News Sentiment Effect on Financial Market Trends

Andrew Darmanin

Supervisor: Dr Joel Azzopardi

June 2023

Submitted in partial fulfilment of the requirements
for the degree of B.Sc. IT (Hons.) Artificial Intelligence.

L-Università ta' Malta
Faculty of Information &
Communication Technology

Abstract

This project aims to investigate the impact of news articles on financial market trends, such as the Standard & Poor's 500 index. The main hypothesis of this project is that news articles contain specific characteristics, such as sentiment and events, that can be leveraged to predict the direction of the stock market. The ultimate goal is to develop a Machine Learning (ML) model that can effectively use news article features, alongside other stock market data if needed, to forecast market trends.

Market trend is defined as the tendency of financial markets to move in a particular direction over a given time frame. The direction is determined by a number of variables, such as events, speculation, supply and demand and government policy. Predicting these variables reliably is difficult. However in reality the market is made up of many investors and traders who are willing to buy or sell shares of a specific stock. The common phenomenon here is that investors and traders use news sources to try to predict where the stock price will be in the future. After this evaluation they place their order accordingly. The premise here is that certain events are expected to lead to a positive outlook – hence attracting more buyers – while other events lead to a negative outlook leading to an increase in selling. This change in supply and demand results in a change in the trend of the markets' value.

For individual investors, reading and analysing daily news articles is a time consuming and tedious task. However, Artificial Intelligence (AI) systems can observe patterns over historical data and generalise efficiently. In this project different ML models are developed to analyse the movement of the stock market in relation to daily news articles and forecast the direction for the immediate movements. The business-related articles are obtained using the New York Times (NYT) Application Programming Interface (API). Textual features from news reports are extracted and weighted using different Natural Language Processing (NLP) methods. In essence, these methods should result in an accurate and machine-readable representation of the textual features. The ML models are trained on these features and stock market data to predict the direction, 'Upwards' or 'Downwards', of a given stock market equity for a given time window, primarily from market open till market close.

The possibility of predicting stock market prices has been a subject of debate among researchers, with some stating that it is infeasible due to the market's volatility and unpredictability. However, the rise of AI solutions has led many to challenge this claim, and this project's results demonstrate the potential of using news articles as features to forecast the direction of the stock market over a long period. Moreover, the results offer valuable insights into the factors that influence the stock market. All in all, this research showcases the use of AI in forecasting financial market trends.

Acknowledgements

I would like to express my sincere gratitude to Dr. Joel Azzopardi for his constant guidance and support throughout this final year project. His expertise and insights have been crucial in shaping the project. Additionally, this project would not have been possible without the invaluable contributions of my family and friends, whose support and motivation have been instrumental in overcoming challenges and staying focused.

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence.

ANN Artificial Neural Network.

API Application Programming Interface.

BoW Bag of Words.

CNN Convolutional Neural Network.

EMA Exponential Moving Average.

EMH Efficient Market Hypothesis.

ETFs Exchange-traded funds.

GBM Gradient Boosting Machines.

GloVe Global Vectors for Word Representation.

GRU Gated Recurrent Unit.

LR Logistic Regression.

LSTM Long Short-Term Memory.

MACD Moving Average Convergence/Divergence.

MFI Money Flow Index.

ML Machine Learning.

NER Named Entity Recognition.

NLP Natural Language Processing.

NLTK Natural Language Toolkit.

NYT New York Times.

POS Part-of-speech.

RBF Radial Basis Function.

RF Random Forest Classifier.

RNN Recurrent Neural Network.

RSI Relative Strength Index.

SVM Support Vector Machine.

TF–IDF Term Frequency - Inverse Document Frequency.

US United States.

VADER Valence Aware Dictionary for Sentiment Reasoning.

1 Introduction

The stock market consists of several exchanges in which traders and investors buy and sell shares of publicly traded companies. Shares represent fractional ownership of equity in an organisation. Changes in the stock price are triggered by many factors, such as events, economic data, speculation and sentiment. Reliably predicting the future price of a stock has been an interest of many researchers and investors as this would yield significant advantages.

In the investment world there are two main hypotheses that seek to explain the change in stock prices. These are the Efficient Market Hypothesis (EMH) [1] and the Random Walk Theory [2]. In essence, EMH states that share prices reflect all available information. On the other hand, Random Walk hypothesis state that stock prices do not follow trends but move seemingly at random. Fundamentally both hypotheses claim that stock market prediction is infeasible. However, many researchers challenged these hypotheses and managed to predict the future trend of stock markets with noticeable accuracy [3–10]. The basic premise of this study is that financial news articles have an effect on the future prices of stocks. This is mostly due to the fact that investors use news sources to evaluate their perception on a stock's future price. Hence certain news stories are expected to lead to a positive outlook and attract more buyers, while other news stories lead to a negative outlook leading to an increase in selling. A notable example of this phenomenon taking place occurred when news about Russia's invasion of Ukraine in 2022 surfaced. As a result of the negative and uncertain sentiment surrounding this event, stock markets trended lower for several days.

1.1 Problem Definition

The primary goal of this research is to investigate the predictive power of news articles on financial market trends. Given the complexity of the stock market and the many factors that contribute to changes in stock prices, as discussed in Section 1, this is a challenging task. To address this challenge, AI is leveraged to obtain sentiment from news articles and effectively generalise over large historical data. The research aims to develop and evaluate ML models that incorporate news data as a feature to predict the direction of major United States (US) stock market indexes. To analyse the performance of news data as a feature, ML models are trained and tested with the following three variations of input types:

1. Technical indicators only: this input type uses mathematical calculations based on historical prices and volume of a stock.

2. News features only: this input type uses NLP techniques to represent the news articles as vectors of features extracted from article text.
3. Combination of both technical indicators and news features: this input type combines both technical indicators and news features to provide a more comprehensive view of market conditions.

In order to determine the effectiveness of the input features in predicting future stock trends, this research implements variations of each of the input types mentioned above. The hypothesis is that news-based models should outperform technical-based models. Furthermore, it is worth exploring the potential performance improvement resulting from combining technical indicators and news sentiment.

The results of this research have the potential to make significant contributions to the field of stock market research, particularly in terms of understanding the impact of news articles on market trends. Additionally, this project aims to provide valuable insights to individuals seeking a better understanding of the relationship between news and the stock market.

1.2 Aims and objectives

The main research aim of this project is to address the following question: "Do news articles contain any predictive power on future stock market prices?". In simpler words, "Can one accurately predict the stock market trend by primarily using news articles?". To achieve this, the following objectives have been set:

- Objective 1 (O1): Identify the best input features and ML model to forecast the short term stock market direction.
- Objective 2 (O2): Quantify the effectiveness of news article features to predict stock market direction.
- Objective 3 (O3): Determine the most effective lookback window and prediction horizon for forecasting stock market direction.
- Objective 4 (O4): Compare the predictive power of news features on different financial securities.

The project investigates the primary research aim by addressing the mentioned four objectives. The first objective involves evaluating the performance of various standard and deep ML methods with different input features, and comparing and contrasting their results. Regarding O2, since news reports are composed of textual data, it is necessary to apply an appropriate technique to convert this textual data into a machine-readable format. Therefore, several NLP techniques are experimented with, including Bag of Words (BoW) approaches and GloVe approaches. Moreover, in O3, the lookback configuration and the prediction horizon for stock market direction are investigated. Finally, through O4, the research examines the predictive performance of the final news-based model on a range of US financial assets, such as major stock market indexes, market sector Exchange-traded funds (ETFs), and popular individual stocks.

1.3 Project Structure

The project layout is structured as follows:

- Background: This chapter provides fundamental information about key concepts of the project, including ML models, NLP techniques and stock market data.
- Literature Review: A comprehensive review that discusses in detail the existing research on AI based stock market trend prediction systems. This includes an examination of the news data used, NLP techniques and features utilised by other researchers.
- Methodology: The proposed methodology explains how the project aims to address the objectives and develop a reliable stock market trend forecasting model.
- Evaluation: This chapter examines the performance and effectiveness of the developed models, emphasising their accuracies and addressing the objectives.
- Conclusion: This chapter highlights the project's accomplishments and interesting findings. Moreover, potential avenues for future work is provided.

By following this project structure, a cohesive and coherent overview of the conducted work is presented.

2 Background

The aim of this chapter is to present fundamental information about the key components of this project. Specifically, it covers the standard ML methods: LR, SVM, and RF Classifier, and the deep learning methods: ANN, LSTM network, and GRU network. These methods were chosen as they have been successfully utilised by other researchers in similar stock market forecasting systems. Furthermore, this chapter also discusses the NLP representations used: TF–IDF and GloVe, as well as the technical indicators employed. Thus, this chapter provides the reader with a comprehensive explanation of the crucial concepts in a bid to improve the overall understanding of the project.

2.1 Standard ML Methods

LR is a supervised ML technique widely used for binary classification tasks [11]. Similar to linear regression, it attempts to fit a line to the data, but it differs by utilising a sigmoid function to transform the output into a probability value ranging from 0 to 1. This probability value represents the likelihood of the item being classified as part of a particular class. Finally the probability is converted into 0 or 1 depending on a threshold, 0.5 for binary classification. To conclude, LR is a useful tool to predict binary outcomes.

SVM is a supervised ML method that uses support vectors to establish hyperplanes for classifying data points [12]. The hyperplanes are determined by identifying the optimal separation between the classes among the data points in the training set. The key idea behind SVM is to find the hyperplane that maximises the margin between the support vectors, being the data points closest to the hyperplane, of the different classes. The margin is defined as the distance between the hyperplane and the closest data points from each class. By maximising the margin, SVM aims to find a decision boundary that generalises well to unseen data. Moreover, SVM offers the flexibility of using kernel functions [13], for example Radial Basis Function (RBF), to transform the input space and hence enabling the separation of non-linearly separable data points.

RF is a supervised ML method that utilises an ensemble of decision trees [14]. The algorithm constructs each decision tree using a random subset of the training data and a random subset of the features. Since RF uses random subspace methods and bagging, they are less sensitive to the training data compared to when using only one decision tree. Each tree in the random forest makes a prediction, in the case of binary classification: 0 or 1. From these votes, the majority vote is typically chosen to classify the input data. Overall, RF is an algorithm that can effectively handle a wide range of data types while containing useful structures that prevent overfitting.

2.2 Deep Learning Methods

ANN is inspired by the structure and function of the biological brain [15]. An ANN is a collection of perceptrons and activation functions that map input to output. By making use of an input layer, hidden layer/s and an output layer, an accurate representation of a function is captured based on the input data. During training, the ANN adjusts the weights of the connections between the perceptrons in order to minimise the error between the predicted output and the actual output. To address overfitting, techniques such as regularisation and dropout can be employed. In conclusion, ANNs are highly flexible and can be used for a wide variety of tasks including binary classification.

LSTM network, proposed by S. Hochreiter et al. [16], is designed to overcome the vanishing gradient problem that occurs in standard Recurrent Neural Network (RNN). LSTM consist of a set of memory cells connected by three gates: the forget gate, the input gate, and the output gate. The forget gate determines the amount of previous cell state to forget, while the input gate determines how much of the new input to add to the cell state. The output gate controls the amount of cell state to output. The gates regulate the flow of information into and out of the cells, thus enabling LSTM networks to retain long-term dependencies. All in all, LSTM networks are highly effective models widely used for processing sequential data.

GRU, proposed by Cho et al. [17], is a special type of RNN similar to LSTM networks but computationally simpler since it consists of only two types of gates: an update gate and a reset gate. Essentially, the update gate determines how much of the previous state to keep and how much of the new state to add, while the reset gate controls how much of the previous state to forget. This gating mechanism solves the vanishing gradient problem by regulating the flow of information throughout the network. During training, the weights of the connections between neurons are adjusted to minimise the error between predicted and actual outputs, a process similar to that of standard RNNs. In conclusion, GRU networks are a powerful type of Neural networks that can capture long-term dependencies in sequential data.

2.3 Text Representation in NLP

Text representation is a crucial aspect of NLP that converts textual data into a numerical format suitable for ML. Several methods exist for text representation, these include: BoW methods, Word2Vec and Doc2Vec. In this project, two methods are employed: TF–IDF and GloVe. TF–IDF assigns weights to words based on their frequency in a document and inversely proportional to their frequency in the whole corpus, resulting in a matrix describing the relevance of each word to a document in a collection of documents. On the other hand, GloVe, introduced by Pennington et al. [18], improves upon Word2Vec by using co-occurrence statistics between words to capture their semantic and syntactic relationships in a text corpus. It constructs a word-word co-occurrence matrix that provides both global and local information to generate word vectors. Moreover, GloVe can represent a whole sentence by averaging the individual word vectors to obtain a single vector representation for the entire sentence. Overall, both TF–IDF and GloVe are considered to be effective textual representation methods in NLP.

