High and Low, respectively.

Figure 2.4 visualizes the three common types of charts discussed above.

Trends

The trend is the broad direction of stock price which is either upward or downward direction. The recognition of such trend patterns is an essential step when using technical analysis for trading. An uptrend happens when the stock price makes successive higher highs and higher lows. In contrast, a downtrend occurs when making lower highs and lower lows. Figure 2.5 illustrate the concept.

A trend might also reverse direction. This occurs when an upward trend becomes a downward trend and vice versa. For a long-term trend, the trend continues in one direction for a long time before it reverses.[26] The opposite applies to short-term trends. The down reversal i.e. uptrend becoming downtrend, suggests a selling signal while the up reversal suggests a buying signal.

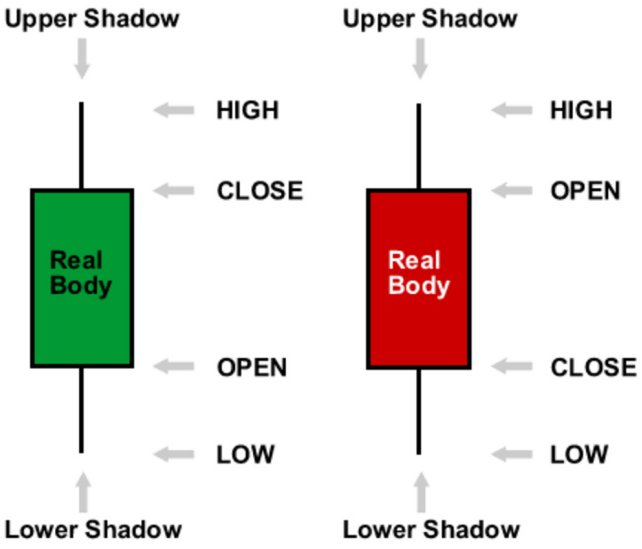

Figure 2.3: Candlesticks

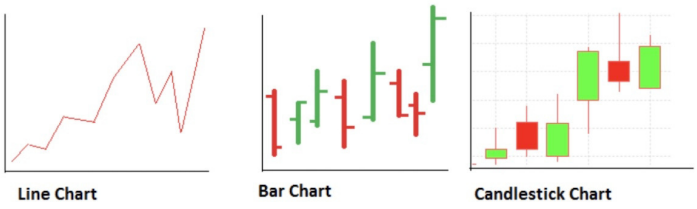

Figure 2.4: Types of Technical Analysis Charts

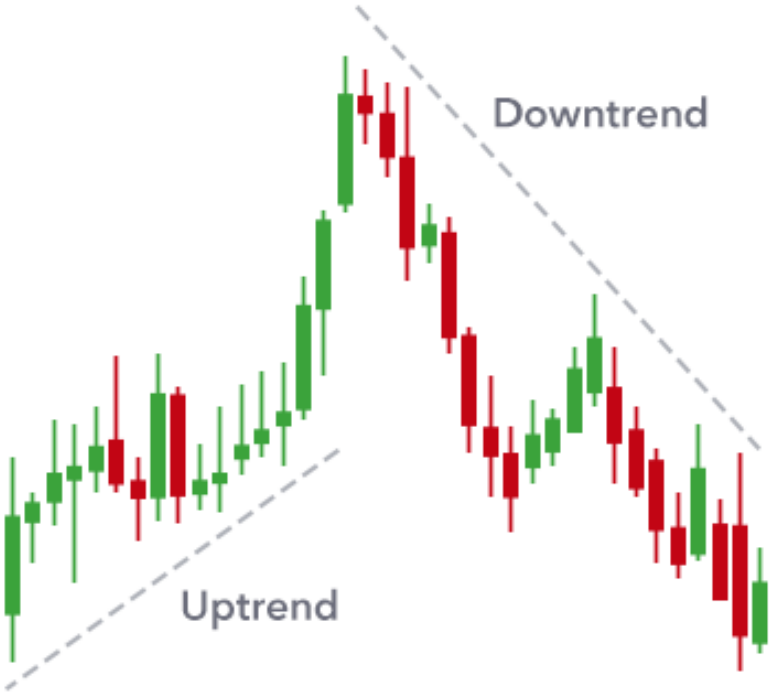

Figure 2.5: Uptrend vs Downtrend

Support and resistance are two powerful trend lines that are usually extended to predict future price movements. Support is an uptrend line that indicates the superiority of buyers over sellers and thus the stock price is supported by this line. On the other hand, resistance represents a downtrend line which indicates the superiority of sellers over buyers, and thus the stock price is capped by this line. Figure 2.6 presents an example of support and resistance lines. In certain cases, a "breakout" happens when the price penetrates either the support or resistance lines.

Price Patterns

Besides the upward and downward trend patterns, some defined patterns are worth analyzing in certain situations. However, I am not going to elaborate on those patterns as they are not essential for this study. Examples of price patterns are:

1. Head and shoulders.
2. Double tops and bottoms.
3. Triangles.

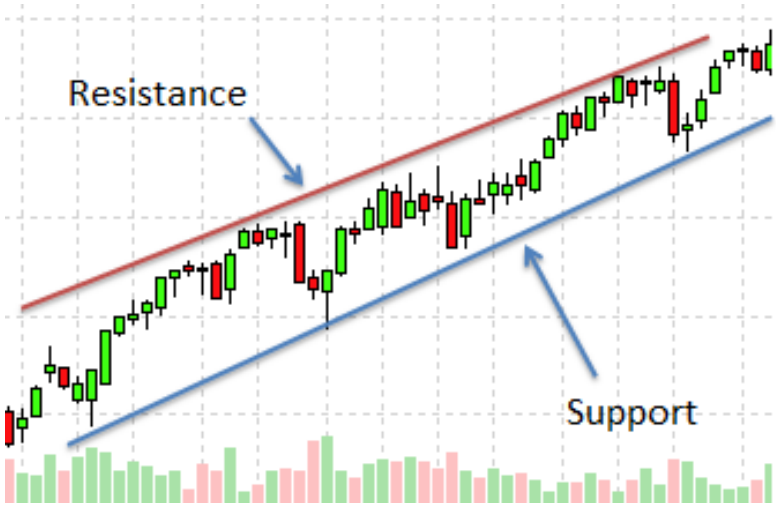

Figure 2.6: Support and Resistance Lines

4. Rectangles.[26]

Indicators

Technical indicators are mathematical formulas that make use of past price values to establish useful knowledge about the future movement of the stock price. Indicators are used to alert buy/sell signals, confirm a trading decision, or forecast the future price movement. There are two types of technical indicators namely leading indicators and lagging indicators.

Leading indicators give insights into the price movement before it takes place. They attempt to predict the future price trend/direction. Some popular examples are:

1. Momentum (MOM).

   The momentum indicator measures the speed (or strength) of a price movement. It is used to predict an upward/downward trend. At each point, the Momentum indicator is calculated by comparing the current price with a previous price, using the following formula.

$$Momentum_i = \frac{price_i}{price_{i-period}} \tag{2.2.1}$$

2. Relative Strength Index (RSI).

   RSI is used to evaluate overbought/oversold conditions by measuring the velocity and magnitude of price movement At each point, the RSI indicator is calculated

using the following formulas. First, we calculate the upward U and the downward D change as shown in equations 2.2.2 and 2.2.3, respectively. Then, they are averaged as shown in equations 2.2.4, and 2.2.5. The final RSI value is computed using the equation 2.2.6

$$U_i = \begin{cases} close_i - close_{i-1}, & \text{if } close_i > close_{i-1}. \\ 0, & \text{if } close_i close_{i-1}. \end{cases} \tag{2.2.2}$$

$$D_i = \begin{cases} close_{i-1} - close_i, & \text{if } close_{i-1} > close_i. \\ 0, & \text{if } close_{i-1} close_i. \end{cases} \tag{2.2.3}$$

$$MA_{U_i} = U_i + MA_{U_{i-1}} * \frac{period - 1}{period} \tag{2.2.4}$$

$$MA_{D_i} = D_i + MA_{D_{i-1}} * \frac{period - 1}{period} \tag{2.2.5}$$

$$RSI_i = 100 - 100 * \frac{1}{1 + \frac{MA_{U_i}}{MA_{D_i}}} \tag{2.2.6}$$

3. Rate of Change (ROC).

   As the name implies, ROC is used to measure the rate of change in the price between the closing price of today and the closing price of *period* days ago. At each point, the ROC indicator is calculated using the following formula.

$$ROC_i = \frac{close_i - close_{i-period}}{close_{i-period}} \tag{2.2.7}$$

Lagging indicators, on the other hand, analyze past price behavior and provide delayed feedback to the trader about a long-term trend/direction. It can be used to confirm a price trend and thus help the trader to make an informed trading decision. Most of the technical indicators fall in this category, the following are some common examples:

1. Simple Moving Average (SMA).

   SMA is simply the mean of all values of x from a certain *period* to the current value. x could be the closing price values, and the *period* could be in days, hours,

or even minutes. At each point, the SMA indicator is calculated using the following formula.

$$SMA_i = \frac{\sum_{k=i-period}^{i} x_k}{period} \tag{2.2.8}$$

2. Exponential Moving Average (EMA).

   EMA applies exponentially decreasing weighting factors. In order to provide more weight to recent observations while still keeping previous observations in consideration, the weighting for each older data point drops exponentially over time. At each point, the EMA indicator is calculated using the following formula.

$$EMA_i = EMA_{i-1} + \frac{2}{period + 1} * (x_i - EMA_{i-1}) \tag{2.2.9}$$

3. Moving Average Convergence Divergence (MACD).

   MACD is used to calculate the difference between the fast and the slow EMA. At each point, the MACD indicator is calculated by the given formula where $EMA[fastperiod]_i$ and $EMA[slowperiod]_i$ are calculated using EMA formula 2.2.9. The default values for the slow period and the fast period are 26 and 12, respectively.

$$MACD_i = EMA[fastperiod]_i - EMA[slowperiod]_i \tag{2.2.10}$$

For all the above formulas of indicators [23], we have:

- *period* - the period of time for which the calculation of the indicator is done.
- x_i - an instance values from the data source.

2.2.2 Sentiment Analysis

In the previous section, we came across sentimental analysis as a form of qualitative fundamental analysis that uses the sentiment analysis for stock market prediction. The aim of introducing it earlier is to put it in the context of stock analysis. However, this section is dedicated to discussing the technical aspects of sentiment analysis and some commonly used tools in practice.

What is sentiment analysis?

Sentiment analysis is a Natural Language Processing NLP technique used to classify a text as either positive, negative, or neutral. It has a wide variety of applications specifically in the business industry to measure the customers' satisfaction and understand their needs. It can also be used in the stock market world to gain some insights into the future movement of a stock price which is the aim of using sentiment analysis in this study.

