CS310 Final Report 2022/23

Stock Market Trend Prediction Using High-Order Information of Time Series

Name: Ayo Toye
University ID: u2012437

Acknowledgements

I would like to take this opportunity to express my gratitude to the individuals and organisations who contributed to the successful completion of this project. First and foremost, I would like to thank Professor Peter Triantafillou for providing guidance, advice, and constructive criticism throughout the project. Without their support, this project would not have been possible.

I would also like to extend my appreciation to Yahoo Finance and Twitter for providing the necessary data to conduct this study. The data they provided was instrumental in the successful completion of this project.

TABLE OF CONTENTS

ABSTRACT

It is widely recognised that predicting stock prices is a difficult task due to the volatile nature of the stock market. Deep learning and machine learning techniques have recently made significant strides, and the results in this area are encouraging in respect of anticipating price movements. This project aims to combine the power of Auto Regressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM) neural networks and sentiment analysis to forecast prices accurately.

In the proposed approach, historical price data for a given stock is analysed using ARIMA modelling to predict the time-series behaviour of the stock, to account for seasonality and other external factors that could influence stock price. LSTM will also be used to identify patterns and trends that can be used to forecast future prices. The public sentiment toward the stock will then be ascertained by applying sentiment analysis towards tweets. Ways to incorporate this data into the model will be investigated to make the prediction model more accurate.

Python and a number of libraries, including TensorFlow and Textblob, will be used to complete the project. A variety of evaluation metrics will be utilised, such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), to assess the model's accuracy. The results will be compared at different stages to determine the effectiveness of each technique. The successful implementation of this project will offer a practical solution to the challenging task of stock price forecasting, allowing investors to make decisions based on accurate forecasts.

Keywords

ARIMA, LSTM, Sentiment Analysis, Natural Language Processing (NLP), Stock, Machine Learning

CHAPTER 1 – INTRODUCTION

This section outlines the reason for carrying out the project, what the project aims to achieve, and the approach taken to meet the aims.

Problem Statement

As the world's economies continue to grow, the importance of the stock market in modern society has continued to strengthen. When deciding whether to buy or sell stocks, investors usually look to the stock price as a key factor. Reliable stock prediction models have become an integral part of the decision-making process for investors in order for them to make educated investment choices and maximise returns. These models are able to give an investor an edge in a highly competitive market, however, their accuracy is debatable and there is no guaranteed method for predicting the stock market.

Technical analysis, which examines patterns and trends in historical stock prices, has traditionally been the foundation of stock prediction models since the 1800s since the Dow theory was introduced by Charles Dow [1]. However, this strategy has some drawbacks, since it ignores outside influences like news stories, social media sentiment, and world events that could affect stock prices. As evidenced by recent market fluctuations brought on by the COVID-19 pandemic [2] and geopolitical tensions, these factors can have a significant impact on stock prices.

There has been a growth of interest in machine learning methods to increase the accuracy of stock prediction over recent years. The ARIMA (Autoregressive Integrated Moving Average) and LSTM (Long Short-Term Memory) machine learning models are two of the most widely used models for time series analysis. LSTM is a type of recurrent neural network (RNN) that is well suited for processing sequential data, while ARIMA is a statistical model that is commonly used for time series analysis.

While stock prediction using ARIMA and LSTM has shown promising results [3], both methods have drawbacks. While ARIMA excels at modelling linear relationships in time series data, it may fall short when it comes to capturing more intricate and complex patterns [4]. On the other hand, LSTM is good at capturing complex patterns but is susceptible to overfitting and can be difficult to interpret [5]. Sentiment analysis is a potentially effective to provide insights into external influences on prices, however, the

accuracy of the analysis can be affected by a number of factors such as languages nuances and sarcasm by users of social media and reliability of sources.
One potential solution is to combine the positive aspects of both models to enhance stock price prediction accuracy. Modelling of both the linear and non-linear components of the time series data can be achieved by feeding the outputs of the ARIMA model's prediction into the LSTM model. Incorporating sentiment analysis may also offer insightful data about external variables that may have an impact on stock prices.

A thorough analysis of the methods and procedures currently employed for predicting stock prices will also be part of the project. The review will include a comparison of these methods' advantages and disadvantages. In addition, this dissertation will examine the ethical issues surrounding the use of predictive analytics in stock market investments. It will propose a technical implementation solution and make recommendations for the ethical and open application of these technologies.

Objectives and Aims

The primary objective of this dissertation is to create a hybrid stock prediction model by utilising the strengths of both ARIMA and LSTM modelling, and to include sentiment analysis to provide an understanding of the external factors that could affect stock prices. This main objective has been split into the following sub-objectives to divide the workload of the project:

1. A thorough review of a selection of common currently used methodologies for predicting stock prices, including technical analysis, statistical modelling, and machine learning techniques such as LSTM and ARIMA. The review will compare each option to the suggested hybrid model and examine the potential benefits and drawbacks.

2. Create a hybrid stock prediction model that combines LSTM's pattern recognition abilities with ARIMA's strengths in linear modelling. Using historical stock price data, the model will be trained and evaluated, and its accuracy will be compared to that of select other models such as The Random Forest Model [6], SVR [7], regression approaches [8] and other ANN techniques [9].

3. Conduct sentiment analysis of tweets using natural language processing and review the results of the analysis in conjunction with the stock price predictions to

examine the impacts that external factors can have on stock prices and determine whether this data is useful.

4. Evaluate the accuracy of the hybrid model with and without sentiment analysis. The results will be based on comparing whether on average, the percentage of trades made with sentiment analysis is higher than without it.

5. Review the ethical implications of using predictive analytics to trade on the stock market. In this dissertation a technical implementation of the hybrid model will be proposed and recommendations for the ethical application of the system will be proposed.