2.4 Stock Market Data

Generally, researchers that aim to forecast stock prices use technical and/or external indicators. Technical data is derived from the stock market itself and includes stock prices, trading volumes, and technical indicators [19] such as Relative Strength Index (RSI), Money Flow Index (MFI), Exponential Moving Average (EMA) and others. For example, RSI is calculated using average price gains and losses over a given period of time. In contrast, external data is derived from outside sources such as news articles and social media sites. News data is particularly important in stock market analysis as it can significantly impact market sentiment and hence investor behaviour. Moreover, news articles related to finance can provide information about events and market conditions. For instance, if the news suggests that a specific sector is likely to grow in the coming years, investors may decide to buy stocks in that sector in anticipation of a potential rise in stock prices.

In summary, this chapter has provided an overview of the fundamental components of this project. It has covered standard ML methods, deep learning models, NLP techniques and stock market data used for prediction. The project methodology incorporates these concepts, and the research findings highlights the effectiveness of the mentioned models in forecasting financial market trends.

3 Literature Review

This chapter provides an extensive review of the relevant research in the field of stock market prediction. This includes exploring the feasibility of stock market prediction, understanding trading decisions, examining commonly used technical indicators, an overview of successful ML techniques used in this area and highlighting the most similar systems. The purpose of this chapter is to enhance the understanding of the problem and existing approaches, establish expectations and facilitate the design and implementation of an effective AI driven stock market predictor solution. In summary, this comprehensive review of existing research enriches the proposed methodology by enhancing the understanding of the related work and field.

3.1 Possibility of Stock Market Prediction

The possibility to predict the stock market over a long period of time has been heavily debated by researchers. As previously mentioned, EMH and Random Walk Hypothesis both state that predicting stock prices using past data is futile. Nevertheless many researchers [3–10, 20–26], especially with the rise of AI solutions, challenged these hypotheses and many managed to develop models that predict stock market movements with noticeable accuracy. These models usually make use of historical stock data (open, close, volume, etc) and/or external data (such as news, social media sentiment, etc). The methodologies used to successfully predict the stock market are discussed in further detail in subsequent sections.

3.2 Understanding Trading Decisions

In the investment world there are two contrasting paradigms used by traders to analyse stock prices, technical analysis [6] and fundamental analysis [27]. Technical analysis focuses solely on historical stock prices and seeks patterns to speculate the future direction of a particular stock. On the contrary, fundamental analysis forecasts future stock movements by factoring in a number of economical and financial variables, such as, revenues, earnings, interest rates, etc. In both cases it is assumed that the traders are rational actors. On the other hand, Behavioural Finance [28] contradicts this theory, stating that investments are influenced by human emotion and contain many cognitive biases when processing information. For instance, H. Kent Baker et al. [29] found that anxiety increases the perceived risk of an investment and familiarity bias causes investors to prefer local assets that they are more familiar with. Moreover behavioural finance provides an explanation for market anomalies, such as stock market bubbles and depressions. Therefore, since news is known to influence public mood, news sentiment should have an effect on stock markets [30]. In essence, the premise is that positive news tends to foster optimism among investors, which in turn attracts more buyers and causes stock prices to trend higher. Conversely, negative news tends to create anxiety among investors, leading to more sellers and hence causing stock prices to trend lower.

3.3 Technical Indicators for Stock Market Prediction

As previously mentioned, a way to analyse the stock market is through technical analysis. In technical analysis, many technical indicators are employed to examine and forecast the future performance of the stock market [31]. These indicators are based on statistical calculations derived from historical stock price and volume data [32]. Many believe that these indicators provide valuable insights into market trends, price movements and volatility. Overall, technical indicators are useful tools for understanding stock market movements. Recently, there has been an increasing interest in combining technical indicators with ML to predict future stock prices [7–10]. This is mostly due to the fact that ML models have the ability to analyse large amounts of historical data and identify recurring patterns. This section highlights a number of successful ML based stock market prediction systems that consist of technical indicators. This review assists in the development of our technical-based models.

Researchers commonly obtain historical stock price data containing daily open, high, low, close prices, and volume from Yahoo Finance API [7, 9, 10]. From these values, various technical indicators can be calculated. A variety of technical indicators and ML models have been used by researchers to predict stock market movements. For instance, Ishita Parmar et al. [7] simply used stock price data as input features to the ML models. While Osman Hegazy et al. [8] used the following five technical indicators: RSI, MFI, EMA, Stochastic Oscillator and Moving Average Convergence/Divergence (MACD) to predict the stock price. Similarly, Mehar Vijh et al. [9] fed the following 6 features: Stock High minus Low price (H-L), Stock Close minus Open price (O-C), Stock price's seven days' moving average (7 DAYS MA), Stock price's fourteen days' moving average (14 DAYS MA), Stock price's twenty one days' moving average (21 DAYS MA) and Stock price's standard deviation for the past seven days (7 DAYS STD DEV) to an ANN and RF. They noted that ANN outperformed the RF model. Moreover, Sanbo Wang [10] used 11 technical indicators; Moving average, Bollinger Bands, Arron Up, Arron Down, Commodity Channel Index, Chande Momentum Oscillator, MACD, RSI, Stochastic Oscillator K%, Stochastic Oscillator D% and WILLR %R to predict the S&P 500 index. This section demonstrates the potential of using several technical indicators combined with the appropriate ML models to forecast stock market movements. In our research, we replicate the last three feature variants to input them into the technical-based models.

3.4 Acquisition of textual news data

In order to develop an accurate stock market prediction system it is crucial to acquire textual news articles from reliable and up-to-date sources. Many researchers utilise news sources coming from mainstream news publications. For instance, Karl Sant Fournier [24] made use of news articles obtained from Guardian's open source API while Zhong et al. [20] used NYT API. Moreover, Duc Duong et al.[3] made use of 1,884 news articles relating to companies within the VN30 index. On the other hand, some researchers made use of social media sites to gather popular news articles. For instance, Yang Liu et al. in [22] obtained the top-5 news stories from Reddit. Moreover, other researchers utilised both social media and news publications, such as, Wasiat Khan et al. [4] using social media data from Twitter and Financial news headlines from Business Insider to predict the direction of certain US stocks. It is important to note that many researchers [4, 20, 22, 24] used the headline instead of the full textual content of the news article.

3.5 NLP Processing of News Articles Text

The reviewed studies utilised various NLP techniques to represent news articles in a machine-readable format. This section discusses the most common representations. BoW methods, specifically, variants of TF–IDF, are commonly opted for with Wasiat Khan et al. [4] concluding that BoW is a simple yet effective representation that achieves satisfactory performance. Moreover, Duc Duong et al. research [3] compared the effectiveness of three TF–IDF methods; TF–IDF, Delta TF–IDF and Delta TF–IDF combined with a custom sentiment dictionary. From these the latter representation performed best. Similarly, Qasem et al. in [25] compared the performance of Unigram TF–IDF and Bigram TF. They noted that the model making use of Unigram TF–IDF had a better overall accuracy.

Approaches that go beyond BoW methods incorporate mark-up of text using NLP tools such as Part-of-speech (POS) tagger and Named Entity Recognition (NER). For instance, in the study done by Schumaker and Chen [5], they compared the accuracy of four textual representations; BoW, Noun Phrases, Named Entities and Proper Nouns. They noted that Proper Nouns scheme had the best results. Additionally, word embeddings are also used, for example Yang Liu et al. [22] used GloVe and TF–IDF representations. They noticed that RNN performed better using GloVe while Convolutional Neural Network (CNN) performed better using TF–IDF. This demonstrates that the NLP representation employed depends on the ML model used. In summary, having an accurate ML model requires having a suitable textual representation, therefore, multiple NLP representation are attempted in our research.

3.6 ML techniques

In the reviewed studies both standard ML methods and deep learning methods were successfully utilised. This section explores various ML methods that performed well in similar stock market prediction systems. For instance, Győző Gidófalvi [26] made use of a Naïve Bayesian text classifier to classify news articles and forecast stock price movements. Moreover, in the research conducted by Qasem et al. [25], they employed LR and an ANN to predict the direction of a stock from Twitter sentiment classification. From the results they concluded that the ANN and the LR model had the same overall accuracy when using representation.

A comprehensive comparison of standard ML methods and deep learning methods is conducted by Yang Liu et al. [22]. They compared the performance of three standard ML methods; LR, SVM and RF and three deep learning methods; CNN, LSTM and GRU at predicting the direction of the stock market using news headlines. They noted that in general the deep learning models outperformed standard ML models. In particular the GRU and LSTM networks achieved superior performance. Additionally, they noted SVM performed best among the standard ML models.

Overall, this section summarises the relevant ML techniques for predicting stock market trends based on news articles, and highlights the success of both standard and deep learning models in this domain. These studies provide insight into how various ML methods perform when given the task to predict stock market direction from news article sentiment. Hence, in our research a variety of standard and deep learning methods are utilised.

3.7 Most Similar systems that Integrating News Sentiment

In this section, AI based systems that integrate news articles in their stock market prediction are identified and discussed in detail. Some researchers relied exclusively on news headlines to forecast financial trends. For instance, Yang Liu et al. [22] implemented a solution that predicts the Dow Jones Industrial Average Index by using daily top-5 news headlines from social media platform Reddit. This solution uses same day's headlines and 5 days' headlines to predict the index's direction from market open to market close. Various NLP techniques were employed to transform the headline text into a machine-readable format. Specifically, the best accuracy using same day headline prediction and 5-day headlines prediction were 57.3% and 59.6% respectively. This suggest the importance of using a lookback.

Similarly, José G. de Araújo Júnior and Leandro B. Marinho [33], investigated the impact of economical news on the leading stock exchange of Brazil, 'A Bolsa do Brasil B3'. This was done by collecting economic news between 2000 and 2015 from a well known Brazilian newspaper. Sentiment analysis was used to label the news article as positive, negative or neutral. From the results they concluded that the ML classifiers outperformed Random and Keep Trend baselines. Additionally, the researchers made predictions for various sectors and observed that news articles have varying predictive power across different sectors. Notably, the Oil, Gas, and Biofuel sector demonstrated the highest predictability. This observation was attributed to the newspaper's extensive coverage of companies in that sector. The finding highlights the important of considering sector-specific news in stock market prediction models.

Other researchers utilised a number of external factors. For instance, Mehak Usmani et al. [23], proposed a ML solution that predicts the direction of the Karachi Stock Exchange by making use of a number of external factors that were deemed to be influential to the index. These input attributes included Oil rates, Gold & Silver rates, Interest rate, Foreign Exchange rate, financial news sentiment, social media sentiment and others. The data used in this study spread over 4 months. They concluded that the best performing model was ANN. Interestingly, they found that the attribute 'Oil Rates' had the greatest impact on the Pakistani stock exchange, while 'Foreign Exchange rate' showed no significant association. These findings emphasise the importance of considering external factors when developing a stock market prediction system.

In a study conducted by Taylan Kabbani et al. [20], both news sentiment analysis and technical indicators were utilised. The news articles were obtained from a publicly available dataset [21] made up of news collected from well-known news publishers such as Reuters, NYT and others. In this study the news data was represented as a single value between -1 (negative) and 1 (positive), obtained by employing the Vader sentiment analysis tool [34]. The technical indicators used in this research included Today Trend, Tomorrow's Trend (target feature), RSI (14 days), simple moving average (14 days), Stochastic Oscillator %K (14 days) and stock's high, low, close, and volume. These features were fed into three ML methods, LR, RF, and Gradient Boosting Machines (GBM), that were trained on data for three US stock stocks; Apple Inc. (AAPL), Amazon.com Inc. (AMZN) and Netflix Inc. (NFLX). The authors noted that RF slightly outperformed the other ML models, achieving an accuracy of 63% when predicting tomorrow's trend for Netflix stock. In our study, the features of this study are duplicated to create a hybrid model that is combining News data and technical data.

In summary, this section provides an overview of the successful stock market trend prediction systems. Delivering valuable insights on problem definition, feature selection and emphasising the utilisation of ML models based on specific arrangements such as input features and output considerations. Overall, the studies reviewed suggest that using news articles can lead to accurate stock market trend forecasting.

In conclusion, this chapter has highlighted a number of approaches and techniques utilised by researchers to develop reliable and accurate stock market forecasting systems. The reviewed studies provide an overview of the potential technical indicators, NLP techniques for representing news data, methods of acquiring reliable news articles and successful ML models. By gaining insights into these existing approaches, a more informed and effective methodology can be proposed for developing an AI driven solution that predicts financial market trends using news as an external feature.