Sentiment analysis is not only about classifying the text into positive, negative, and neutral. Some of the common tasks of sentiment analysis are graded(fine-grained) polarity sentiment analysis, emotion detection, and aspect-based sentiment analysis. In the basic sentiment analysis task, we have 3 levels of polarity i.e. positive, negative, and neutral. However, in certain applications, higher polarity precision is required. In such cases, the graded polarity sentiment analysis is best suited. It provides 5 levels of polarity: very positive; positive, neutral, negative, and very negative. This is commonly used in the 5 stars reviews.

Emotion detection as the name implies is used to detect the emotion in the text. Emotion could be happiness, sadness, anger, hope, etc.. and thus classify the text accordingly. Aspect-based sentiment analysis, on the other hand, applies polarity analysis to the text with an additional feature of detecting which aspect is being classified as positive, negative, or neutral. For example, Apple is collecting reviews from customers about the latest iPhone. There are some positive as well as negative reviews. To improve the product and raise customer satisfaction, Apple needs to know which aspect of iPhone was driving the negative reviews. Let us say all the negative reviews were about the price, battery, or weight of the device. Then, Apple could focus on improving these aspects in future releases. Thus, aspect-based sentiment analysis provides even more than just polarity analysis.

Approaches

There are different ways of implementing sentiment analysis, but they fall into one of the three categories.

1. Rule-based Approach

This approach is based on a manually written set of rules for measuring the polarity, emotion, or subjectivity of the text. For example, one might make two lists of positive and negative words. Given a text, it counts the number of positive and negative words. If the number of positive words is more than negative words, then classify as positive and vice versa. This approach does not consider the whole sequence of the text. Even if more rules are added to address these issues, the system might become very complex to manage and the rules might override one another.

2. Automatic Approach

   This approach is based on using machine learning techniques to learn the sentiment from the text itself. Figure 2.7 describes the process of machine learning-based sentiment analysis. In the training phase, we have the textual data and their tags/labels. The text is pre-processed and converted into a features vector. The feature vectors along with their associated tags are fed to the machine learning algorithm to start the training process. Once the training is done, the trained model is tested on unseen data. After being transformed into a suitable format, the test data are sent to the model to predict their tag. The predicted tag is compared with the actual tag for evaluation of the model performance.

3. Hybrid Approach

   This approach takes advantage of both rule-based and automatic approaches by combining them. In this approach, you can have some custom rules tailored to your specific application which can guide the prediction of the machine learning algorithm. The performance of hybrid-based models is often more accurate compared to single approach models.

Tools

There are many available tools and pre-trained models to use for sentiment analysis. In this section, I will present four examples of sentiment analyzers i.e. VADER, TextBlob, Naive Bayes, and BERT. The first two are rule-based sentiment analyzers whereas the latter two are machine learning-based algorithms for sentiment analysis.

1. **VADER**

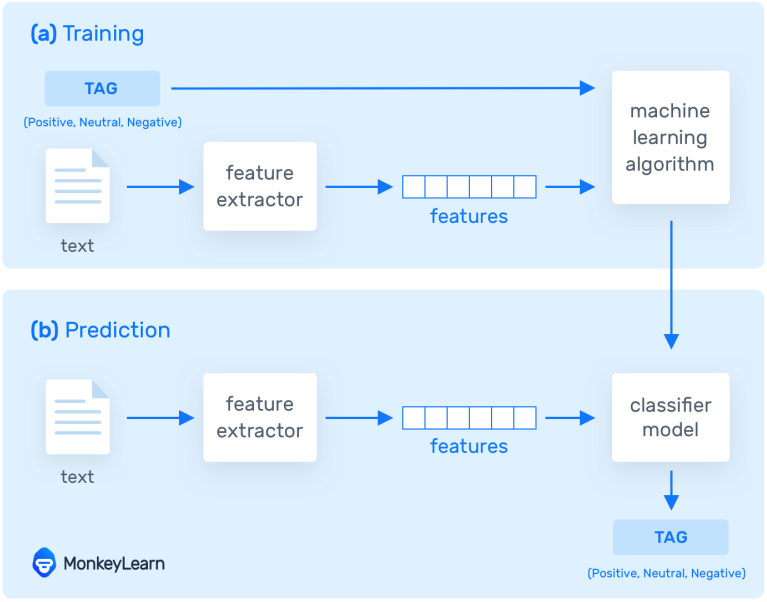

Figure 2.7: Sentiment Analysis Process

VADER (Valence Aware Dictionary for sEntiment Reasoning) was introduced in 2014 by C. Hutto, and Eric Gilbert in their paper [8]. It is a lexicon and rule-based sentiment analysis tool that is specifically designed for social media text analysis. In VADER, the sentiment of a text is returned as a numerical sentiment score ranging from -1 (most negative) to 1 (most positive).

As defined above, VADER has two major components: the lexicon or the dictionary, and the analyzer that makes use of the dictionary to predict the sentiment score of a text. The dictionary is constructed by human raters to map the lexical features i.e. words, slang, acronyms, and emoticons to sentiment scores. To increase the reliability of the dictionary, many raters are involved in the mapping process, then the scores of different raters are averaged to give the final score. This method benefits from the so-called "wisdom of the crowd" to raise confidence in the lexicon.

The second part of VADER, the analyzer, uses the built dictionary to predict the sentiment score of the text along with five rules/heuristics. These heuristics consider other factors that affect the sentiment of the text such as punctuation, capitalization, etc... The following are the five heuristics:

(a) Punctuation

VADER takes into consideration the punctuation marks by either adding or subtracting a certain amount to/from the sentiment score to address the effect of the punctuation. For example, the two texts " I love it." and "I love it!!!" are expressing positive emotions. However, it is obvious that the second one has stronger emotions due to the use of the exclamation marks. In this case, VADER calculates the sentiment score of the sentence. Then, if the sentiment is positive, like in this example, it adds a certain amount i.e. 0.292 for every exclamation mark which gives it a higher score compared to the first sentence. If the text was negative, it subtracts the same amount. The amount to be added/subtracted for every punctuation is already calculated.

(b) Capitalization

The capitalization is treated exactly like the punctuation except that it is given different weights. So, a word with a positive sentiment score is increased by 0.733 when capitalized. On the other hand, a word with a negative sentiment score is decreased by 0.733 when capitalized.

(c) Modifiers

There are certain words (modifiers) that when proceeding a text, they either increase its intensity or decrease it. For example, we have “effing cute” and “sort of cute”. , the modifier "effing" has increased the intensity of cute compared to the second modifier. Thus, VADER has a modifiers dictionary that contains all the positive and negative modifiers. The amount to be added/subtracted differs for the same modifier according to the distance between the base word and the modifier.

(d) “but”

Usually, "but" connects two phrases with opposite sentiments. However, the later phase has dominance. For example, "I love math, but I fail the test". The first part "I love math" is positive whereas the second one " I fail the test" is negative. Thus, this sentence should have an overall negative sentiment score. VADER handles such cases through the use of the so-called "but" checker. It works by reducing the score by 50% for all the words before the "but" and increasing the score by 150% for all the words after "but".

(e) Negation

VADER handles the negation by multiplying the sentiment score of the negated text by -0.74. It maintains a dictionary of all the negation words.[12]

2. **TextBlob**

   TextBlob is a python library for NLP tasks created by a team of developers led by Steven Loria.[18] TextBlob provides handy APIs for sentiment analysis, POS tagging (part-of-speech tagging), classification, noun phrase extraction, translation, and more. Since we are only concerned about the sentiment analysis in this study, I will briefly go through the sentiment API.

   TextBlob sentiment analyzer is a rule-based analyzer that maintains a dictionary for the words and their polarity weights to help calculate the sentiment score. In TextBlob, "sentiment" is a property that return the polarity and the subjectivity of the text. The polarity ranges from -1 to 1 where -1 for very negative and 1 for very positive. The subjectivity ranges from 0 to 1 where 0 is very objective and 1 is very subjective. For example, something like

   TextBlob("I feel great today").sentiment

   would return,

   Sentiment(polarity=0.8, subjectivity=0.75)

3. **Naïve Bayes**

   Many machine learning algorithms can be used for sentiment analysis such as Naive Bayes, Support Vector Machine SVM, Neural Networks, and ensemble algorithms. Naïve Bayes is a probabilistic machine learning algorithm based on the Bayes theorem. It is mainly used for classification tasks, although specific types of the model can be used for regression. Equation 2.2.11 state the theorem

   $$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{2.2.11}$$

   Bayes theorem calculates the probability of hypothesis A given evidence/predictor B. In the case of more than one predictor, the theorem assumes independent, and equal contribution of the predictors to the final result, this is why it is called Naïve.

Bayes theorem can be rewritten as shown in equation 2.2.12 to better fit the machine learning job.

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)} \tag{2.2.12}$$

where y is the class or label, and X is the feature vector. After expanding the y and removing the denominator, the theorem is given by equation 2.2.13

$$P(y|x_1, ..., x_n) = P(y) \prod_{i=1}^{n} P(x_i|y) \tag{2.2.13}$$

The above equation work for binary classification, in the case of multi-class classification, the value of y could be calculated by equation 2.2.14

$$y = argmax_y(P(y) \prod_{i=1}^{n} P(x_i|y)) \tag{2.2.14}$$

There are also specific forms for the regression models, however, we are skipping them here as they are not relevant to the topic.

According to the machine learning task, we can choose from three Naïve Bayes models.

(a) Multinomial Naive Bayes

This model is used for classification tasks where the feature vectors represent the occurrence frequencies of each event in a multinomial distribution. It is so commonly used for document classification and sentiment analysis.

(b) Bernoulli Naive Bayes

This model is similar to the Multinomial Naive Bayes classifier except that the features have only boolean values. This model can also be used for document classification where the binary values of a feature tell us whether this word occurs in the document or not.

(c) Gaussian Naive Bayes

This model is used for regression tasks and takes continuous values. By assumption, the input values are sampled from a gaussian distribution.

4. **BERT**

   BERT (Bidirectional Encoder Representations from Transformers) was first introduced in 2018 by Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (researchers at Google AI Language) in their paper [14]. BERT is based on transformers [13], and thus benefits from its attention mechanism. Transformers have two main components: the encoder which converts the text to an embedded format, and the decoder which predicts the output for the given task. BERT only uses the encoder part because it is just a language representation model. An output layer may be added during the fine-tuning of the model for custom applications.

   In the standard transformer, the encoder read the text sequentially in one direction (from left to right, or from right to left). However, BERT's encoder can read the text from both directions which results in better context understanding. Actually, BERT can read the whole text at once which is considered to be bidirectional. There are two main stages when working with BERT which are pre-training, and fine-tuning detailed below.

   (a) Pre-training BERT

   BERT is pre-trained using unlabelled data with 3.3 billion words from Wikipedia (2,500M words) and Google's BooksCorpus (800M words) on two unsupervised NLP tasks:

   - Masked Language Model (MLM)

     The bi-directionality of BERT is credited to the MLM technique of training. In this technique, 15% of the input sequence is masked and the model has to predict the masked words. Masking is achieved by replacing the masked word with the [mask] token.

   - Next Sentence Prediction (NSP)

     MLM does not capture the relationship between two sentences which is very important for some NLP tasks such as Question/ Answering and Natural Language Inference. To overcome this, BERT was also trained on Next Sentence Prediction (NSP) task to help BERT understand if there is a relationship between two sentences or not. This is done by providing the model with pair of sentences (A and B). Half the data are

actually two related sentences. The second half of the data is just two randomly selected sentences. In this way, the model can be trained to understand the relationship between two sentences.

Figure 2.8 illustrate the architecture of the pre-training process of BERT. Pair of unlabeled sentences are fed to the model. Some words of these sentences are masked. The model has to predict the masked words which are shown in the figure as Mask LM on the top. At the same time, the model has to say whether the two sentences are related or not through the NSP output of the model. [CLS] is a special symbol used at the beginning of each input, and [SEP] is used to separate the two sentences. The model is trained on both tasks at the same time.[14]

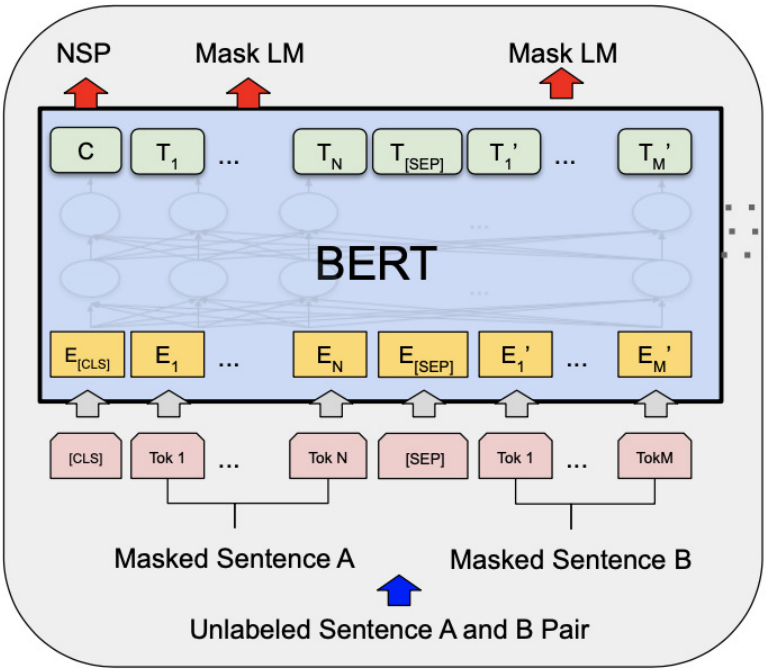

Figure 2.8: Pre-training BERT Process Architecture

(b) Fine-tuning BERT

Fine-tuning is the process of customizing the model for a downstream task. During the pre-training, BERT is trained on unlabeled data. However, during fine-tuning, we start with the pre-trained model and then use labeled data to fine-tune the model parameters for the given task. In the case of classification problems, like sentiment analysis, we use the [CLS] representation as the

label. Figure 2.9 show a custom architecture of BERT for the sentiment analysis task of tweets. The input sentence represents the tweets and the class label represents the polarity classes i.e. positive, or negative.

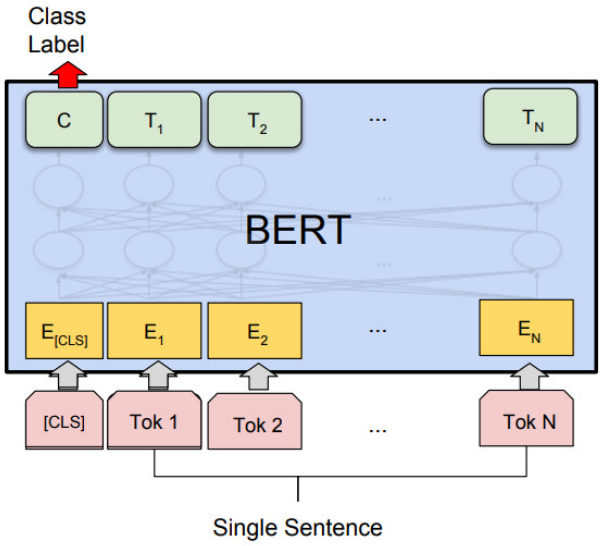

Figure 2.9: BERT for Tweet Sentiment Analysis

BERT model has two sizes i.e. base and large with a different number of parameters and encoders. Actual BERT is the large one, however, the base BERT was created to have the same size as OpenAI GPT and thus can be compared. Table 2.2 provides size-related details about the models.

Number of.	**BERT_base**	**BERT_large**
Transformer Layers	12	24
Hidden Size	768	1024
Attention Heads	12	24
Parameters	110M	340M

Table 2.2: BERT Model Sizes

2.2.3 Machine Learning Algorithms

Stock price prediction is a time-series problem, where the input is a sequence of data points. For this kind of problem, Recurrent Neural Network (RNN) is best suited for

its ability to capture the short-term dependencies in the sequence. This is achieved by the use of an internal hidden state. Figure 2.10 depicts the architecture of an RNN cell. The RNN cell receives two inputs: x_t which is a feature vector and h_{t-1} which represents the output of the previous hypothetical cell. Therefore, RNNs can remember some previous input. However, this is only short-term memory, i.e. in the case of long sequences vanishing gradient problem might occur and the effect of the very previous words will vanish [4]. To overcome this problem, long-short-term memory RNN models can be used. These models can handle long-term dependencies by some sort of selective memory. LSTM and GRU are two examples of such models which are discussed below.

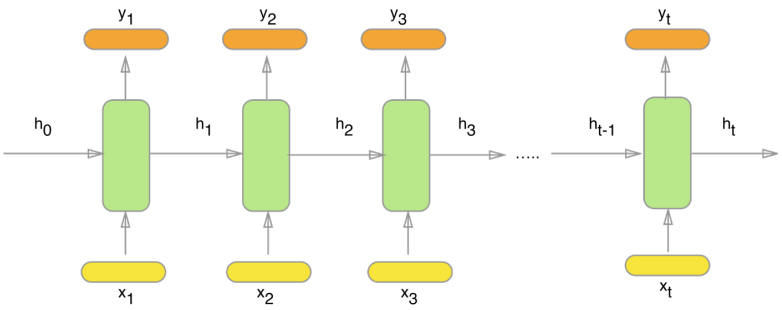

Figure 2.10: RNN Cell

LSTM

LSTM for Long-Short-Term Memory is an RNN-based model with the additional feature of being able to learn long-term dependencies. This model can avoid the vanishing gradient problem in long sequences by the means of selective memory that can remember the important parts of the sequence and forget others. This is achieved by maintaining a self-state (cell state). The cell state is controlled by three gates: forget gate, input gate, and output gate. Figure 2.11 highlights the three gates in the LSTM cell.

Like RNN, the LSTM cell receives x_t which is a feature vector, and h_{t-1} which represents the hidden state output of the previous hypothetical cell. In addition, LSTM receives the cell state variable C_{t-1} which represents the long-term memory of the LSTM cell. This is shown in figure 2.12. From the same figure, we can also see that LSTM cell output both hidden state and cell state variables that are fed to the next hypothetical

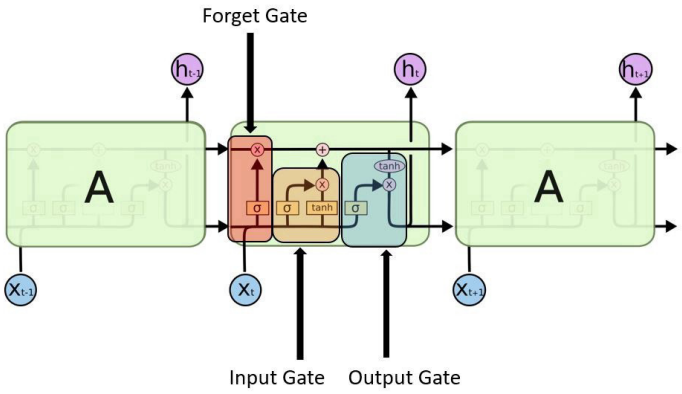

Figure 2.11: LSTM Gates

cell. The following provides a description of each gate in the LSTM cell:

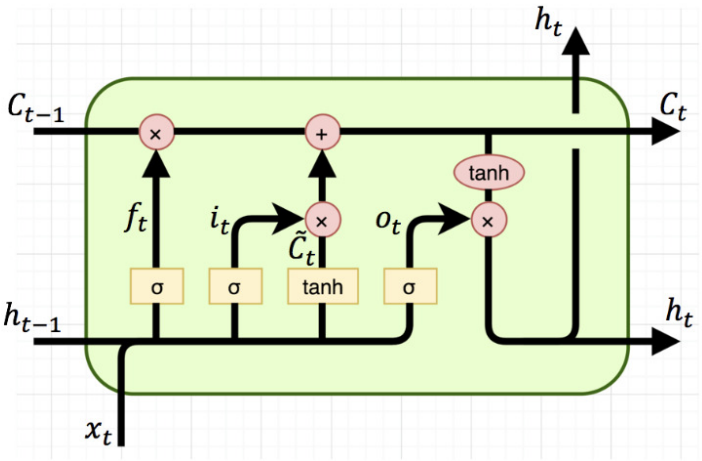

Figure 2.12: LSTM Cell

1. Forget Gate

   This gate determines the extent to which the previous state is to be forgotten by the value of f_t. The value of f_t is calculated by the following equation:

   $$f_t = \sigma(x_t U^f + h_{t-1} W^f) \tag{2.2.15}$$

   Since f_t is an output of a sigmoid function, its value range between 0, and 1. In its extreme, 0 means we forgot everything from the previous cell state. On the other hand, 1 means we don't forget anything i.e. the whole previous cell state is kept.

2. Input Gate

   This gate decides if the new input should be added to the cell state and to which extent it will affect the current cell state. This is controlled by the following equations:

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \tag{2.2.16}$$

$$\tilde{C}_t = tanh(x_t U^g + h_{t-1} W^g) \tag{2.2.17}$$

   Equation 2.3.2 is similar to equation 2.3.1 of the forget gate. However, the value of the sigmoid function is multiplied by the results of equation 2.3.3 which is then added to the cell state. The output of the tanh function in equation 2.3.3 is in the range between 1 and -1. The negative result means that the new input is updating the cell state by reducing some of its components by that negative value.