4 Methodology

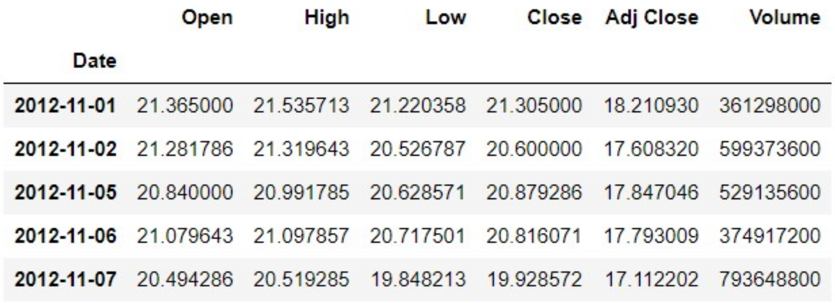
The methodology chapter aims to provide a detailed overview of the design and implementation of the AI driven stock market prediction system. This chapter covers all the important decisions and steps taken in the project, including selecting and preprocessing the data, selecting and developing the ML models, the training and testing of the model and the evaluation plan. This chapter aims to provide a comprehensive overview of the steps taken in the project to address the main objectives and the reasons behind every key design decision.

For access to the complete code used in this project, please refer to the GitHub repository available at: `https://github.com/adarm08/FYP_source_code`

4.1 Data Acquisition and Preparation

The data plays a vital role in determining how successful the ML models can be. This is especially true when considering stock market prediction systems. These systems require accurate and relevant data. Moreover, stock market data is a time series, this adds another layer of complexity due to the temporal nature of the data. For instance, specific patterns may only occur during certain time periods and historical data may not always be relevant for predicting future trends. This makes the task of acquiring reliable data a crucial step in developing any stock market prediction system.

4.1.1 Stock Market Data



	Open	High	Low	Close	Adj Close	Volume
Date						
2012-11-01	21.365000	21.535713	21.220358	21.305000	18.210930	361298000
2012-11-02	21.281786	21.319643	20.526787	20.600000	17.608320	599373600
2012-11-05	20.840000	20.991785	20.628571	20.879286	17.847046	529135600
2012-11-06	21.079643	21.097857	20.717501	20.816071	17.793009	374917200
2012-11-07	20.494286	20.519285	19.848213	19.928572	17.112202	793648800

Figure 4.1 Historical Stock Market Data for Apple Stock ($AAPL).

Stock market data is the essential source of information for the system. Hence, it is extremely important to be correct. As mentioned in the Literature review many researchers opted for the Yahoo Finance API to obtain historical stock market data. Therefore, we query the Yahoo Finance API to obtain historical stock market data for a specific stock and time frame.

In the development stage, historical market data for the S&P 500 index over the last decade (2013-2023) was utilised. The S&P 500 is widely recognised as a key benchmark representing the overall performance of the US market, making it a suitable index to focus the research on. The Yahoo Finance API returns a dataframe that contains information about the stocks' daily key data points. These are the opening and closing prices, the intraday high and low prices, and the volume of trades on the specific day. Figure 4.1 visualises an exemplary dataframe obtained for Apple stock. Many important technical indicators and performance features are calculated from this data. We are focusing on predicting from market open till market close, therefore the label is 1 if open $<$ close and 0 if open $>$ close.

4.1.2 News Data

News article text is the primary source of data in this stock market trend forecasting system. Therefore, it is important to collect relevant news articles that closely relate to the stock market. As mentioned in the Literature review there are various sources we can consider. Given that the research's main focus is the news data, various options were examined and tested. The first dataset considered was a popular Reddit dataset[1] used by Yang Liu et al. [22]. This dataset consists of the top 25 news headlines for days between 2008-08-08 and 2016-07-01. The drawbacks with this dataset is that it consists of popular world news articles that in some cases are irrelevant to the stock market. For that reason this dataset was dismissed. Another interesting dataset was the 'All the news' dataset [2], that was used by Kabbani et al. [21]. This consists of 2.1 million news articles originating from reliable business sources. However, the issue with this dataset was that it is computationally expensive given its size and additionally when exploring the data it was noted that it consisted of redundant news stories. Those were the main reasons why this dataset was dismissed.

A different approach for collecting the news articles observed in the Literature review was by making use of a news publication APIs. Two non-commercial API were tried and compared: Guardian API[3] and NYT API[4] used by [24] and [20] respectively. These APIs allow individuals to query for historical news articles from specific news sections, such as, business and financial. This is especially useful as it allowed us to construct flexible datasets containing only business related articles for any given time period. The limitation of these APIs is that they do not provide the full text, only the headline and an abstract. However, as noted in the Literature review, many researchers simply used the headlines. When comparing the two news sources, Guardian and NYT,

[1] Daily News for Stock Market Prediction, Version 1 `https://www.kaggle.com/aaron7sun/stocknews`

[2] All the news 2.0 https://components.one/datasets/all-the-news-2-news-articles-dataset

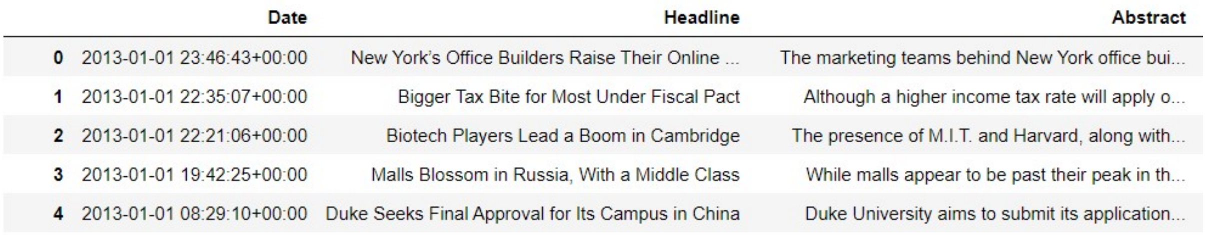
[3] https://open-platform.theguardian.com/

[4] https://developer.nytimes.com/apis

it was concluded that although both report high quality news articles, the stories reported by Guardian seemed to relate more to the UK market while the NYT stories focused more on US stories. Therefore, the decision was made to make use of the US based publication rather than the British newspaper, as the stocks in question are more relevant to news and events concerning the American market.

4.1.3 Building the Financial News Article dataset

In the previous section, we discussed how the NYT API was selected as the data source. The next step is to construct a dataset that can be utilised to train and test the ML models. For this purpose, a time period from 2013-01-01 till 2023-05-01 was chosen, this spans over the last decade and hence is expected to include a variety of market cycles and events. In order to build the required dataset, python library pynytimes[5] was used. This library offers a simple means of querying the NYT API. Articles from the financial and business sections for each day between 2013-01-01 and 2023-05-01 were extracted by querying the API. From the response string, the publication date and time, headline and abstract were stored. This resulted in approximately 57,500 articles.



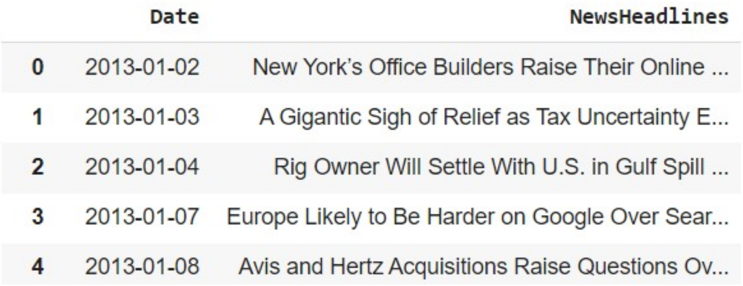
	Date	Headline	Abstract
0	2013-01-01 23:46:43+00:00	New York's Office Builders Raise Their Online ...	The marketing teams behind New York office bui...
1	2013-01-01 22:35:07+00:00	Bigger Tax Bite for Most Under Fiscal Pact	Although a higher income tax rate will apply o...
2	2013-01-01 22:21:06+00:00	Biotech Players Lead a Boom in Cambridge	The presence of M.I.T. and Harvard, along with...
3	2013-01-01 19:42:25+00:00	Malls Blossom in Russia, With a Middle Class	While malls appear to be past their peak in th...
4	2013-01-01 08:29:10+00:00	Duke Seeks Final Approval for Its Campus in China	Duke University aims to submit its application...

Figure 4.2 Sample dataset as extracting from NYT API

To create a more suitable dataset for prediction, the original dataset shown in Figure 4.2 was transformed to consist of news articles for every trading day, similar to how Yang Liu et al. [22] preprocessed the headlines. Considering that not every day is a trading day, with the stock market being closed on weekends and certain holidays, the number of articles per trading day adds up to approximately 23. To create a more compelling representation of what the article is about the decision was made to combine the headline and abstract columns, following the approach of Zhong et al. [20]. Additionally, the news articles within Figure 4.2 were sorted according to market days. This was done by grouping and joining the articles that were released between market close and next market open. For example, the articles released from Friday, after market close, till Monday, before market open, are used to predict Monday's trading day (if Monday is not a holiday). The resulting dataset, shown in Figure 4.3,

[5] https://github.com/michadenheijer/pynytimes

provides the necessary news data to serve as the foundation for ML training. The appendix Section A presents an exploration of headline keywords across different years, examining their correlation with subsequent market trends.



	Date	NewsHeadlines
0	2013-01-02	New York's Office Builders Raise Their Online ...
1	2013-01-03	A Gigantic Sigh of Relief as Tax Uncertainty E...
2	2013-01-04	Rig Owner Will Settle With U.S. in Gulf Spill ...
3	2013-01-07	Europe Likely to Be Harder on Google Over Sear...
4	2013-01-08	Avis and Hertz Acquisitions Raise Questions Ov...

Figure 4.3 Final news dataset

4.2 NLP representation

In order to use news article headlines as the foundation of our predictive model, it is necessary to convert the text into a machine-readable numerical format. This is required since ML models cannot directly process textual data. Therefore, the next step is to convert the article text into a format that can be used by the ML models. As mentioned in the Literature review there are a number of NLP representations available, such as, TF–IDF, GloVe and Noun Phrases. To investigate Objective 2, multiple methods have been selected so that their performance can be examined and compared. Namely, TF–IDF and TF–IDF variants, due to its success in [3, 22, 25] and GloVe due to its success in [22].

4.2.1 TF–IDF Technique

The methodology used to compute TF–IDF and TF–IDF variants are discussed in detail in this section. To make this process straightforward, the python library Natural Language Toolkit (NLTK)[6] was utilised. The methodology applied to compute the TF–IDF vectors is the following:

1. Tokenize and stem the news articles using RegexpTokenizer and PorterStemmer respectively. Stop words were not included in the token list, since these common English words do not contain meaningful information for the prediction. The NLTK stop word list was utilised.

2. Extract the vocabulary, that is a list that contains all the unique words found in the list of tokenized text data.

[6] https://www.nltk.org/

3. Find Term frequency (TF). This is calculated by counting the number of times a word appears in the article, then normalise it by dividing it by the maximum count of any word in the same article.

4. Reduce number of terms. Find the standard Document Frequency (DF) for every term in vocabulary by counting in how many articles the term is present and if DF is less than N, a numerical threshold, example 100, remove the term.

5. Find Inverse Document Frequency (IDF) by taking the logarithm of the number of documents present in the corpus divided by the number of documents where the term appears. The documents in this case are news articles grouped by trading days, hence they are time bounded. Since future information must not interfere with present prediction, a lookbackdf() function was created. lookbackdf() is a function to get document frequency, number of documents in which term is present, in the previous n trading days. For simplicity n is fixed at 5 trading days.

6. Compute TF–IDF, By multiplying TF and IDF. This matrix is then normalised using L2 normalisation.

It is ideal to test various TF–IDF variants. In step 4, N plays a significant role in determining the size of the vocabulary by filtering out uncommon words. Varying this variable leads to sparser or denser representations. Moreover, one is to expect that terms that are frequently mentioned carry more meaning than other less mentioned terms. For these reasons 4 variations were developed with N $>$ 3, N $>$ 50, N $>$ 100 and N $>$ 200. Another way to make the representation more compact is by only using the terms that have negative or positive connotations. This will lead to a TF–IDF representation with a custom sentiment dictionary similar to [3]. In this case sentiment Valence Aware Dictionary for Sentiment Reasoning (VADER)[7] is used to determine the sentiment of the term. Terms with neutral sentiment were removed. This variation uses just 658 unique terms as vocabulary, however certain context might be missing due to only keeping terms with positive or negative sentiment. Nevertheless, it should be interesting to compare and contrast with the other TF–IDF variants.