3. Output Gate

   The final output of the LSTM includes both short and long memory states. The long memory state i.e. new cell state is given by equation 2.3.4. The new cell state is already acquired by passing through the forget and the input gates. The same is stated by the equation.

$$C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t) \tag{2.2.18}$$

   However, for the short memory state i.e. the new hidden state, we need to add some filters to the new cell state to get the new hidden state. It also needs to consider the previous hidden state and the new input. Putting the three together as given by equation 2.3.5. and 2.3.6, we get the new hidden state.

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \tag{2.2.19}$$

$$h_t = tanh(C_t) * o_t \tag{2.2.20}$$

   Both the new cell state and the new hidden state are fed to the next hypothetical cell along with a new input vector.

GRU

Gated Recurrent Unit (GRU) is a relatively recent variation of LSTM that was introduced in 2014 by Cho, Van Merrienboer, Gulcehre, Bahdanau, Bougares, Schwenk, and Bengio in their paper [6]. Similar to LSTM, GRU has the ability to learn long-term dependencies without running into the vanishing gradient problem by maintaining an internal state memory. However, unlike LSTM, GRU has only one hidden memory state for both long and short-term memory. The hidden state is controlled by two gates: the reset gate and the update gate. Figure 2.13 depicts the architecture of the GRU cell which is detailed below.

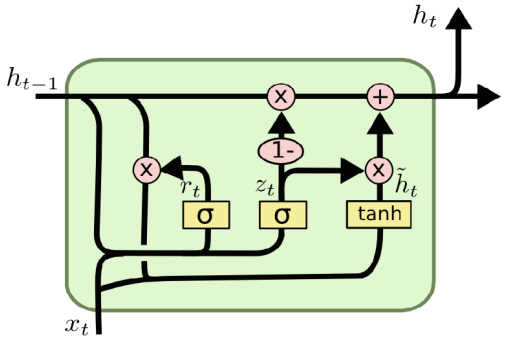

Figure 2.13: GRU Cell

1. Reset Gate

   Similar to the forget gate in LSTM, the reset gate is responsible for deciding how much of the past memory to forget. The reset amount is calculated using equation 2.3.7.

$$r_t = \sigma(x_t U^r + h_{t-1} W^r) \tag{2.2.21}$$

2. Update Gate

   Similar to the input gate in LSTM, the update gate is responsible for deciding how much of past memory to retain given by equation 2.3.8.

$$z_t = \sigma(x_t U^z + h_{t-1} W^z) \tag{2.2.22}$$

The output of the two gates is used to reset/update the previous hidden state to get the new hidden state. Equations 2.3.9, and 2.3.10 define how the new hidden state is

obtained given the previous hidden state, the new input, and the results of the reset and update gates.

$$\tilde{h}_t = tanh(x_t U^g + r_t * h_{t-1} W^g) \tag{2.2.23}$$

$$h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t \tag{2.2.24}$$

For all the equation from 2.3.1 to 2.3.10, we have:

- $U^{(f,i,g,o,r,z)}$ are learned weights for the new input vector x_t and are different for every gate.
- $W^{(f,i,g,o,r,z)}$ are learned weights for the previous hidden state h_{t-1} and are different for every gate.
- σ stand for the sigmoid function.
- * is used for the point-wise multiplication (Hadamard product)

2.3 Related Work

Stock market prediction using machine learning is an extremely active and wide research area. To make this review as thorough as possible, I first present the findings of a systematic review of machine learning-based solutions for stock market prediction. The review covers the research during the period from 2007 to 2018. With this, I was able to narrow down my search to the most recent and relevant works. In the next section, I present 7 of the latest (2018- 2022) and most relevant research works in the area of predicting the stock market with RNN-based models by using technical and sentimental analysis.

2.3.1 Findings of Systematic Review (2007-2018)

Isaac Kofi Nti1, Adebayo Felix Adekoya, and Benjamin Asubam Weyori in [19] have done a systematic review on using fundamental and technical analysis to predict the stock market. The reviews covered around 122 relevant research articles published in

scholarly publications during 11 years (2007–2018) in the field of machine learning-based stock market prediction. The aim is to cluster these works according to the type of analysis used, the number of data sources, and the machine learning model used. The results showed that 66% of the reviewed papers used technical analysis, 23% were based on fundamental analysis and only 11% used combined analyses. Thus, 13 out of 122 have employed a combination of technical and fundamental analysis to predict the stock market with only 3 of them using Twitter data. Regarding the number of data sources used, the review revealed that 89.34% of the reviewed work used single sources; while 8.2% used two data sources and 2.46% used three sources. The most used machine learning algorithms were support vector machine and artificial neural network.

2.3.2 State of the Art Relevant Research Works (2018-2022)

Swathi, T and Kasiviswanath, N and Rao, A Ananda in [25] introduced a Teaching and Learning Based Optimization (TLBO) model with LSTM-based sentiment analysis. The tweets were pre-processed to remove the irrelevant information and transformed into a format that is suitable for the model. Then, the LSTM model was used to categorize tweets into those that express positive and negative attitudes towards stock values. The Adam optimizer was employed to decide the learning rate to enhance the LSTM model's prediction performance. Additionally, the TLBO model was used to adjust the LSTM model's output unit. The TLBO-LSTM model produced a maximum precision of 95.33%, a recall of 85.28%, and an F-score of 90%, with an accuracy of 94.73%.

Zhigang Jin, Yang Yang, and Yuhong Liu in [17] proposed the S_EMDAM_LSTM, or sentiment and empirical mode decomposition-based LSTM with attention mechanism, as a method of predicting stock market closing prices. A CNN-based model was used to calculate the sentiment index of the users' reviews on a certain stock. The sentiment index was calculated based on the number of daily bullish/bearish comments made by the traders. An empirical mode decomposition EMD model was used to extract the trend item from the closing price sequence. The output of the two models was fed to the improved LSTM model to predict the closing price. The LSTM model is improved by integrating an attention mechanism. To test the model, an experiment was conducted on AAPL (stock of Apple inc.) and performed with an accuracy of 70 % and RMSE of 3.2 compared to 60% accuracy and 8.7 RMSE of pure LSTM.

Sreyash Urlam, Bijit Ghosh, and Dr. A. Suresh in [22] used LSTM to predict the closing price of AMZN (Stock of Amazon) using the historical price data from 2000 to 2020. Then, the same model was used to predict the closing price after incorporating the sentiment of tweets. The tweets were collected for the period from 2010 to 2020 and were filtered by the keyword "Amazon" with an average of 20 tweets per day. The polarity of each tweet is calculated and then averaged for the day. The model showed better performance on its first run i.e. without including the sentiment data. It was justified that the latter was considering more factors like people's opinions to predict something volatile like the stock market. It was suggested that ensemble deep-learning techniques might improve the performance of the implemented model.

Marah-Lisanne Thormann, Jan Farchmin, Christoph Weisser, Ren´e-Marcel Kruse, Benjamin Safken, and Alexander Silbersdorff in [21] used LSTM model to predict the closing price of APPL(Stock of Apple inc.) using a combination of historical price data and sentiment analysis of Twitter data. The historical data was gathered for the period from 30th November 2020 to 31st January 2021 to forecast the stock price of 30 minutes and 60 minutes ahead. The tweets were collected for the same period, and were filtered with two keywords "APPL", and "$APPL" using Tweepy API. Sentiment analysis and descriptive statistics were used to extract features from the Twitter data. The former used TextBlob to calculate the polarity and subjectivity of the tweets. The latter is used to calculate other statistical features like the number of tweets, length of tweets, and minimum/ maximum number of tweets per 1 minute for 30 minutes period. The data were fed to the LSTM model which used ADAM optimizer and mean squared error loss. The experiment was conducted on three different kinds of data sets. The first contained the tweets retrieved by the keyword"APPL". The second contained the tweets retrieved by the keyword "$APPL". The third contained both. The results showed the smallest MSE when using the second set of data. They showed also that the incorporation of "$APPL" tweets has enhanced the performance compared to only using "APPL" tweets.

M Sai Revanth, Tarun Madamanchi, M Likith Kumar, M Jeevan Kumar, and Srisailanath in [24] utilized different models (ARIMA, LSTM, and linear regression) to predict the stock price of a user-specified ticker. A combination of historical data and sentiment of tweets were used to make real-time predictions of a stock price. The model was deployed as a web service by providing an online website that enables the users to

choose a ticker and get a real-time prediction of the price.

Xin Huang, Wenbin Zhang, Xuejiao Tang, Mingli Zhang, Jayachander Surbiryala, Vasileios Iosifidis, Zhen Liu, and Ji Zhang in [20] employed an LSTM model to predict the price movement of the cryptocurrency using the sentiment of social media posts. The sentiment analysis was applied to Chinese posts from the most popular Chinese social media application "Sina-Weibo". First, the posts were collected for a period of 8 days with some keywords like "Bitcoin", "ETH" and "XPR" using a web crawler. Then, a crypto-specific sentiment dictionary was created and used to vectorize the data. The processed data were fed to the LSTM model which classifies them as positive or negative. The majority voting on the LSTM sentiment analyzer's output is utilized to determine whether the price will increase or decrease. The experiment was conducted on 7 days of data as training and 1 day for testing. The model performs with a precision of 87% and a recall of 92.5% compared to the auto-regressive model which yields 73.4% precision and 80.2% recall.

Dibakar Raj Pant, Prasanga Neupane, Anuj Poudel, Anup Kumar Pokhrel, and Bishnu Kumar Lama in [15] implemented an RNN model to predict the Bitcoin price based on the sentiment of the tweets and the Bitcoin historical price data. Both the tweets and the price data were gathered for the period from 1st January 2015 to 31st December 2017. First, the tweets were manually classified and labeled as positive and negative. Then, they were pre-proceeded. After that, the tweets were fed to a voting classifier to predict the sentiment of the tweets. The voting classifier takes outputs from five different classifiers and picks the class with the maximum votes. The classifiers are Naïve Bayes, Bernouli Naïve Bayes, Multinomial Naïve Bayes, Linear Support Vector Classifier, and Random Forest. The classifier performed with 81.39% accuracy when evaluated against the actual classes of the tweets. The historical price data along with the sentiment data were sent to an RNN model which forecasts the Bitcoin price with an accuracy of 77.62%

From the above work examples, I noticed that very few studies really combined technical and sentimental analysis. Most of them used row historical price data with Twitter data. Second, the use of the state-of-the-art model in sentiment analysis BERT to calculate the tweets' sentiments with a testing accuracy of 82.88%. None of the mentioned projects have benefited from the power of this model in sentiment analysis. Besides the high-

accuracy prediction of the sentiment, additional data were used to give more meaning to the sentiments like the number of followers and the number of re-tweets in the form of sentimental indicators. By combining some of the powerful technical indicators with sentimental indicators produced by the state-of-the-art sentiment analyzer BERT to predict the future stock price, I contribute to enhancing the prediction.

Chapter 3

Methodology

This chapter answers the question about how to address the problem of forecasting the stock price using a combination of technical and sentimental indicators with RNN-based machine learning algorithms. In this chapter, I briefly describe the overall approach to this problem.

3.1 Overall Approach

The first step is data collection. I collected the data for conducting the technical analysis, sentiment analysis, and BERT training. I used three different datasets collected from three different sources. The datasets are:

1. Historical price data for 100 stocks of Nasdaq index. This dataset was used to conduct the technical analysis. It was sourced from Yahoo Finance.[10]

2. Labeled tweets dataset. The dataset contains tweets, their date, and their sentiment. This dataset was used to train the BERT model and evaluate its performance. It was sourced from Kaggle.[16]

3. Stock tweets dataset. The dataset contains tweets for the Nasdaq-100 stocks, their time, date, and others. This dataset was used to conduct the sentiment analysis. The sentiment of those tweets was calculated by trained-BERT and some other sentiment indicators. It was sourced from data.world.[11]

A detailed description of the datasets will be provided in the next chapter.

Once all the needed data are gathered, pre-processing of the data is required to bring them into a format that is suitable for the machine learning task. For the sake of conducting the technical analysis, technical indicators must be calculated that will be later used as features for predicting the closing price of a stock. Five of the most common technical indicators are calculated i.e. Simple Moving Average (SMA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), and Momentum(MOM) indicators. Technical Analysis Library in python (Ta-Lib) is used to calculate the technical indicators given the historical price data.

Different processing is applied to the tweets datasets to extract their sentiment. First, the labeled tweets are used to train the BERT model and evaluate its performance. Once trained, BERT is used to classify the stock tweets into positive and negative tweets. From the pure sentiment of the tweets, two sentimental indicators are calculated namely averaged weighted sentiment and sentiment moving average along with the tweets' volume, and tweets' volume moving average with a total of four sentimental indicators.

Both technical and sentimental indicators are merged in one dataset with the closing price values as our target. The dataset is transformed into a time series with a look-back period of 60 days to forecast the closing price of the next day.

Recurrent neural network (RNN) models that offer long-short-term memory capabilities are best suited for this kind of time series data forecasting. Two popular examples are Long-short term memory (LSTM), and Gated Recurrent Unit (GRU). The two models are tested on the dataset. The predictions of the two models are compared and the results are analyzed.

Figure 3.1 depicts a flowchart of the proposed scheme design. As shown in the flowchart, it starts with the row datasets. Pre-defined processes/ modules Talib and BERT are applied to get the technical and sentimental indicators, respectively. The indicators are merged into one dataset. The same dataset is fed to two models i.e. LSTM and GRU which predict the closing price of a stock. The outputs of the two models are compared and analyzed.

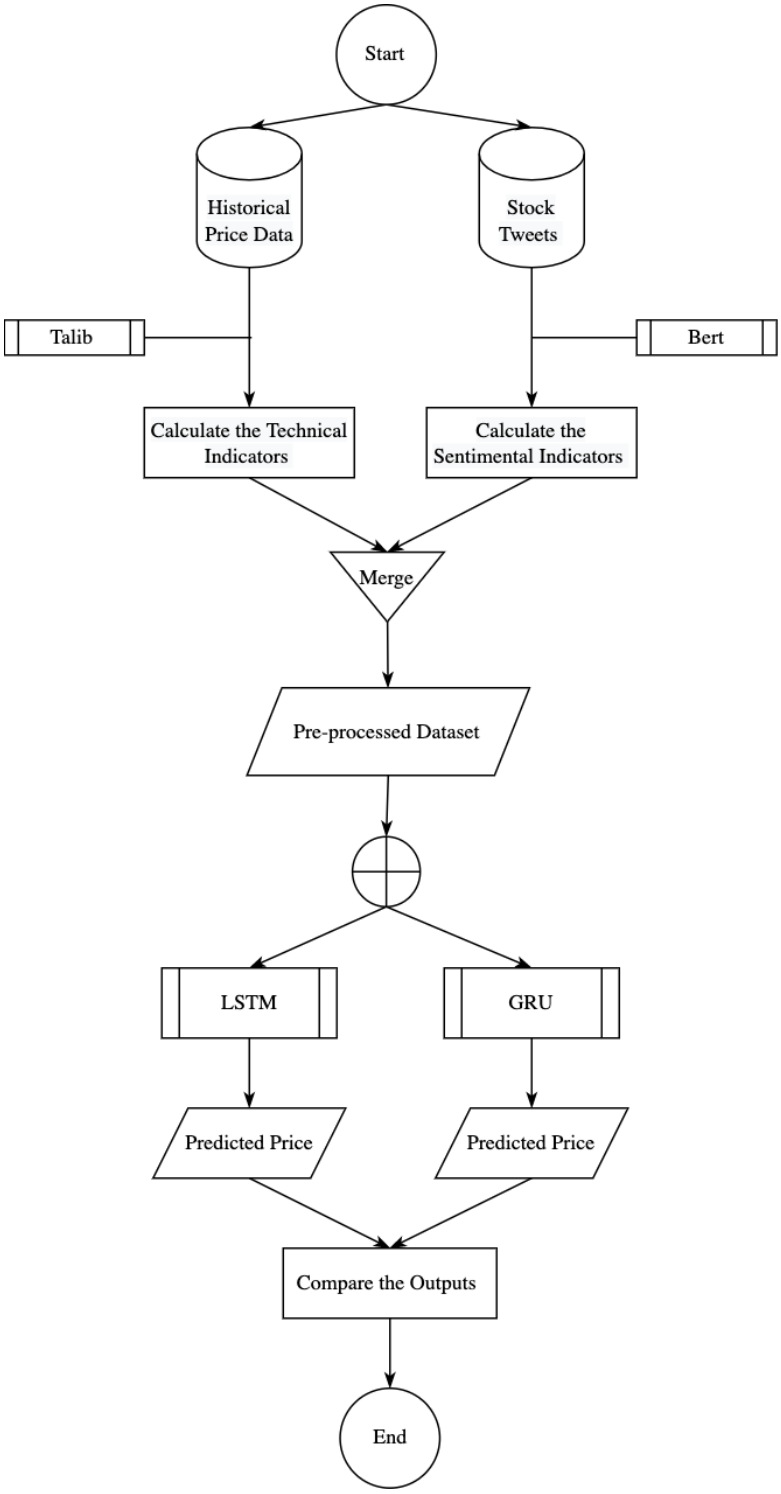

Figure 3.1: Scheme Design Flowchart

Chapter 4

Implementation

4.1 Overview

In this chapter, I thoroughly cover every phase of the solution. I start with a description of the different datasets used in the project and their designated purposes. Then, I explain the pre-processing steps applied to these data in order to get them ready for the machine learning algorithms. I provide justification for my choices when necessary. In the last section, I talk about the choice of the machine learning algorithms used in this project and the model design.

4.2 Data Collection

To predict the daily stock price, technical analysis of the past stock prices and sentimental analysis of the tweets related to the stocks of concern need to be performed. These are two separate tasks and thus, need different kinds of data. To perform the technical analysis, we need the historical price data of stock. On the other hand, sentimental analysis for the same stock is conducted on tweets about that stock. Hence, the stock tweets data is another data needed for this project. In order to find the sentiment of a tweet, a sentiment analysis model has to be trained for this task. With this, labeled tweets data have to be collected to train the model. This study is conducted on 100 stocks of the well-known Nasdaq index. Some examples of companies on the Nasdaq-100 are Apple, Microsoft, Tesla, Amazon, etc... Therefore, the price data and the tweets are collected for the same stocks i.e. Nasdaq-100 stocks.

The three used datasets are described below:

1. Historical Price Dataset.

   This dataset contains the price data for the Nasdaq-100 stocks for the period from 1st Jan 2016 to 15th June 2016. The data is 90 days ahead of the stock tweets, which start 28th March 2016, to allow a look back period for the calculation of the technical indicators without losing any tweets data. This data is collected from Yahoo finance [10] and it contains the date, open, high, low, close, adjusted close, and volume of the price. This dataset is used to calculate the technical indicators.

2. Labeled Tweets Dataset.

   This is a simple dataset sourced from Kaggle [16] that contains some labeled tweets with stock news. This dataset has two columns: the tweet's text and the sentiment of the tweet. The sentiment column can either be 1 for positive tweets or -1 for negative tweets. There are 5791 unique tweets in the dataset with 2,106 negative tweets and 3,685 positive tweets. This dataset is mainly used to test different sentiment analysis tools and models, and to fine-tune the chosen model i.e. BERT.

3. Stock Tweets Dataset.

   This dataset contains unlabeled tweets of 100 stocks of the Nasdaq-100 index. The tweets are collected over 79 days from 28th March 2016 to 15th June 2016. It contains around 1 million tweets. The dataset is sourced from data.world [11] as a zip folder that contains a separate folder for each company. In each company's folder, there are 6 Excel files, one of which is named the "dashboard" which is the one containing the tweets. Besides the tweets' text, the file includes much statistical information about the tweets like the number of retweets, followers, followings, and favorites. This dataset is used for the sentimental analysis part of the project by feeding it to BERT to calculate the sentiment of the tweets along with other sentimental indicators.

   The same data could have been collected manually using the Twitter APIs and I have actually started with this option. However, I abandoned this option for a few reasons:

   (a) Time constraints: Collecting the tweets for about 100 stocks and putting them together takes a relatively long time. In addition, if I am doing this

through the Twitter API, I have to request academic research access which also takes a few days.

(b) Same results: Using either manually collected tweets, or the Nasdaq-100 tweets, is not making a significant change in the results of this project, So, I better devote this time to something else that will contribute to the quality of the study.

(c) Quality of the Nasdaq-100: The Nasdaq-100 datasets contain a good number of tweets about all 100 companies during the same period. Constructing such balanced data might be hard, especially for small companies with less number of tweets.

4.3 Data Pre-Processing

The use of multiple datasets serving different purposes resulted in separate pre-processing for each dataset to suit the designated purpose. Table 4.1 lists all the datasets and their uses in the project. The following subsections are use-wise divided. Each subsection describes the applied pre-processing steps and justifies performing such steps.

Dataset	**Uses**
Historical Price Dataset	Calculate the technical indicators to be used for stock price prediction.
Labeled Tweets Dataset	Evaluate different sentiment analysis models to help find the best model. Train the chosen model.
Stock Tweets Dataset	Calculate the sentimental indicators to be used for stock price prediction.