4.2.2 GloVe

As mentioned in the literature review, while there exist several word embedding methods, such as, Word2Vec and Transformers, GloVe has been a popular choice to represent news data in previous similar research studies. In order to transform textual data into numerical vectors, GloVe was implemented using the spaCy library[8]. The

[7] https://vadersentiment.readthedocs.io/en/latest/

[8] https://spacy.io/usage/embeddings-transformers

‘en_core_web’ model in spaCy includes pre-trained word embeddings that can be used for this purpose. The implementation process involves tokenizing the news article text, and finding the length of the longest vector to ensure uniform length for all vectors. Then, for each token in the headline, the corresponding GloVe vector is obtained using spaCy. The list of vectors is then padded to the required length, and the GloVe vector for the entire text is computed by averaging the GloVe vectors for each token in the text.

4.2.3 Sentiment Scores

An alternative approach to deal with news article textual data is to obtain the sentiment score for each trading day from the news articles. To achieve this, the sentiment VADER tool is utilised to analyse the sentiment of each daily news items. The tool provides compound polarity scores that measure how negative or positive the news articles are. Therefore, each trading day can be represented as a single value representing the collective news articles sentiment. It is interesting to examine whether relying solely on sentiment scores produces comparable performance to that when utilising NLP representations.

4.3 Feature Selection

Feature selection aims to investigate the primary research question: ”Do news articles contain any predictive power on future stock market prices?”. To address this question, various input data features must be tested. This is done by mainly compering between the following three types: models that base their prediction solely on technical indicators, models that base their prediction primarily on news data and models that use both news data and technical data. Furthermore, as there are several technical indicators and news data representations available, variations of these three feature types are explored. Tables 4.1, 4.2 and 4.3 illustrate the types of input data features that have been tested along with justification where appropriate.

Table 4.1 Technical Models

Model	Features	Justification
Technical Model 1	rsi_7, mfi_7, ema_7, so_%K, so_%D, macd + Tomorrow's open price	The selection of the technical indicators is based on those utilised in the study conducted by Osman Hegazy et al. [8].
Technical Model 2	high-low, open-close, sma_7, sma_14, sma_21, std + Tomorrow's open price	The selection of the technical indicators is based on those utilised in the study conducted by Mehar Vijh et al. [9].
Technical Model 3	sma, bb_middleband, ar_up, ar_down, cci, cmo, macd, rsi, so_%K, so_%D, willr + Tomorrow's open price	The selection of the technical indicators is based on those utilised in the study conducted by Sanbo Wang [10].

Table 4.2 News Based Models

Model	Features
News Based Model 1	TF-IDF vectors with df $>$ 3
News Based Model 2	TF-IDF vectors with df $>$ 50
News Based Model 3	TF-IDF vectors with df $>$ 100
News Based Model 4	TF-IDF vectors with df $>$ 200
News Based Model 5	TF-IDF vectors with df $>$ 100 + sentiment score
News Based Model 6	TF-IDF vectors with custom sentiment dictionary
News Based Model 7	GloVe vectors

Table 4.3 Hybrid Models combining Technical Indicators and News Data

Model	Features	Justification
Hybrid Model 1	High, Low, Close, Volume, Today return, Tommo_Open, rsi, %K, sma + News sentiment score	Features based on those utilised in the study conducted by Taylan Kabban [20].
Hybrid Model 2	High, Low, Close, Volume, Today return, Tommo_Open, rsi, %K, sma + TF-IDF vectors with df $>$ 200	Features similar to Hybrid Model 1, but instead of using news sentiment scores, TF-IDF vectors with df $>$ 200 are employed.
Hybrid Model 3	sma, bb_middleband, ar_up, ar_down, cci, cmo, macd, rsi, so_%K, so_%D, willr + News sentiment score	A model combining the best performing technical indicators with the news sentiment scores.
Hybrid Model 4	sma, bb_middleband, ar_up, ar_down, cci, cmo, macd, rsi, so_%K, so_%D, willr + TF.IDF vectors with df $>$ 200	A model combining the best performing technical indicators with the TF-IDF vectors with df $>$ 200.

4.4 Developing the Models

In this section, the development of the ML models that are employed to predict the stock price direction are discussed in detail. As observed in the Literature review, both standard and deep learning methods have been shown to perform successfully. To investigate Objective 1 various ML models have to be considered. Hence, three standard ML models: LR, SVM and RF, and three deep learning models: ANN, GRU and LSTM were opted for. The data preprocessing and standard ML models were implemented using python library scikit-learn[9], while the deep learning models were implemented using Pytorch[10].

[9] Scikit-learn https://scikit-learn.org/stable/

[10] PyTorch https://pytorch.org/

4.4.1 Preprocessing the Data

Prior to fitting the data to the ML models, the technical and stock data were normalised using z-score normalisation. To ensure a reliable representation of the predictive performance of the model, several train-test splits were applied. This was inspired by how Fazlija et al. [35] evaluated their models. The various train-test splits are displayed in Table 4.4. Moreover, the last 10% of the training set was partitioned to be used as the validation set. Generally, the train set is used to train the model, the validation set is used to tune the hyperparameters and the test set is left to evaluate the final performance of the model. In summary, this approach ensures that the data is appropriately used for the training and testing of the various ML models.

Table 4.4 Train-Test Splits

Train + Validation Period	**Test Period**
2013-01-01 to 2019-12-31	2020-01-01 to 2020-12-31
2013-01-01 to 2020-12-31	2021-01-01 to 2021-12-31
2013-01-01 to 2021-12-31	2022-01-01 to 2022-12-31

4.4.2 Standard ML Models - LR, SVM and RF

In this section the implementation choices for the standard ML models are discussed. Firstly, LR was fitted to the entire training dataset, without the need for a validation set since there was no hyperparameter tuning required in this case.

In contrast, RF has a crucial hyperparameter: n_estimators, that specifies the number of decision trees in the forest. This parameter controls how much of the training data is learnt, and with a validation set it can be used to mitigate overfitting. The tested values for n_estimators were 64, 128, 246, 400 and 500. These values cover a wide range of complexities for the RF model, enabling the selection of a suitable balance between learning and overfitting. The selected RF model was that with the highest validation accuracy.

Additionally, SVM has a couple of interesting hyperparameters that need to be considered. Three important hyperparameters are the kernel type determining the decision boundary, the regularisation parameter, C, controlling the complexity and gamma, controlling the shape of the decision boundary in non-linear SVMs. These hyperparameters affect the performance of the SVM model hence various combinations are attempted. The tested values were: 'kernel': ['linear', 'rbf'], 'C': [0.1, 1, 10], 'gamma': [0.1, 1, 10]. The chosen kernel types were popular in similar research and the values for C and gamma were selected to cover a suitable range of options. A grid search approach was employed to conduct an exhaustive search of the hyperparameter space, ensuring that a variety of hyperparameter combinations are considered. Furthermore, to assess the model's performance, a five-fold time series cross-validation approach was chosen. The hyperparameter combination with the highest cross validation accuracy was selected and used to build the final SVM model.

4.4.3 Deep Learning Models - ANN, GRU and LSTM

This section focuses on the design and decisions made when implementing the deep learning models. Prior to training the models, further preprocessing is required. The data sets are converted to tensors of type float32 and stored on the GPU. Additionally, for the GRU and LSTM the data needs to be in the following format: 3D tensor (batch size, sequence length, input size). To achieve this format, the data is processed by creating a sliding window view of the data using the lookback value. Prior to processing the data, to maintain consistency between the validation and test sets, they are concatenated with the preceding lookback days of training data. These sequences are then converted to PyTorch tensors using the torch.stack() function. The corresponding labels are sliced to match the length of each sequence. Furthermore, to enable batch processing PyTorch TensorDataset and DataLoader objects are created from the processed data. These dataloaders are used to feed the data to the GRU or LSTM model in batches.

A crucial aspect of deep learning models is the architecture of the Neural Network. Here, a challenge arises since the model's input changes depending on the feature configuration. Thus, it is necessary to have a suitable architecture that tries to accommodates all feature types. This is a particularly difficult task due to the fact that there are 14 configurations. The proposed solution was to design a simple architecture and then rely on other hyperparameters to adjust the complexity as needed. When designing a Neural Network the important design decisions are:

- The number of layers in the network. As previously mentioned, a simple architecture was opted for. The ANN layout consists of an input layer, followed by three fully connected layers with RELU activation and an output layer. Furthermore, for the GRU and LSTM models the layout includes an input layer, a fully connected layer with RELU activation, a dropout layer, a GRU/LSTM unit, another fully connected layer and an output layer. The model architecture can be summarised as follows: input $\rightarrow$ FC $\rightarrow$ ReLU $\rightarrow$ Dropout $\rightarrow$ GRU/LSTM $\rightarrow$ FC $\rightarrow$ output.

- The hidden layer size, in general the more hidden units the complex the representation is. For the ANN the hidden layer size was evaluated at 64, 128, 256, 400, 500 and 750. For the GRU and LSTM there are two different hidden layer size variables, one determining the output size of the first fc layer and another determining the output size of the GRU or LSTM layer. The tested values for both variables were: 32, 64, 128, 256, 400, 500 and 750. In all cases the values were selected as they provide a suitable range of complexities.

- The activation function, here leaky ReLu was employed since it is commonly used in similar research. It is an improved version of the ReLU activation function. This type of activation function contains a parameter, Relu factor, that controls the angle of the negative slope. The applied Relu factor was set to 0.1, to provide a small negative slope.

- The optimisation function, here Adam, an improved version of stochastic gradient descent, was employed. This method was selected as it incorporates momentum, hence allowing it to escape from local minima during search. The learning rate parameter controls the step size of the weight updates at each iteration. For ANN, the tried learning rates were 0.001, 0.0001 and 0.00001. While for GRU and LSTM the tried learning rates were 0.001, 0.0001, 0.00001 and 0.000001. Smaller learning rates were preferred due to the tendency of RNN models to overfit quickly when the learning rate is large. The learning rates were chosen in both cases to find a suitable balance between effective learning of the training data and avoiding overfitting.

- Number of epochs. The number of epochs should be sufficient enough for the model to learn the training data. The tried number of epochs were 500, 750 and 1,000. These values should provide a suitable range of epoch values to enable the model to learn the different training data. However, to prevent further overfitting, if the training accuracy reached over 99%, training was terminated since the model learned the training data.

- The loss function selected was binary cross entropy since the output (Direction) is a binary variable, either 1 ('Up') or 0 ('Down'). This function computes the error between the predicted class and the actual class.

- Dropout rate, a dropout layer helps reduce overfitting by relying less on certain neurons. This is done by randomly selected neurons to be ignored during training. Dropout rate sets the ratio of how many neurons are to be ignored. The tried dropout rates of 0.2, 0.3 and 0.4 cover a suitable range for finding a suitable balance between overfitting and training performance.

- Mini batch size, that is the number of samples used in one training iteration. For GRU and LSTM the tested batch sizes were 16, 32, 64, 128 and 256. These values provide a wide range of training efficiencies to mitigate overfitting.

- For GRU and LSTM, the num_layers parameter determines the number of GRU or LSTM layers that are stacked on top of each other, enabling the network to learn more complex representations. The tested values of 1, 2, and 3 provide a range of complexities to explore.
- For GRU and LSTM an important hyperparameter to consider is the lookback. This hyperparameter determines the number of past time steps, in this case, trading days, that the network uses as input to predict the output for the current time step. This hyperparameter is essential for investigating Objective 3 since it aids in finding the optimal lookback value for stock market prediction. The values tested for the lookback parameter were 1, 3, 5, 10, and 15. These tested values cover a range of trading day lookback options, allowing for comprehensive exploration of the optimal lookback value for stock market prediction.

To develop sufficient model specifications a random search through the above options is constructed. This has been shown to be a suitable method of hyperparameter optimisation [36]. This is done through the following method:

1) Generate a random combination of parameters.

2) If the combination is already tested, go back to step 1.

3) Develop and train the model with these specific parameter choices.

4) Test model accuracy using 10-fold cross-validation.

5) If the accuracy is the best seen so far, save the model's parameters and update best validation accuracy.

6) Repeat for a number of times.

The above method should be sufficient to obtain suitable model parameters. To further improve the model's performance early stopping is performed to mitigate overfitting. This is performed by retraining the model and saving the model that obtained the best accuracy on the validation set. Therefore, validation accuracy is computed and checked after every epoch. Due to the stochastic nature of Neural Networks this process is repeated three times and the model that achieved the overall best validation set accuracy is saved.

4.5 Testing the Models

The performance of each model is primarily assessed using accuracy, a commonly employed metric in stock market trend prediction research, as demonstrated in previous studies [20, 22, 23, 33]. Accuracy represents the proportion of samples that are correctly classified, defined as:

$$ACC = \frac{Number of Correct Predictions}{Total Number of Predictions}$$

To achieve the objectives of the study, the accuracy of different models is compared, finding the best technical model, best news-based model, and the best hybrid model. This enables the identification of the best NLP representation for news. Moreover, a comparative analysis of these models is conducted to determine the best performing model, achieving Objective 1. Furthermore, Objective 3 is addressed by exploring the optimal lookback value for the GRU and LSTM models, as well as investigating the prediction horizon.

In order to assess the effectiveness of the proposed model, its accuracy score is compared with common stock market trend baseline scores, used in prior research [33]. These baselines are random baseline, where all potential outcomes are equally likely and keep trend baseline, where the prediction simply follows the previous trend. It is crucial that the performance of the proposed model exceeds these basic baselines.

To explore Objective 4, the best performing model is made to forecast the direction of various financial instruments. These instruments are carefully selected to gauge the model's predictive capabilities. The chosen instruments are four major US market indexes: Dow Jones Industrial Average (DJI), NASDAQ Composite (IXIC), Russell 2000 (RUT) and S&P 500 (GSPC), these indexes provide a comprehensive view of the US market performance. Additionally, four market sectors: Technology (XLK), Energy (XLE), Consumer Discretionary (XLY) and Healthcare (XLV), these represent significant segments of the economy. Finally, four individual stocks: Microsoft ($MSFT), JPMorgan Chase ($JPM), Coca-Cola ($KO) and Johnson & Johnson ($JNJ), representing influential companies in various large industries

In conclusion, this chapter has outlined the methodology employed to address the research objectives. The implemented plan is expected to yield valuable insights into the effectiveness of AI and news articles in financial markets. Moreover, by following the proposed testing strategy, reliable and accurate results can be obtained to assess the prediction performance of the AI model.

5 Evaluation

The objective of this chapter is to analyse the overall performance and effectiveness of the implemented ML models in predicting future stock market trends. The accuracies achieved using technical models, news-based models and hybrid models are compared, providing insights into the predictive power of news articles as features. Moreover, the best NLP representation for news is identified and the most suitable lookback period for GRU and LSTM models is established. Furthermore, a comparative analysis of the underlying methods is conducted. The section concludes by evaluating the best-performing model on its ability to predict a range of financial instruments, including large individual stocks, major US market indices and market sectors, using a new unseen time period from January 1st 2023 till May 1st 2023.

5.1 Time period and Baseline Accuracy

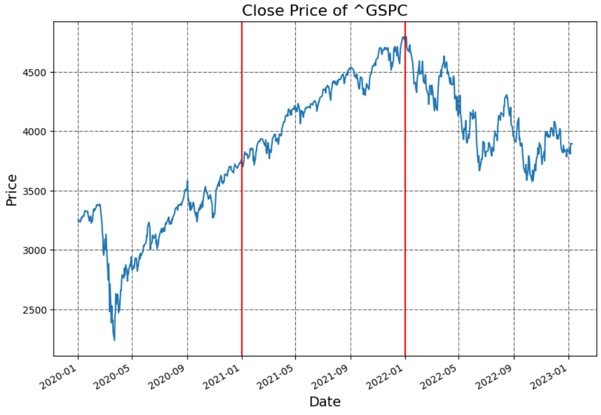


Figure 5.1 S&P 500 index stock chart for 2020-2023

As explained in Section 4.4.1, to evaluate the models various train-test splits are used, shown in Table 4.4. Additionally, the aim is to predict the direction of the S&P 500 index (GSPC) from market open till market close. The combined test years: 2020, 2021 and 2022 are visualised in Figure 5.1. The test years are relatively balanced, with 402 trading days classified as 'Up' and 354 trading days classified as 'Down'. Therefore, if all days were classified as 'Up' it would achieve an accuracy of 53.2%. In order to establish a benchmark for evaluating the performance of the different ML models, Keep Trend and Random Baselines are employed. In this case the Random model is similar to flipping a coin (2 outcomes) therefore it is assumed that the accuracy of this baseline is 50%. To generate the Keep Trend scores, the Keep Trend algorithm was implemented with lookback periods of 3, 5, 7, and 9 days. Since this baseline is rule-based, the train set is not required. However, the validation set, which consists of the last 10% of the train period is still utilised. For each test year the configuration that performed best on the validation set is selected and used to predict the test set. For the test period the Keep Trend accuracies are displayed in Table 5.1. These results indicate that surpassing an average accuracy of 50% can be considered as the benchmark to strive for by the models.

Table 5.1 Keep Trend Baseline Results

Test Year	Validation Accuracy (%)	Test Accuracy (%)
2020	46.3	49.8
2021	51.5	49.2
2022	49.3	50.9
Average	**49.0**	**50**

5.2 Technical vs News-Based vs Hybrid Models

This section highlights the results obtained by the technical models, news-based models and hybrid models. The best performing method in each case is identified and a comparison between the various models is conducted. This comparison provides valuable insights into the effectiveness of the developed models and their accuracy in predicting stock market trends.

5.2.1 Technical Models

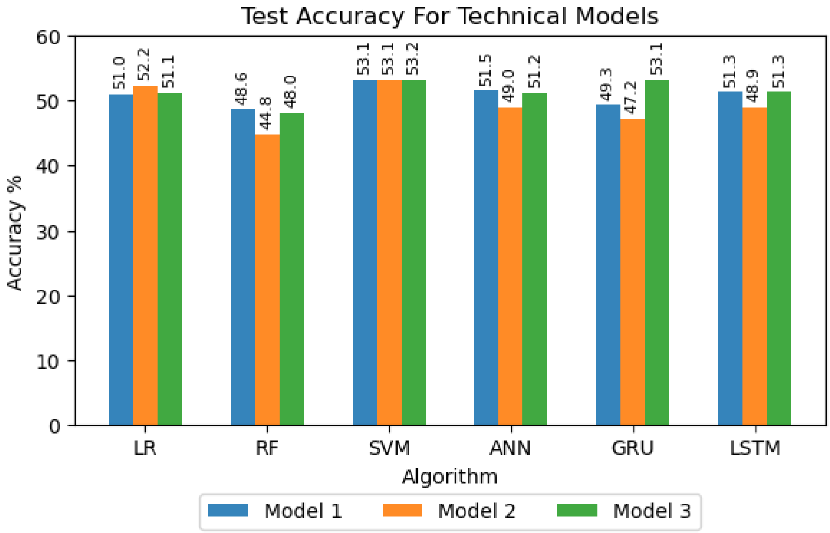


Figure 5.2 Bar chart representation of the accuracy of the Technical Models

The results for the three technical model variations, described in Table 4.1, are presented in Figure 5.2[1]. Most notably, SVM achieved the highest accuracy of 53.2% when using Model 3. With that said, SVM also performed well with the other models, achieving an accuracy of 53.1%. On the other hand, RF underperformed the 50% baseline with all variants. From the deep learning models, GRU obtained the highest accuracy when using Model 3. The highest average score across all the six ML methods was that of 51.3%, achieved by Model 3.

In conclusion, the results demonstrate that relying solely on technical indicators resulted in many ML methods underperforming the 50% threshold, with the exception of SVM. Moreover, SVM outperformed the other ML methods, this can be attributed to its distinct ability to generalise well and mitigate overfitting by finding the optimal separation between the classes.

5.2.2 News-Based Models

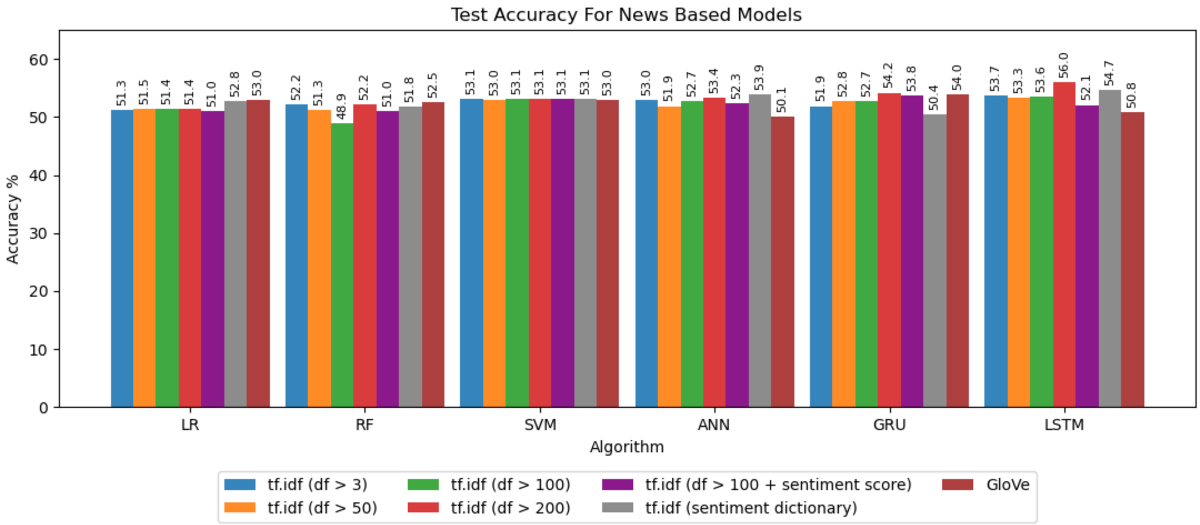


Figure 5.3 Bar chart representation of the accuracy of the News Based Models

In this section, we provide a comprehensive analysis of the performance of the models that solely utilise news article data. Figure 5.3[2] displays the results for the models that only use NLP representations. Interestingly, when using standard ML methods, the GloVe representation showcased slightly better performance compared to the TF–IDF representations. This can be attributed to its more dense representation. Among the standard ML methods: LR and SVM demonstrated the highest accuracy of 53%, while RF achieved an accuracy of 52.5%. On the other hand, when employing deep learning models TF–IDF representation outperformed GloVe representation. ANN obtained the highest accuracy of 53.9% with TF–IDF having a custom sentiment

[1] Full results can be found in Appendix Tables B.1,B.2 and B.3

[2] Full results can be found in Appendix Tables B.4, B.5, B.6, B.7, B.8, B.9 and B.10

dictionary, GRU achieved the highest accuracy of 54.2% with TF–IDF having df $>$ 200 and LSTM achieved the highest accuracy of 56% with TF–IDF having df $>$ 200 also. The highest average score across all the six ML methods was that of 53.4% and this was obtained by using TF–IDF representation with terms having df $>$ 200. Overall, the findings investigate Objective 2 and indicate that utilising news articles for stock market prediction is effective, as all of the models, except one, exceeded the 50% accuracy threshold. Furthermore, the majority of models exhibited notably improved accuracies in comparison to those obtained when utilising technical indicators.