Table 4.1: Uses of Different Datasets

4.3.1 Technical Indicators

Traders use different combinations of technical indicators to predict future stock prices. Therefore, the first step is to decide on the combination that offers the best price prediction. This could be accomplished through the use of feature extraction algorithms. However, after reviewing the literature for similar studies, I found that, among a large

number of technical indicators, only a few (five to six) indicators were used by most of the studies. Those common indicators are mostly used because of their effectiveness in predicting stock prices. Thus, using a features extraction algorithm to come up with my own combination of technical indicators is a redundant task. Therefore, I chose the most common indicators used in the literature [19]. In addition, to enhance the price prediction, I use a combination of lagging and leading indicators. The technical indicators used in this project are:

1. SMA (lagging indicator)
2. EMA (lagging indicator)
3. MACD(lagging indicator)
4. RSI (leading indicator)
5. Momentum (leading indicator)

To calculate the selected indicators, historical price data is required. Thus, the historical price dataset is downloaded. Then, the technical indicators are calculated using technical analysis library in python called Ta-Lib. Finally, the excess columns are removed i.e. all the original columns except the closing price column. At the end of this process, we have the five technical indicators along with the closing price for the given range of tweets.

4.3.2 Choice of Sentiment Analysis Model

Now that we have the technical indicators, it is time to calculate the sentimental indicators. The calculation of sentimental indicators is based on the sentiment score of the tweets. The sentiment analysis model is used to calculate such scores. Different models are available (discussed in section 2.2.2) and need to be evaluated for the sake of choosing one for our project. The following models are evaluated using the labeled tweets dataset:

1. TextBlob
2. VADER

3. Naïve Bayes

4. BERT

The four models are used to predict the sentiment of the tweets in the labeled tweets dataset. The predicted sentiment is evaluated against the actual sentiment to find the accuracy of the models. Table 4.2 lists the accuracy of the four models.

Model	**Accuracy**
TextBlob	62.67 %
VADER	66.48 %
Naïve Bayes	66.78 %
BERT	82.88 %

Table 4.2: Evaluation of Sentiment Analysis Models

As seen from the table, BERT has outperformed the other models by about 6%. Based on this experiment, BERT was chosen to be the sentiment analysis model for this project.

4.3.3 Training of Sentiment Analysis Model

In the previous section, I talked about evaluating the different sentiment analysis models. The rule-based models i.e. VADER and TextBlob can be directly evaluated without the need for further training. However, in the case of machine learning models such as Naïve Bayes and BERT, further training (fine-tuning) is required before evaluating them. Since we are only concerned with BERT in this project. I describe the training of BERT here in this section. The Naive Bayes training is provided with the full implementation but not explained in this report.

As mentioned before, the labeled tweets dataset is used to train and evaluate BERT. The first step is to clean the text of the tweets by removing all the hyperlinks, symbols, emojis, extra spaces, and ending punctuation. Once the text of the tweets is clean, the data is split into training and testing with an 80/20 split ratio. Cleaning the text is not enough to feed the data to BERT, the text needs to be encoded for BERT. This is done using the BERT tokenizer. Now that we have the tweets encoded and ready, we need to prepare the model itself.

BERT model is already trained on large data as we explained earlier in this report. However, we need to add some extra layers to it and fine-tune its hyperparameters. I added three layers: input, hidden, and output layers. Since I use the $BERT_{base}$ model with a hidden size of 768, my input layer is the same size. For the hidden layer size, I choose 50 by trial and error. The output size is 2 for the two classes i.e. positive and negative. I use Adam optimizer and cross-entropy loss. I train the model on 4 epochs with patch size 16. The patch size is dependent on the GPU used and available memory. So, batch size of 16 was the highest I could afford. After training, the model is evaluated against the actual labels. The model performed with a final test accuracy of 82.88%.

4.3.4 Sentimental Indicators

Twitter contains valuable data that if analyzed well, could give good insights into the stock market movement. Sentiment analysis of Twitter data for stock price prediction is the heart of this project. Thus, we aim to best utilize these data.

The sentiment analysis model chosen for this task is BERT as seen in the previous sections. After being trained and evaluated, BERT is ready to receive unlabelled stock tweets to predict their sentiment i.e. positive or negative. However, to satisfy our goal of maximizing the benefits of the Twitter data, we went beyond sentiment classification to sentiment analysis and descriptive statistics.

Based on the pure sentiment score, two sentimental indicators are calculated:

1. **Averaged Weighted Sentiment**

   Two tweets of the same sentiment might have a different impact on the stock price. The number of followers of the person tweeting, and the number of retweets contributes to the significance of the tweet. Thus, it is important to assign different weights to the tweets. The weights are based on the number of followers and the number of retweets. One more thing to consider is the publicity of certain stocks over others. For example, apple stock might get more tweets compared to less popular stock. This should not affect the overall weight of the daily tweets about a certain stock. To tackle this, the mean and the standard deviation of the followers of people tweeting about certain stocks are calculated. The same is calculated for

the number of retweets. Then, the weights are computed based on how far these values i.e. followers, and the retweets from the mean and the standard deviation.

The weights are calculated according to the following algorithm:

- For all tweets, initialize weight to 1
- Calculate the rolling mean of the number of followers
- Calculate the rolling standard deviation of the number of followers
- For all tweets,
  - if the number of followers >= mean but < standard deviation
  - then, weight +=1
  - else if number of followers >= standard deviation but < standard deviation* 2
  - then, weight +=2
  - else if the number of followers >= standard deviation*2
  - then, weight +=3
- Calculate the rolling mean of the number of retweets
- Calculate the rolling standard deviation of the number of retweets
- For all tweets,
  - if the number of retweets >= mean but < standard deviation
  - then, weight +=1
  - else if the number of retweets >= standard deviation but < standard deviation* 2
  - then, weight +=2
  - else if the number of retweets >= standard deviation*2
  - then, weight +=3

The algorithm increases the weight by 1 if the number of followers/retweets is between the mean and the first standard deviation. It increases the weight by 2 if the number of followers/retweets is between the first and the second standard deviation. And by 3 if the number of followers/retweets is greater than the second standard deviation.

The weight of each tweet is then multiplied by its sentiment to give the weighted sentiment of the tweet. Since I am interested in day-wise indicators, the weighted sentiment is added for the day and divided by the number of tweets for that day to produce the averaged weighted sentiment indicator.

2. **Sentiment Moving Average**
   This indicator is tracking the sentiment movement by calculating the moving average of the averaged weighted sentiment. Thus, we have an overall idea of the sentiment trend.

Besides the sentiment analysis, descriptive statistics are used to extract some features from Twitter data. The following are descriptive indicators :

1. **Tweets volume**
   It is the total number of tweets about a certain stock per day.

2. **Tweets Moving Average**
   This is the moving average of the volume of the tweets. This indicator tells us about the direction of the tweet traffic about a certain stock and whether it is increasing, decreasing, or still.

4.3.5 Merge the Technical and Sentimental Indicators

The last pre-processing step is to merge the technical and the sentimental indicators based on the date. Since technical indicators are missing for the weekend and holidays, rows with missing values are dropped. Now that I have an indicator-wise merged dataset for each stock, I then merge the datasets for all stocks in one final dataset. The final dataset has 2255 data instances and the following columns:

1. Closing price (target)
2. Averaged Weighted Sentiment
3. Sentiment Moving Average
4. Tweets volume
5. Tweets Moving Average

6. SMA
7. EMA
8. MACD
9. RSI
10. Momentum

The dataset is then split into training and testing based on the date. All the instances before 2016-06-01 are training data and all the instances after this date belong to the test set.

The training and the testing data are scaled to the range(0, 1) as part of preparing them for the machine learning model. The dataset is also split into dependent and independent variables i.e. x, and y.

The machine learning models used in this project expect the input to be in the 3-dimensional form [no.of samples, on. of time steps, no. of features]. Therefore, the shape of the x-train and x-test is checked before sending them to the model. The data were already in 3D shape.

4.4 Model Design

This project is all about predicting the stock price using machine learning algorithms. In this section, I justify the choice of the machine learning models used in this project. I also describe the model design, training, and optimization.

4.4.1 Choice of the Models

The prediction of a future stock price is dependent on past stock data. Thus, among all of the machine learning algorithms, I am interested in algorithms with memory capabilities. A board category of this type is Recurrent Neural Network RNN. RNN models are the best option for handling data sequences for their ability to capture short-term dependencies. However, in the case of long sequences, they suffer from vanishing gradients problem [4]. Two advanced versions of RNN are LSTM and GRU (explained

in chapter 2) can handle long sequences without running into the vanishing gradient problem by the means of selective memory. These models can remember relatively important information, and forget unnecessary details with the help of some gates.

In this project, I predict the stock price of the next day using the data of the past 60 days. With such a long look-back period, LSTM and GRU are chosen to handle the long sequence effectively. Since the two models have lots of similarities, it was theoretically hard to find the best among them. Therefore, I considered testing both of them in this project and comparing their performance.

4.4.2 Design of the Models

The process of building a machine learning model pass through different stages with many decisions involved in each stage. From building the model layers to compiling, and fitting the model. Some decisions are based on facts, and some are based on trial and error. As mentioned in the previous section, two machine learning models are used in this project. However, for the sake of comparison, the model design is identical for both of them i.e. both have the same layers and same hyper-parameters. Figure 4.1 show the overall structure of the model layers.

The model receives input of shape [60,10] for the time steps and the number of features, respectively. The model is constructed by stacking LSTM/GRU layers followed by Dropout layers. The number of LSTM/GRU layers is decided by trying different numbers of layers. Few layers like one, or two, produce loose predictions when compared to the actual prices. Five layers and above captures unnecessary short-term swings of the price i.e. too detailed predictions. However, three to four layers give reasonable results.

Naturally, the LSTM/GRU layer is followed by an activation function layer. Two candidates' functions are: 'relu', and 'tanh'. 'relu' is the most used activation function for hidden layers in neural networks. However, in the case of LSTM/GRU hidden layers, 'tanh' is the default function. Besides that, the model fits much faster with 'tanh' and gives more accurate results. Therefore, 'tanh' was fixed for this model.

Without any dropout layers, the model performs too well that I suspect an overfitting situation. A secure option is to add some Dropout layers to regulate the learning process

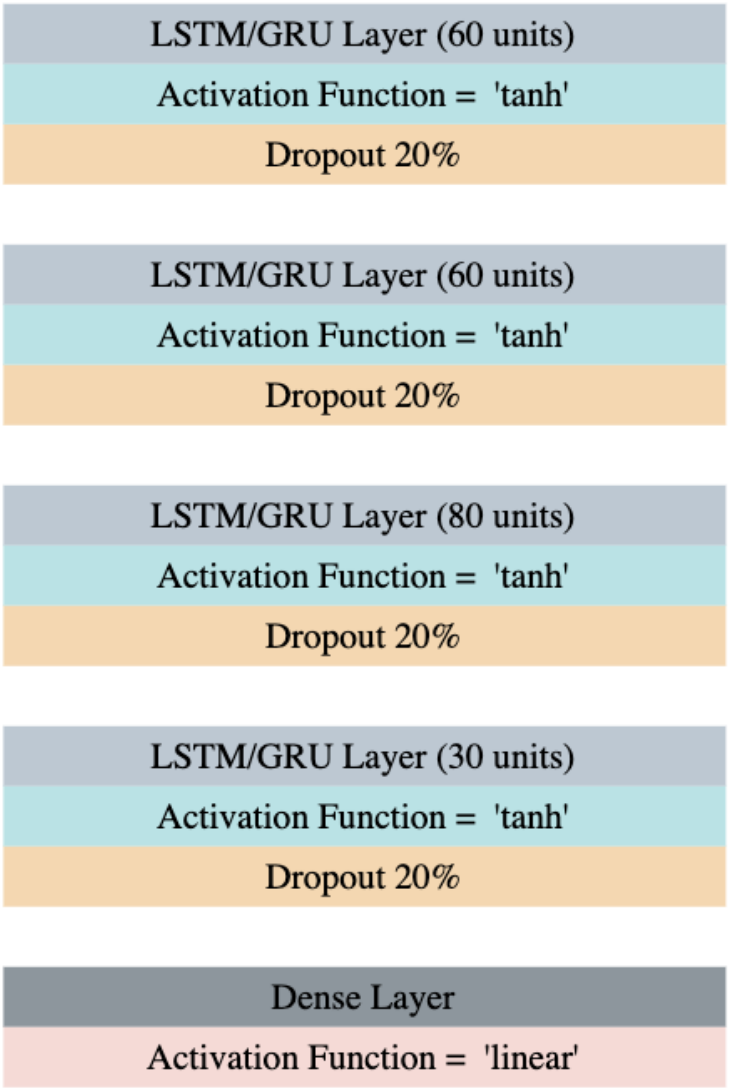

Figure 4.1: Model Design

and avoid overfitting. A dropout of 20 % is enough to help us avoid overfitting. A higher dropout percentage affects the model performance negatively.

The number of units is a matter of trial and error. However, the first layer is chosen to be 60 to meet the input size although it is not so relevant but acts as an initial value to build on.

The last layer of the model should be an output layer (Dense layer). Since I am building a regressor, 1 unit is required. The default activation function is used which is the 'linear' activation function because I have continuous output values.

Once the model is constructed, the training process starts. Optimizers play a vital role in this process by optimizing the weights of the network iteratively. Adam optimizer is surpassing the traditional stochastic gradient decent optimizer with its ability to adaptively change the learning rate [9]. Thus, the Adam optimizer is a safe option to consider for my model.

The optimizer's job is to improve the performance of the model. To keep the optimizer informed about the model performance, the Loss function is used. Since I am dealing with a regression problem, 'mean_squared_error' is the used loss function.

To speed up the learning process, the mini patch technique is used with a batch size of 32. The model starts to converge after about 20 epochs. Thus, the model is trained on 20 epochs.

Chapter 5

Results and Discussion

5.1 Overview

While approaching the problem of predicting the stock price using a combination of technical and sentimental indicators, candidates' solutions were considered and different options were evaluated. However, these evaluations helped construct the final design of the machine learning pipeline for forecasting the stock price. In this chapter, I evaluate the performance of this design under different conditions with two different machine-learning models.

As mentioned in the previous chapter, the same model design is used to compare two different RNN models i.e LSTM and GRU. Therefore, the performance of the two models is evaluated and compared in this chapter.

To highlight the significance of Twitter data on the stock price, the two models are tested with two sets of indicators. The first set contains only the technical indicators. In the second test, the sentimental indicators are added to the technical indicators. This is done to prove our claim that sentiment analysis of stock-related data on Twitter has a relative impact on the stock price.

This leaves us with four different experiments each with different results. The following sections present and discuss the results of each experiment. The final section of this chapter provides an interpretation of the overall outcomes of these experiments altogether.

5.2 Experiment 1 LSTM with Technical Indicators

In this experiment, the LSTM model is used to predict the next day's closing price with 60 days look-back period given the five technical indicators:

1. SMA with a period of 10 days.
2. EMA with a period of 30 days.
3. MACD with 26 days, and 12 days for slow and fast periods respectively.
4. RSI with a default period of 14 days.
5. Momentum with 10 days period.

The model followed the design and training explained in the previous chapter.

This experiment provides the basis for this evaluation and improvement cycle. The next experiments are built upon this experiment for the sake of achieving better results. They also follow the same settings for the prediction horizon, the look-back period, and the indicators' periods.

Before providing any numerical results, it is worth mentioning that the models produce slightly different results on each run. This is due to the randomness in the training process that might be attributed to the dropout technique, and other factors. However, the relative results of the different experiments followed an almost consistent pattern i.e. the results of experiment x are consistently better than experiment y regardless of the error magnitude. Thus, the numbers provided are an example run of the models to illustrate the overall behavior of the results.

The LSTM model performed with a root mean squared error (RMSE) of 13.83. The interpretation of this number will become clearer as we progress with the other experiment. Nevertheless, it was a good point to start with. Figure 5.1 visuals the LSTM prediction of the stock price aligned with the actual prices.

From this figure, one can clearly see how the LSTM model copied the general movement of the prices and trends. However, the main weakness here is that the LSTM prediction

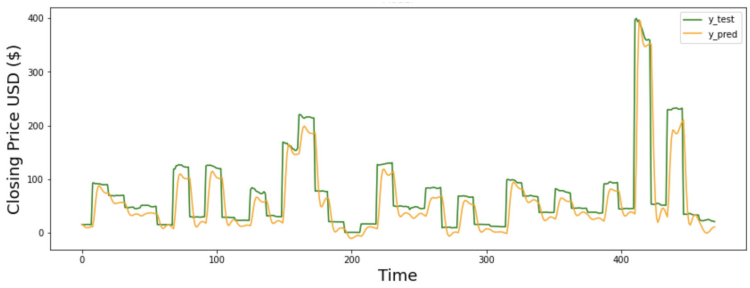

Figure 5.1: LSTM Price Prediction using Technical Indicators

is so lagging behind despite the use of two leading indicators. Another observation is that the predicted price is always under the actual price i.e.the LSTM model is underestimating the stock price. An explainable machine learning tool could be used to explain such shortcomings, however, it is recommended for future works.

5.3 Experiment 2 LSTM with Technical and Sentimental Indicators

This experiment reveals very crucial information about the significance of incorporating Twitter data in the form of sentimental indicators for stock price prediction. This experiment is based on the first experiment with the addition of sentimental indicators. Therefore, the LSTM model in this experiment predicts the stock price provided both technical and sentimental indicators. The technical indicators are the same as in the first experiment. The sentimental indicators used are:

1. Averaged Weighted Sentiment.
2. Sentiment Moving Average.
3. Tweets Volume.
4. Tweets Moving Average.

With the addition of these indicators, the performance of the LSTM model dramatically improved by more than 70 % with an RMSE of 3.96. This was an expected result as the incorporation of such valuable information should intuitively enhance the prediction results. Figure 5.2 show the LSTM model prediction after incorporating the sentimental indicators.

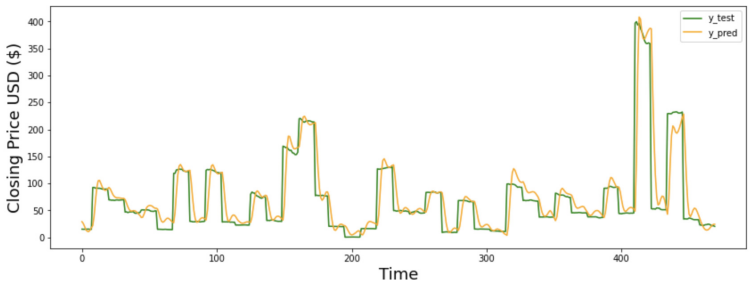

Figure 5.2: LSTM Price Prediction using Technical and Sentimental Indicators

The figure shows that the use of combined indicators could relatively reduce both the horizontal and vertical gaps between the predicted price and the actual price. The LSTM model in the first experiment was crawling behind the actual price movement. The addition of sentimental indicators gives the models some insights about the future movement of the price which in turn reduced the vertical gap.

5.4 Experiment 3 GRU with Technical Indicators

Based on the findings of [5], the performance of LSTM and GRU is comparable in the field of machine translation. The empirical evaluation of the two models in [7] could not conclude which one is better in the field of sequence modeling.

In this study, we compare the performance of LSTM and GRU in the field of stock price prediction. In the previous experiments, LSTM was evaluated with two different sets of indicators to predict the next day's stock price. In this experiment, and the next one, GRU is tested with the same sets of indicators. This allows us to observe the performance of the two models and arrive at some conclusion.

In this experiment, GRU is used to predict the stock price with a set of technical indicators only. This experiment has the same settings as experiment 1 except that LSTM is replaced with GRU. In the first experiment, LSTM performed with an RMSE of 13.83. Surprisingly, GRU outperformed LSTM with an RMSE of 2.01. This result was unexpected as the two models theoretically look the same.

As mentioned in section 5.2, the RMSE changes with different runs of the models. However, 99% of the time, GRU achieved better performance. Sometimes, the error

gap between the two models is less than the shown results with the superiority of GRU over LSTM. Figure 5.3 illustrate the outstanding performance of GRU in predicting the stock price with a set of technical indicators.

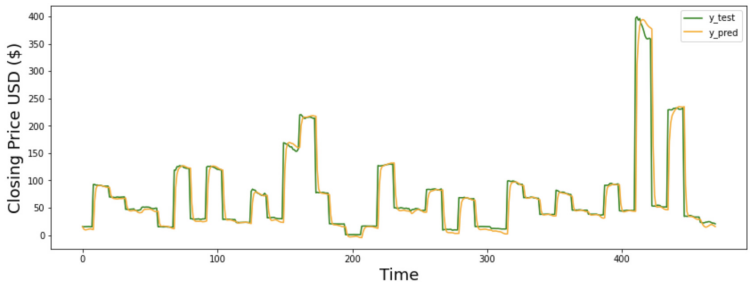

Figure 5.3: GRU Price Prediction using Technical Indicators

The figure shows how GRU almost copied the actual price movement with only a slight delay. This delayed prediction mostly occurred at the trend reversal points. This is because the used dataset is composed of multiple stocks, each with different behavior. In addition, the number of data instances for each stock is low. This affects the learning process of the model.