5.2.3 Hybrid Models

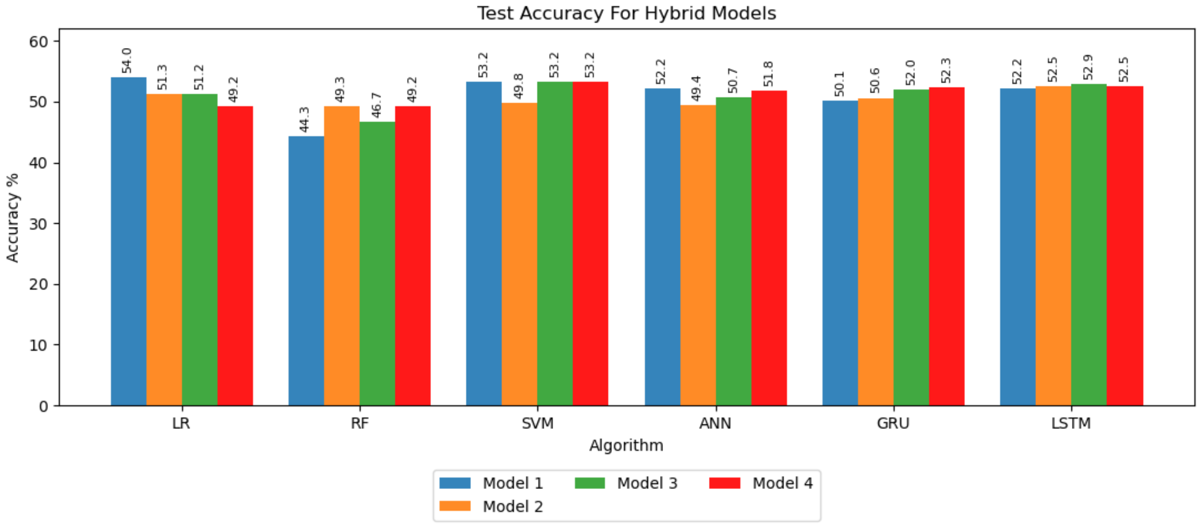


Figure 5.4 Bar chart representation of the accuracy of the Hybrid Models

The results for the four hybrid model variations, described in Table 4.3, are presented in Figure 5.4[3] . It is worth noting that Models 1 and 3 incorporate sentiment scores, whereas Models 2 and 4 utilise TF–IDF representation. From the standard ML methods, LR achieved the highest accuracy of 54% using Model 1. Meanwhile, among the deep learning methods, LSTM performed the best with an accuracy of 52.9%. It is important to note that all deep learning methods variants, except one, surpassed the 50% benchmark. On the other hand, RF failed to beat the benchmark with all configurations. Overall, the highest average score across all the six ML methods was that of 51.4% using Model 4, which utilises TF–IDF representation of the news data. This suggests that the news article terms indeed carry valuable information for stock market trend prediction. Furthermore, it is worth noting that excluding RF, most of the methods exceeded the 50% accuracy threshold, with many variations performing better than the technical-based models. In conclusion, the combination of technical

[3] Full results can be found in Appendix Tables B.11, B.12, B.13 and B.14

data and news data is shown to be a promising approach for achieving accurate stock market trend predictions.

In this section, we have examined the performance of the various technical models, news-based models and hybrid models. To summarise, many technical models underperformed the 50% benchmark, while models that incorporated news data generally surpassed this benchmark. Notably, models that based their prediction on solely news data exhibited the strongest performance. Among the tested methods, the best method was an LSTM network utilising TF–IDF with df $>$ 200, achieving an impressive average accuracy of 56% over the three year testing period. To conclude, these results strongly support our initial hypothesis, highlighting the advantage of incorporating news data sentiment for predicting stock market trends.

5.3 Determining the optimal lookback for RNN models

As outlined in the Methodology Chapter, various lookback values were evaluated during hyperparameter tuning for GRU and LSTM models. The other ML methods did not incorporate any lookback, as they do not posses the sequential information processing capabilities of RNNs. Based on the results obtained, the news-based model variant TF–IDF (df $>$ 200) produced the highest performance for both GRU and LSTM models. The GRU model used a lookback value of 3 trading days, while the LSTM model used a lookback value of 15 trading days. To further investigate the impact of the lookback value on model performance, an experiment is conducted using various lookback values while keeping the other hyperparameters constant. The results displayed in Table 5.2 indicate that the lookback value is a crucial factor in the model's performance. Interestingly, the GRU network achieved better results with shorter lookback values, while the LSTM network performed better with longer lookback values. These findings suggest that it is important to carefully choose the appropriate lookback value when developing a GRU or LSTM model for stock price prediction.

Table 5.2 Effect of Varying Lookback Values on GRU and LSTM Model Accuracies

Lookback value	**1**	**3**	**5**	**10**	**15**	**20**
GRU (Accuracy (%))	53.3	54.2	52.0	53.5	52.3	50.5
LSTM (Accuracy (%))	53.5	52.9	53.2	54.1	56.0	54.2

5.4 Determining the Best Performing Model

After evaluating the performance of all the model variations, the focus is to answer Objective 1, that is, determining the best ML model for stock market trend prediction.

The highest accuracy achieved on the test set was 56%, when using the LSTM network of the news-based model with news article data represented as TF–IDF vectors having df $>$ 200. The LSTM network is constructed with the following specifications: one LSTM unit with a hidden layer size of 750, a batch size of 128 with a lookback of 15 days, a hidden layer size of 128 for the fully connected layers, a dropout rate of 0.3 and a learning rate of 0.00001. Among the six ML methods that were evaluated, the LSTM network demonstrated superior performance. This can be attributed due to its exceptional ability to capture long-term dependencies and patterns in sequential data. This attribute is advantageous in the context of stock market analysis, as it enables the LSTM to effectively leverage trends and patterns to derive accurate predictions.

5.5 Predicting Financial Securities with the Best found Model

This section evaluates the predictive performance of the best found model on various financial assets over a new unseen time period, from January 1st 2023 to May 1st 2023. The training period covers trading days from January 1st, 2013 to December 31st, 2022. The model's predictive capabilities are tested on three main categories: major US market Indexes, market sectors ETFs and individual stocks. Figures 5.5, 5.6 and 5.7 showcase the model's accuracies on these three categories. From the findings it is evident that the model excels in predicting market indexes, with an average test accuracy of 57.7%. On the other hand, the model is considerably less accurate when predicting individual stocks, achieving an average accuracy of only 51.3% i.e. only 1.3% better than a random guess. Moreover, the model achieved an slightly above threshold average accuracy of 52% in predicting market sectors ETFs.

These results are deemed satisfactory as they have fulfilled the primary objective of forecasting major US market indexes using news data with an accuracy exceeding 50%. However, these findings suggest that forecasting individual stocks and specific sectors is a more challenging task. This is to be expected as these assets are generally more volatile and take into account more company-specific news. The difference in accuracy could also be due to the fact that the news articles, originating from the New York Times, may not cover specific industries or companies. For instance, the results suggest that the news data does not provide an accurate representation of the healthcare sector. Overall, the results indicate that the model performs well in predicting major US stock market indexes.

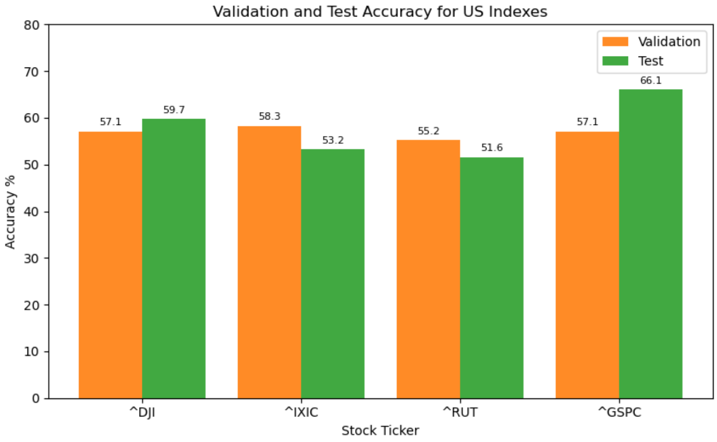


Figure 5.5 Bar chart representation of the accuracy on US Indexes

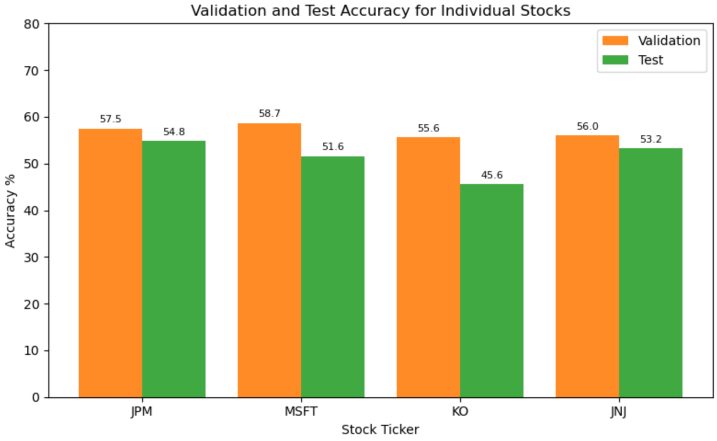


Figure 5.6 Bar chart representation of the accuracy on Individual Stocks

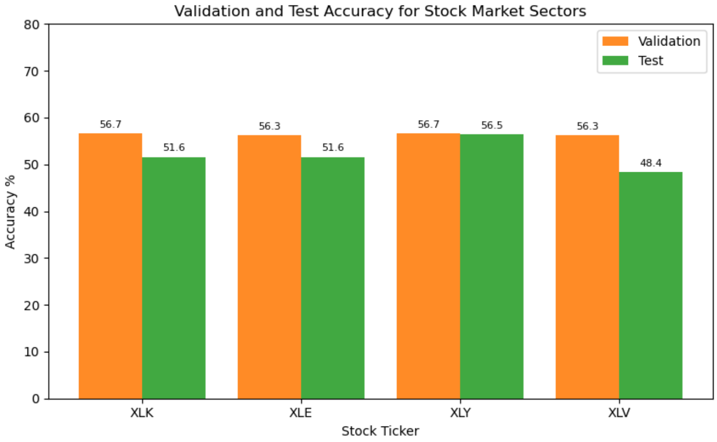


Figure 5.7 Bar chart representation of the accuracy on Market Sectors

5.5.1 Assessing the Prediction Horizon

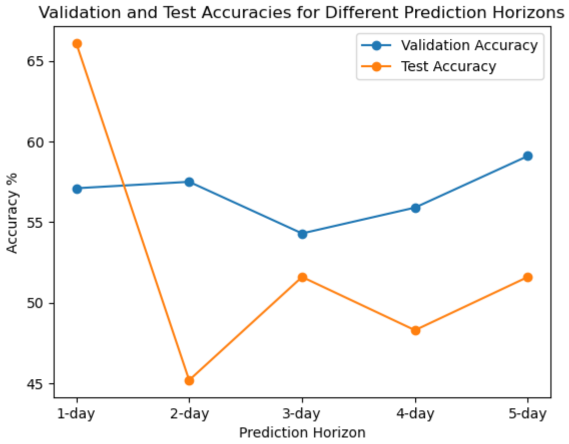


Figure 5.8 Results for different Prediction Horizons

Until this point the models were made to predict the direction from market open till market close (1-day), in this section we investigate the ability to predict longer horizons. Four additional time horizons were considered: 2-day, 3-day, 4-day, and 5-day. For instance, the 2-day horizon involves predicting the stock's direction from market open to the market close of the following trading day, the other time horizons are calculated similarly. The performance of the model across these prediction horizons is visualised in Figure 5.8. Notably, there is a considerable decrease in test accuracy observed after the 1-day prediction horizon. This suggests that the model is more suitable for predicting immediate market movements (from market open to market close). This is to be expected as news articles are quickly reflected in stock prices, this phenomenon is usually referred to as the information being 'priced in' by the market. Furthermore, the figure reveals that a higher validation accuracy does not necessarily correspond to a higher test accuracy, indicating the complexity of accurate stock market trend predictions.

5.5.2 Analysing the Performance of the Best found Model

Section 5.5 of the report evaluates the predictive performance of the model on various financial assets for the time period ranging from January 1st to May 1st, 2023. It achieved the highest accuracy of 66.1% when forecasting the S&P 500 (GSPC) index. During this test period, there were 62 trading days, with 34 classified as 'Up' and 28 classified as 'Down'. Hence, if we were to classify all trading days as 'Up', we would achieve an accuracy of 54.5%. The confusion matrix, displayed in Figure 5.9, reveals that the model demonstrated superior performance compared to this non-ML strategy. Additionally, Table 5.3 provides a breakdown of the results for the time period between

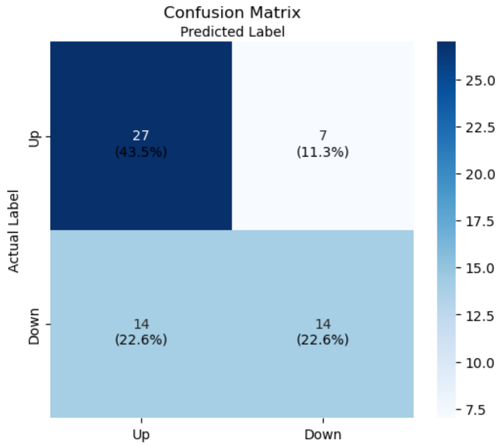


Figure 5.9 Confusion Matrix for S&P 500 Index Prediction

2020 and 2023 when using the LSTM network. It can be observed that the model managed to predict with an above threshold accuracy for all three years.

Directly comparing the results to previous studies is challenging due to the nature of the research problem. Various external factors greatly affect the predictive success of the model, such as the specific years predicted (2020-2023), the predicted financial asset, the prediction horizon and the news source utilised. However, it is worth noting that the findings are consistent with the expectations outlined in the Literature review. For example, TF–IDF representation is shown to be effective, the best performance was achieved by a deep learning model and SVM performed superior among the standard ML methods. Overall, these findings demonstrate that the model achieved the goal of predicting market trends over a long period of time.