5.5 Experiment 4 GRU with Technical and Sentimental Indicators

Similar to experiment 2, in this experiment, GRU is used to predict the stock price given the combined set of indicators i.e. technical and sentimental. In experiment 2, we have seen the great impact of the sentimental indicators in predicting the stock price with LSTM.

With GRU, we have already achieved good performance with the set of technical indicators in the previous experiment. This leaves us with a small window for improvement. Yet, the use of a combined set of indicators did improve the performance with an RMSE of 0.33. Figure 5.4 depicts the price prediction of the GRU with combined indicators. The improvements might not be as visible as the previous ones but the error values proved that the overall performance of the model improved with the addition of the sentimental indicators.

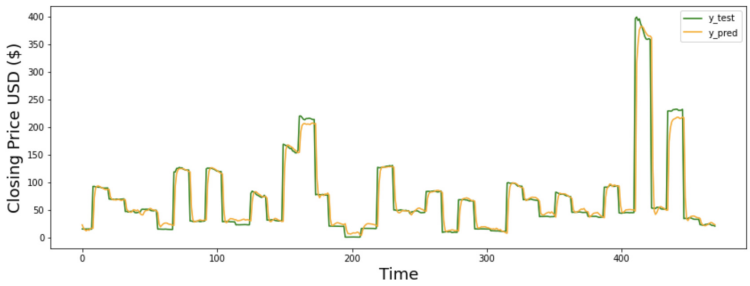

Figure 5.4: GRU Price Prediction using Technical and Sentimental Indicators

5.6 Main Findings

In this chapter, I evaluated the performance of the two RNN models namely LSTM and GRU in forecasting the stock price with two sets of indicators. One set includes only the technical indicators. The other set contains a combination of technical and sentimental indicators. Table 5.1 summarises the RMSE of the four experiments. The values of the RMSE change every time we run the experiments. The values in the table are example results that represent the overall outcomes of the experiments.

Indicators	**LSTM**	**GRU**
Technical	13.83	2.01
Combined	3.96	0.33

Table 5.1: Evaluation of Sentiment Analysis Models

Regardless of the exact error values, this study arrived at two main findings:

1. Although they have comparable performance in many machine learning fields, GRU outperformed LSTM in the field of stock market prediction.

2. The sentiment analysis of stock-related tweets gives better insights into future stock price movement.

Chapter 6

Conclusion

When acting in an uncertain environment such as the stock market, all strategies and techniques boil down to minimizing the risk and maximizing the benefits. An accurate prediction of the stock price help in making a profitable deal, and confirming a trading decision. This study combines the two schools of stock analysis, technical and fundamental analysis, to predict the stock price using machine learning-based solutions. In this study, five of the most effective technical indicators are used to predict the stock price.

The fundamental analysis involves a sentiment analysis of Twitter data to identify its impact on stock prices. While testing different models for sentiment analysis, the sentiment prediction accuracy is raised from 62.67% to 82.88%, which is achieved by BERT. The pure sentiment scores are mined for additional information. Two sentimental indicators are extracted i.e. Averaged Weighted Sentiment, and Sentiment Moving Average to provide a bigger picture of the sentiment trends and movement. Descriptive indicators i.e. Tweets Volume, and Tweets Moving Average are also used to measure how the activity on certain stock affects its price.

The technical and the sentimental data are then fed to the machine learning models i.e. LSTM, and GRU to predict the stock price. During the experiment, the RMSE is reduced from 13.83 to 0.33. The first RMSE results from using LSTM with only technical indicators. The second RMSE results from using GRU with a set of technical and sentimental indicators. These numbers show how the sentimental analysis contributes to better stock price prediction. This study also demonstrates the superiority of GRU

over LSTM in the field of stock market prediction.

6.1 Limitations

This project went through three main stages, technical analysis, sentimental analysis, and the development of the machine learning model. I faced difficulties and challenges in each of the stages. I was able to overcome some, but I also decided to give up others under time constraints.

During the technical analysis stage, the plan was to calculate up to 40 technical indicators. Then, I was supposed to minimize the number to 20 indicators based on my literature review. Finally, I had to apply a features selection algorithm to select the most five/six relevant indicators. This plan provides a more informed selection of the technical indicators. However, due to time constraints, I had to sacrifice the usage of the features selection algorithm. Thus, the indicators were chosen based on the literature review.

While collecting the Twitter data for the sentimental analysis, I had two options: use the Nasdaq-100 dataset, or conduct a manual collection of stock-related tweets. In the beginning, I was leaning toward the second option as it will provide as much data as needed to train the machine learning model. However, for the reasons mentioned at the end of section 4.2, I abandoned this option and went for the Nasdaq-100 dataset.

In many fields, the performance of LSTM and GRU is comparable. The same was expected in the field of stock market prediction. However, the results showed the superiority of GRU over LSTM with no convincing explanation provided. To interpret these results, I could implement an explainable machine learning model. However, the time was against me, so I had to accept the results as it is.

6.2 Future Works

In the limitations section, I have mentioned some ideas that could not be implemented due to time constraints. This opens a gate for improvement in the future.

To make an informed and confident selection of technical indicators, one may use a feature selection algorithm to find the most relevant indicators. In addition, more data over a long period may be collected to provide better training for the machine learning algorithms.

In this project, the next day's stock price is predicted. I would recommend predicting the price over a longer time horizon such as a week ahead. Explainable machine learning tools such as LIME, or SHAP are highly recommended for more understanding of the results and the model behavior.

One last recommendation I would like to make is to do a more thorough review of the literature in the field of stock market prediction to find more profound issues to address.

6.3 Reflection

My master's dissertation is an experience full of lessons on the personal and professional aspects. Roughly saying, 70% of the knowledge and skills reflected in my work are gained through this experience. This project motivates me to go beyond computer science and learn more about the field of the stock market. I have learned many trading strategies that I didn't mention in the report while I was reading about the stock market analysis. I became interested in the field, especially trading with technical analysis. This motivates me to take professional training in the future to start trading in the real market.

Besides the knowledge I gained in the problem domain, I realized, for the first time, how scientific research takes place. I came to know that there is no one right method for approaching the problem. One should keep exploring new options and improving the current solution until finding a satisfactory solution. It was not as systematic as I thought. The beginning of this journey involves lots of struggles and challenges. Once the path to the solution is mapped, the journey becomes smoother and more entertaining.

One thing that I have overlooked is the writing of the report, I thought it is just a matter of documenting my code. Thus, I assigned a short period for it. I first discovered this when my supervisor notified me that I should assign more time for report writing. I

realized how profound and ruled the process was when I had to cite my references, justify my choices, link my ideas, and so on. This dissertation taught me how significant is academic writing without which all the findings and discoveries could have been lost or even attributed to the wrong person.

On the personal side, I discovered a few strengths and weaknesses. This dissertation helped me appreciate my communication skills. When writing the report, I did not face any difficulties in explaining my ideas. Although I was a bit reluctant in decision-making, once the decision is made I was good at execution. Two things I realized I have to work on after this dissertation. The first one is time management. I should have a balance between the quality of the project and the time limit. Another skill that I missed is how to do a systematic literature review. I tried to learn while doing my dissertation yet I believe I should dedicate more time to learning how to do a literature review in the right way.

Finally, I experienced what Joe Abercrombie said: "The more you learn, the more you realize how little you know. Still, the struggle itself is worthwhile. Knowledge is the root of power, after all."

Bibliography

[1] Fama EF. "Random walks in stock market prices." In: *Financ Anal J 21:55–59* (1965).

[2] Burton Gordon Malkiel. *A random walk down Wall Street: including a life-cycle guide to personal investing*. WW Norton & Company, 1999.

[3] Suresh AS. "A study on fundamental and technical analysis". In: *International Journal of Marketing, Financial Services & Management Research* 2.5 (2013), pp. 44–59.

[4] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *International conference on machine learning*. PMLR. 2013, pp. 1310–1318.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[6] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[7] Junyoung Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[8] C. Hutto and Eric Gilbert. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text". In: 8 (May 2014), pp. 216–225. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[9] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[10] Yahoo Finance. *100 Nasdaq Data*. https://uk.finance.yahoo.com/. 2016.

[11] Enrique Rivera. *Nasdaq 100 Tweets*. https://data.world/kike/nasdaq-100-tweets. 2016.

[12] Pio Calderon. *VADER Sentiment Analysis Explained*. shorturl.at/cgsW8. 2017.

[13] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[14] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[15] Dibakar Raj Pant et al. "Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis". In: *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*. 2018, pp. 128–132. DOI: 10.1109/CCCS.2018.8586824.

[16] YASH CHAUDHARY. *Stock-Market Sentiment Dataset*. https://www.kaggle.com/datasets/yash612/stockmarket-sentiment-dataset. 2020.

[17] Zhigang Jin, Yang Yang, and Yuhong Liu. "Stock closing price prediction based on sentiment analysis and LSTM". In: *Neural Computing and Applications* 32.13 (2020), pp. 9713–9729.

[18] Steven Loria. *Authers*. https://textblob.readthedocs.io/en/dev/authors.html. 2020.

[19] Isaac Kofi Nti, Adebayo Felix Adekoya, and Benjamin Asubam Weyori. "A systematic review of fundamental and technical analysis of stock market predictions". In: *Artificial Intelligence Review* 53.4 (2020), pp. 3007–3057.

[20] Xin Huang et al. "LSTM Based Sentiment Analysis for Cryptocurrency Prediction". In: *Database Systems for Advanced Applications*. Ed. by Christian S. Jensen et al. Cham: Springer International Publishing, 2021, pp. 617–621. ISBN: 978-3-030-73200-4.

[21] Marah-Lisanne Thormann et al. "Stock Price Predictions with LSTM Neural Networks and Twitter Sentiment". In: *Statistics, Optimization amp; Information Computing* 9.2 (May 2021), pp. 268–287. DOI: 10.19139/soic-2310-5070-1202. URL: http://www.iapress.org/index.php/soic/article/view/1202.

[22] Sreyash Urlam et al. “Stock Market Prediction Using LSTM and Sentiment Analysis”. In: *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.11 (2021), pp. 4653–4658.

[23] Any chart Documentation. *Technical Indicators Mathematical Description*. https://docs.anychart.com/Stock_Charts/Technical_Indicators/Mathematical_Descriptions. 2022.

[24] M Sai Revanth, Tarun Madamanchi, and M Likith Kumar. “STOCK FORECASTER”. In: (2022).

[25] T Swathi, N Kasiviswanath, and A Ananda Rao. “An optimal deep learning-based LSTM for stock price prediction using twitter sentiment analysis”. In: *Applied Intelligence* (2022), pp. 1–14.

[26] Naveen B. Kumar. *The use of technical and fundamental analysis in the stock market in emerging and developed economies*. eng. First edition. Bingley, England: Emerald, 2015 - 2015. ISBN: 1-78560-404-X.