Table 5.3 Model's performance on years 2020-2023

Test Year	Train Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
2020	63.3	57.2	55.7
2021	60.5	59,7	56.4
2022	97.7	60.3	55.9
Average	**73.8**	**59.1**	**56.0**

6 Conclusion

From the onset of the study, the primary goal was to design a model that can effectively forecast the stock market trend using news data. To collect the news data, the project utilised the NYT API, which was queried to extract articles published in the business section between January 1st, 2013, and May 1st, 2023. To achieve the desired model, four objectives were set up. The **first objective** was to identify the best ML model for stock market trend forecasting. The methodology used to accomplish this task was by first outlining possible useful features to aid in stock market trend prediction. These can be grouped into three categories: models that mainly use technical indicators, models that use news article data representation and models that use both technical data and news data. From each category, a number of variants were constructed. Additionally, each variant made use of six underlying algorithms, three standard ML methods: LR, RF and SVM, and three deep learning methods: ANN, GRU and LSTM.

Based on the findings, it was noted that in general, news-based models outperformed technical-based models. The best performing model was found to be a news-based model that utilised TF–IDF having df $>$ 200. Moreover, both standard methods and deep learning methods showed satisfactory performance with deep learning methods outperforming standard ML ones. Moreover, the best performing underlying method was LSTM. This model achieved an accuracy of 56% at predicting the direction from open till close of the S&P 500 Index over a 3-year test period and an accuracy of 66.1% when forecasting an additional 4 months between 1st Jan 2023 and 1st May 2023.

At the start of the project, the fundamental hypothesis was that news articles carry useful information that can be leveraged to accurately predict stock market trends. **Objective 2** sought to quantify the effectiveness of news article features at this task. To investigate this objective, various news article NLP representations were tested. These can be grouped into 3 main categories: representations that utilise the terms themselves (TF–IDF vectors), representations that capture the semantic meaning of words (GloVe) and sentiment scores that capture the negative or positive polarity of the news. To quantify the effectiveness, various ML models were trained on news data in relation to the S&P 500 direction for the time period between 1st Jan 2013 and 1st Jan 2020 and made to predict the S&P 500 daily direction for the following year. From the results achieved, it was noted that the performance of the ML models vary with the representation given. However, the best performance was achieved when using TF–IDF vectors that filter out less common terms (df$>$200). Moreover, most news-based models achieved above 50% accuracy, suggesting that news articles can indeed be used to effectively predict stock market trends.

To further investigate the effect of daily news articles on stock market trends, **Objective 3** was set up. This objective determines the most effective lookback window and prediction horizon. The GRU and LSTM networks have an important parameter that determines the number of past time steps the model considers when making a prediction. Therefore, identifying the optimal lookback period should provide insights into the number of past trading days that influence the current prediction. The findings displayed in Section 5.3 suggest that the optimal value is 3 trading days for GRU and 15 trading days for LSTM. Additionally, the optimal model found for prediction was an LSTM having a lookback value of 15. Hence this suggests that there is a need for a lookback for RNN models. On the other hand, when investigating the prediction horizon, it is observed that it is only accurate at predicting the direction from market open to market close. The accuracy decreases considerable beyond the 1-Day time horizon. This is in line with expectations since news stories might develop or change as time progresses and the stock market will reflect these new unseen changes.

The **final objective** of this study was to evaluate the predictive performance of the developed model across various financial assets. To accomplish this, the best found model (LSTM) was used to forecast the stock price direction, from market open till market close, of various financial assets over an unseen time period from January 1st 2023 to May 1st 2023. The assets tested included individual stocks, US market indexes, and market sector ETFs. Based on the results from Section 5.5, it was determined that the model performed superior when predicting major US indexes, such as, the S&P 500 Index and the Dow Jones Index. However, it was noted that the news obtained from NYT may not provide full coverage of specific industries or companies, which could have affected the model's accuracy in predicting individual stocks and sectors.

To sum up, this study emphasises the potential of utilising AI to forecast the stock market trend by incorporating news articles. The effectiveness of using news articles for stock market prediction was a key focus of this work, and the findings demonstrated that news data can be used to develop a useful predictive model. This highlights the importance of considering external factors in market forecasting. The study also highlights the process of developing ML models that lead to adequate predictive performance. Overall, this research provides valuable insights into the utilisation of AI techniques combined with news data for financial market forecasting.

6.1 Future Work

In our opinion, this project has successfully achieved its goals and objectives, however there are still opportunities for future work that can expand on the findings of this study. One potential avenue for exploration is the use of additional external features, such as social media sentiment, commodity prices and exchange rates. Incorporating multiple external features may lead to better performance, as demonstrated in a previous study conducted by Usmani et al.[23].

Additionally, the implemented model was found to be inaccurate in predicting individual stocks. This was noted to be a limitation of the news source used. Hence, further work can be conducted by using a number of mainstream news publications and filtering news articles to only select relevant articles to the company or sector relevant to the prediction. This approach has been shown to lead to more accurate predictions for individual stock in a similar study conducted by Zhong et al. [20].

Finally, one can focus on predicting shorter time horizons, similar to the work done by Gidofalvi et al. [26], such as 30 minutes after a breaking financial news article is released. This would investigate the immediate effect of news articles on stock market trends. In conclusion, the field of AI in stock market trend prediction is extensive and constantly evolving. This section has identified promising directions for future research.

6.2 Closing Remarks

In summary, this project accomplished its primary goal of designing a model for financial market trend forecasting using news sentiment. In brief, this was achieved by obtaining financial news article from NYT between 2013 and 2023, representing these textual content in a machine-friendly format and employing a ML method to learn the pattern between news sentiment and subsequent stock market trend. From the experiment conducted the best found model was an LSTM network utilised TF–IDF with df $>$ 200 representation. This model outperformed technical-based models and basic non-ML baselines. The model achieved an accuracy of 56% in predicting the direction of the S&P 500 Index over a 3-year test period (2020-2023) and an accuracy of 66.1% for an additional 4 months in 2023. The model limitations are that it is not suitable for individual stocks, mostly due to the broad nature of the news utilised. To conclude, this study highlights the effects of news sentiment on financial market trends and the potential of AI in leveraging news sentiment to develop effective trend forecasting models for financial markets.

References

[1] B. G. Malkiel, “Efficient market hypothesis,” in *Finance*, Springer, 1989, pp. 127–134.

[2] M. D. Godfrey, C. W. Granger, and O. Morgenstern, “The random-walk hypothesis of stock market behavior a,” *Kyklos*, vol. 17, no. 1, pp. 1–30, 1964.

[3] D. Duong, T. Nguyen, and M. Dang, “Stock market prediction using financial news articles on ho chi minh stock exchange,” in *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, 2016, pp. 1–6.

[4] W. Khan, M. A. Ghazanfar, M. A. Azam, A. Karami, K. H. Alyoubi, and A. S. Alfakeeh, “Stock market prediction using machine learning classifiers and social media, news,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–24, 2020.

[5] R. P. Schumaker and H. Chen, “Textual analysis of stock market prediction using breaking financial news: The azfin text system,” *ACM Transactions on Information Systems (TOIS)*, vol. 27, no. 2, pp. 1–19, 2009.

[6] G. A. Griffioen, “Technical analysis in financial markets,” 2003.

[7] I. Parmar *et al.*, “Stock market prediction using machine learning,” in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018, pp. 574–576. DOI: `10.1109/ICSCCC.2018.8703332`.

[8] O. Hegazy, O. S. Soliman, and M. A. Salam, “A machine learning model for stock market prediction,” *arXiv preprint arXiv:1402.7351*, 2014.

[9] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, “Stock closing price prediction using machine learning techniques,” *Procedia Computer Science*, vol. 167, pp. 599–606, 2020, International Conference on Computational Intelligence and Data Science, ISSN: 1877-0509. DOI: `https://doi.org/10.1016/j.procs.2020.03.326`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S1877050920307924`.

[10] S. Wang, “The prediction of stock index movements based on machine learning,” New York, NY, USA: Association for Computing Machinery, 2020, ISBN: 9781450376785. [Online]. Available: `https://doi.org/10.1145/3384613.3384615`.

[11] R. E. Wright, “Logistic regression.,” 1995.

[12] W. S. Noble, “What is a support vector machine?” *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[13] scikit-learn contributors, *Support vector machines*, `https://scikit-learn.org/stable/modules/svm.html`, 2021.

[14] Y. Liu, Y. Wang, and J. Zhang, “New machine learning algorithm: Random forest,” in *Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3*, Springer, 2012, pp. 246–252.

[15] A. Krogh, “What are artificial neural networks?” *Nature biotechnology*, vol. 26, no. 2, pp. 195–197, 2008.

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

[17] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.

[18] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[19] R. T. F. Nazário, J. L. e Silva, V. A. Sobreiro, and H. Kimura, “A literature review of technical analysis on stock markets,” *The Quarterly Review of Economics and Finance*, vol. 66, pp. 115–126, 2017.

[20] S. Zhong and D. B. Hitchcock, “S&p 500 stock price prediction using technical, fundamental and text data,” *arXiv preprint arXiv:2108.10826*, 2021.

[21] T. Kabbani and F. E. Usta, “Predicting the stock trend using news sentiment analysis and technical indicators in spark,” *arXiv preprint arXiv:2201.12283*, 2022.

[22] Y. Liu, J. Trajkovic, H.-G. H. Yeh, and W. Zhang, “Machine learning for predicting stock market movement using news headlines,” in *2020 IEEE Green Energy and Smart Systems Conference (IGESSC)*, IEEE, 2020, pp. 1–6.

[23] M. Usmani, S. H. Adil, K. Raza, and S. S. A. Ali, “Stock market prediction using machine learning techniques,” in *2016 3rd international conference on computer and information sciences (ICCOINS)*, IEEE, 2016, pp. 322–327.

[24] K. Sant Fournier, “Financial time series forecasting : From machine learning to deep learning,” 2018.

[25] M. Qasem, R. Thulasiram, and P. Thulasiram, “Twitter sentiment classification using machine learning techniques for stock markets,” in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2015, pp. 834–840.

[26] G. Gidofalvi and C. Elkan, “Using news articles to predict stock price movements,” *Department of computer science and engineering, university of california, san diego*, vol. 17, 2001.

[27] S. Baresa, S. Bogdan, and Z. Ivanovic, “Strategy of stock valuation by fundamental analysis,” *UTMS Journal of Economics*, vol. 4, no. 1, pp. 45–51, 2013.

[28] D. Jurevičienė and O. Ivanova, “Behavioural finance: Theory and survey,” *Mokslas-Lietuvos Ateitis*, vol. 5, no. 1, pp. 53–58, 2013.

[29] H. K. Baker and V. Ricciardi, “How biases affect investor behaviour,” *The European Financial Review*, pp. 7–10, 2014.

[30] Q. Li, T. Wang, P. Li, L. Liu, Q. Gong, and Y. Chen, “The effect of news and public mood on stock movements,” *Information Sciences*, vol. 278, pp. 826–840, 2014.

[31] R. W. Colby, *The encyclopedia of technical market indicators*. McGraw-Hill, 2003.

[32] J. Fang, Y. Qin, and B. Jacobsen, “Technical market indicators: An overview,” *Journal of behavioral and experimental finance*, vol. 4, pp. 25–56, 2014.

[33] J. G. de Araújo and L. B. Marinho, “Using online economic news to predict trends in brazilian stock market sectors,” in *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, 2018, pp. 37–44.

[34] C. J. Hutto and E. E. Gilbert, “VADER: A parsimonious rule-based model for sentiment analysis of social media text,” in *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, Ann Arbor, MI, 2014.

[35] B. Fazlija and P. Harder, “Using financial news sentiment for stock price direction prediction,” *Mathematics*, vol. 10, no. 13, p. 2156, 2022.

[36] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization.,” *Journal of machine learning research*, vol. 13, no. 2, 2012.

Appendix A News Data Visualisation

This appendix section focuses on further understanding the news article data. This is done through visualising the important news words for various years in relation to the daily movement of the S&P 500 index. The ranking of importance for each vocabulary term is calculated according to the cumulative TF–IDF score for the specific time period. This ranking is used to create weighted word cloud that visually represents the importance of the news terms.

Figure A.1 visualises the important news terms from the NYT articles headlines published between January 1st 2013 and January 1st 2023. Furthermore, these terms are categorised based on the daily direction of the stock market, ('Up' or 'Down'), and are displayed in Figure A.2 and Figure A.3. These Figures gives us a better idea of the most important terms in financial news articles and the relation with subsequent 'Up' and 'Down' movement.

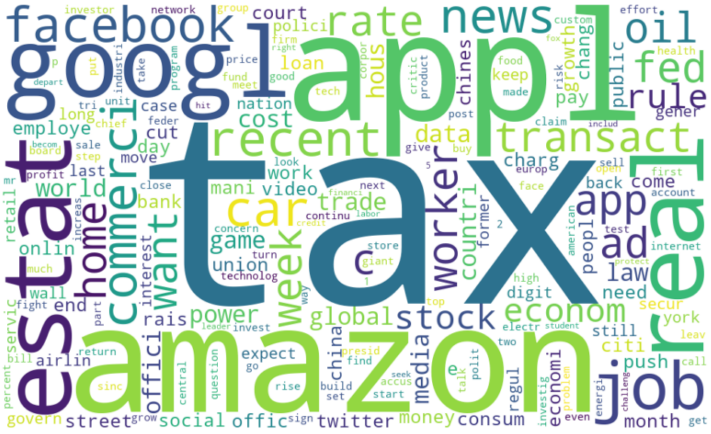


Figure A.1 News Terms for 2013-2023

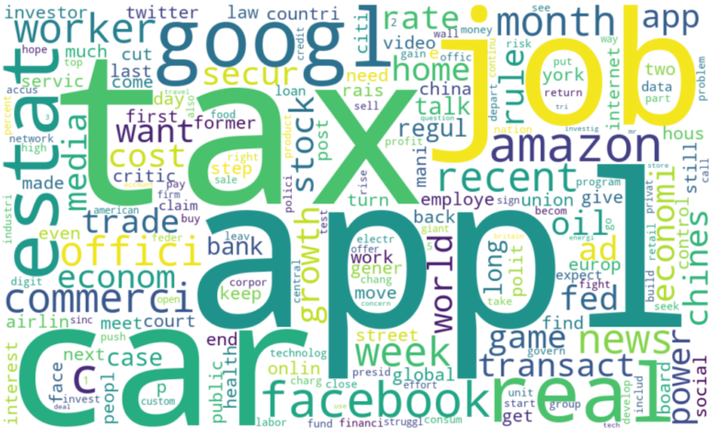


Figure A.2 News Terms before Up Movement (2013-2023)

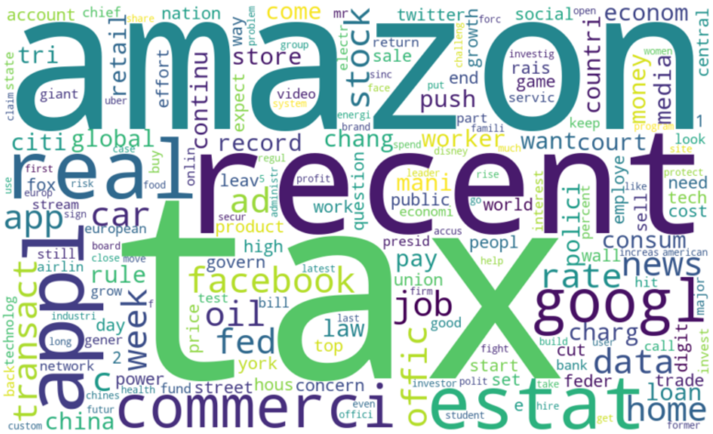


Figure A.3 News Terms before Down Movement (2013-2023)

Additionally, Figures A.4, A.5 and A.6 showcase the important terms for the 3 test years: 2020, 2021 and 2022 respectively. Furthermore, Figure A.7 showcases the term for the additional test period 1st Jan 2023 till 1st May 2023. These word clouds give us insights into the key news terms associated with the time period trends. For example, in 2021, the term ’vaccine’ held significant importance. The idea is that ML model should be capable of identifying the prominent terms that correlate with upward and downward movements. However as seen from these figures this is a challenging problem due to the evolving nature of news stories. Terms that were once considered important for a specific time period may lose their significance in other periods.

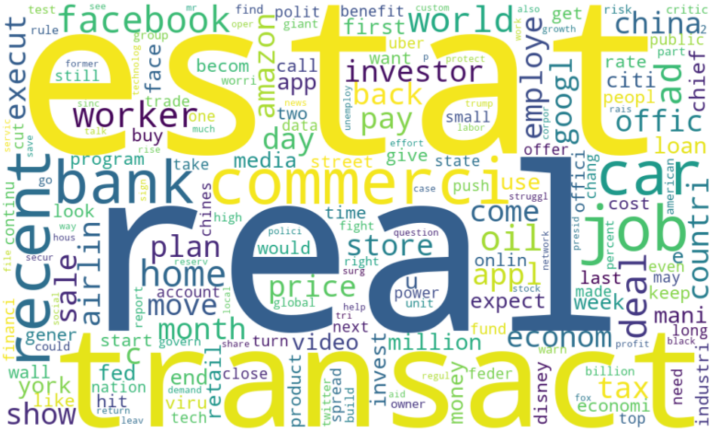


Figure A.4 News Terms for year 2020

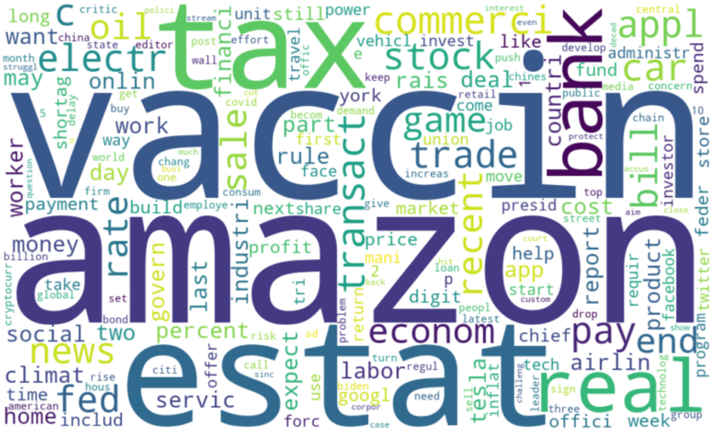


Figure A.5 News Terms for year 2021

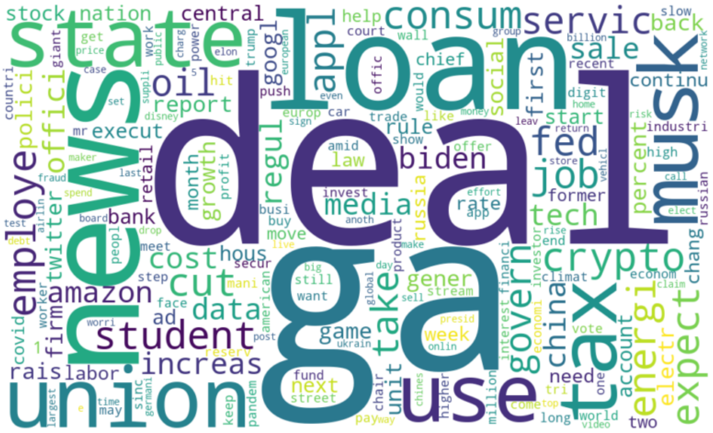


Figure A.6 News Terms for year 2022

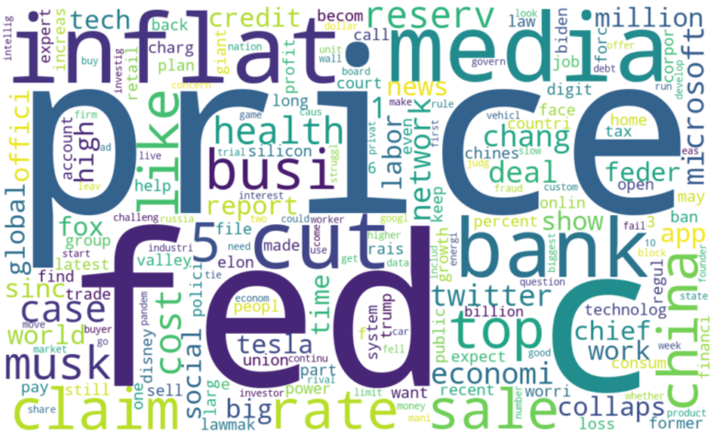


Figure A.7 News Terms for test period 1st Jan - 1st May 2023

Appendix B Detailed Experimental Results

We present the average results obtained over the three-year test period (2020, 2021, and 2022) that were used to evaluate the model variants in Section 5.2.

B.1 Technical-Based Models Results

The results for the technical-based model variations, described in Table 4.1, are presented in this section.

Table B.1 Technical Model 1

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	54.3	-	51.0
RF	100	51.9	48.6
SVM	-	54.4	53.1
ANN	60.2	59.2	51.5
GRU	57.4	58.2	49.3
LSTM	56.1	56.8	51.3

Table B.2 Technical Model 2

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	54.4	-	52.2
RF	100	59.4	44.8
SVM	-	54.4	53.1
ANN	54.8	59.5	49.0
GRU	55.2	56.7	47.2
LSTM	58.4	57.3	48.9

Table B.3 Technical Model 3

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	54.6	-	51.1
RF	100	50.5	48.0
SVM	-	54.4	53.2
ANN	57.7	59.0	51.2
GRU	78.3	57.6	53.1
LSTM	65.9	58.8	51.3

B.2 News-Based Models Results

The results for the news-based model variations are presented in this section.

Table B.4 News Model 1: Tf-Idf vectors with df $>$ 3

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	89.9	-	51.3
RF	100	52.9	52.2
SVM	-	54.4	54.1
ANN	71.7	56.8	53.0
GRU	70.5	58.0	51.9
LSTM	71.2	59.5	53.7

Table B.5 News Model 2: Tf-Idf vectors with df $>$ 50

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	86.2	-	51.5
RF	100	53.1	51.3
SVM	-	54.2	53.0
ANN	75.3	58.4	51.9
GRU	76.9	59.4	52.8
LSTM	65.8	58.3	53.3

Table B.6 News Model 3: Tf-Idf vectors with df $>$ 100

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	83.5	-	51.4
RF	100	51.6	48.9
SVM	-	54.4	53.1
ANN	65.3	57.6	52.7
GRU	73.9	59.0	52.7
LSTM	86.7	57.6	53.6

Table B.7 News Model 4: Tf-Idf vectors with df $>$ 200

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	78.2	-	51.4
RF	100	53.0	52.2
SVM	-	54.4	53.1
ANN	59.2	57.6	53.4
GRU	66.4	58.8	54.2
LSTM	73.8	59.1	56.0

Table B.8 News Model 5: Tf-Idf vectors with df $>$ 100 + sentiment score

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	83.2	-	51.0
RF	100	54.0	51.0
SVM	-	54.4	53.1
ANN	65.6	57.4	52.3
GRU	68.9	60.4	53.8
LSTM	88.6	58.3	52.1

Table B.9 News Model 6: Tf-Idf vectors with custom sentiment dictionary

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	57.4	-	52.8
RF	100	53.7	51.8
SVM	-	54.4	53.1
ANN	55.2	57.0	53.9
GRU	73.2	58.3	50.4
LSTM	61.2	56.5	54.7

Table B.10 News Model 7: GloVe

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	54.4	-	53.0
RF	100	55.1	52.5
SVM	-	54.4	53.0
ANN	57.8	59.1	50.1
GRU	64.5	57.5	54.0
LSTM	59.0	59.9	50.8

B.3 Hybrid Models Results

The results for the hybrid model variations, described in Table 4.3, are presented in this section.

Table B.11 Hybrid Model 1

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	54.3	-	54.0
RF	100	50.8	44.3
SVM	-	56.9	53.2
ANN	58.3	59.9	52.2
GRU	56.8	58.2	50.1
LSTM	55.2	58.4	52.2

Table B.12 Hybrid Model 2

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	78.0	-	51.3
RF	100	53.1	49.3
SVM	-	54.4	49.8
ANN	65.2	54.9	49.8
GRU	57.7	56.8	50.6
LSTM	59.3	56.7	52.5

Table B.13 Hybrid Model 3

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	54.3	-	54.0
RF	100	50.8	44.3
SVM	-	56.9	53.2
ANN	58.3	59.9	52.2
GRU	56.8	58.2	50.1
LSTM	55.2	58.4	52.2

Table B.14 Hybrid Model 4

ML Method	Train Accuracy %	Validation Accuracy %	Test Accuracy %
LR	77.9	-	49.2
RF	100	52.5	49.2
SVM	-	54.4	53.2
ANN	63.2	58.4	51.8
GRU	74.0	58.5	52.3
LSTM	83.2	57.5	52.5