6. Describe prospective paths for future research in stock price prediction and how these techniques could be incorporated into the proposed system, such as the integration of other external variables and other machine learning techniques that have been recently developed.

In summary, the aim of this dissertation is to contribute to the development of stock price prediction models that are more accurate than existing models to help investors make more informed decisions. The proposed model has the potential to improve the accuracy of stock price prediction using machine learning techniques and by incorporating sentiment analysis into the investment process.

Approach

This section describes the approach taken to meet the objectives of this project, as well as providing justification for design and methodology decisions made after research into existing systems was conducted.

Choosing the data sources:

Yahoo Finance [10] was used as a data source, as it offers reliable and accessible historical stock data and has become a popular choice for academic research. There also exist Python libraries such as ‘yahoo_fin’ [11] which make it easy to programmatically gather and manipulate data.

Twitter was used for sentiment analysis as it provides real time information about people’s opinions making it useful in determining the collective mood of potential investors. The

‘Tweepy’ Python library [12] also integrates well with Twitter’s API, making it easy to extract tweets.

Selecting the prediction model:

A hybrid ARIMA and LSTM model was selected to make stock price predictions. These models were chosen as ARIMA is effective at modelling linear components whilst LSTM can capture non-linear dependencies effectively [13]. The selection process for the machine learning techniques used will be discussed further in chapter two.

Model Combination:

One of the challenges of this project was deciding the best way to combine the two models to maximise accuracy. By feeding the outputs of the ARIMA model into the LSTM model, the underlying trend and seasonality can be captured by ARIMA, and the more complex non-linear patterns can be captured by LSTM that may have been missed by ARIMA. By combining the models in this way rather than the other way around the accuracy is improved as the LSTM model can learn to make more accurate predictions by improving upon the underlying structure captured by the ARIMA model [14].

Data Extraction and Data Pre-processing:

Stock data from Yahoo Finance contains various metrics such as open price, close price, volume and range. For this project the close price and the date were used to form the time series, so these features needed to be extracted. The strategy for dealing with missing or redundant data was to merge the data. This was sufficient as Yahoo Finance is a high-quality data source so there was limited need for this strategy to be used.

The search filter for tweets will be the stocks ticker symbol. Using the ticker symbol rather than the name of the company ensures that the subject of the tweet will be about the company in the context of the stock market rather than in any other context that may be irrelevant.

Training the model:

TensorFlow is an open-source software library for numerical computation developed by Google for machine intelligence [15]. The library provides various functions that are useful for training machine learning models making it suitable for training the models used in this project.

The TextBlob library provides natural language processing functionality that is useful for determining sentiment of phrases. It will be used to categorise tweets as ‘positive’, ‘negative’ or ‘neutral’.

Displaying Results:

Stock price predictions are displayed as an overlay on a graph, as this allows the users of the application to visually compare predicted values with the historical stock data.
The sentiment analysis is displayed as a pie chart divided into ‘positive’, ‘negative’ and ‘neutral’, allowing users to easily draw conclusions about the general public’s opinion on the stock.

The system has been designed as web-based application using JavaScript and Flask since this allows convenient access to predictions, and by providing an interactive and user-friendly interface, the application can attract a wide audience.

Evaluating the model performance:

Mean Absolute Error (MAE) and Root mean squared error (RMSE) will be used as metrics to compare the performance of the model using only ARIMA with the hybrid model. This is standard practice in related work and enables a fair comparison to be made.

To determine whether sentiment analysis can have a positive impact on investment choices the system’s performance will be evaluated and compared under the following conditions:

- Without sentiment analysis.
- With sentiment analysis.
- With sentiment analysis one month into the future.
    - This condition allows us to explore whether incorporating predictive sentiment analytics algorithms would be beneficial to the model’s performance.

Legal, Social, Ethical and Professional Issues

Stock price prediction using machine learning techniques has attracted attention in recent years due to its potential to generate profit. However, it is associated with various legal, social, ethical and professional issues, which has made it necessary to establish guidelines and regulations to ensure that these systems are used in a responsible manner.

Legal Issues:

One of the main legal issues to consider with stock price prediction is insider trading. Insider trading is where confidential information is used to make a profit in financial market. It is essential to ensure that any data that the model uses for predictions is obtained legally, and that any trading decisions are made based on predictions are done in compliance with laws and regulations since machine learning and sentiment analysis may provide opportunities to identify patterns in stock price movements, and as a result provide insights that could be considered insider trading. [16]

Social Issues:

Stock price prediction can also have significant social implications. For example, the stock market financial crisis in 2008 resulted in widespread economic hardship. Machine learning prediction applications may also contribute to market volatility, which could adversely affect investor’s savings and pensions. Therefore, as these prediction models become more common there is a growing need to ensure that they do not create inequality and further hardship. [17]

Ethical Issues:

Prediction models also raise ethical concerns around accountability, bias and transparency. For example, when an algorithm produces a prediction that is inaccurate, who is then responsible for the resulting implications? In addition, it is important that programs and algorithms used are transparent and traceable. This way we can ensure that they do not include biases against a gender, race or other group that could lead to discrimination. [18]

Professional Issues:

Professional issues such as privacy and confidentiality are also raised. Data privacy regulatory bodies such as the General Data Protection Regulation (GDPR) require that an individual’s personal data is protected whenever it is used. Therefore, any data that is used for stock price prediction must be made anonymous and protected to ensure that it is kept confidential. [19]

CHAPTER 2 - BACKGROUND AND RESEARCH

Machine Learning in stock price prediction

This section covers a brief history of machine learning in stock price predictions, the theory behind the models chosen for this project and reasoning for choosing them.

ARIMA (Autoregressive Integrated Moving Average)

ARIMA is a well-established method for modelling time-series data. This section will provide an overview of how it works, the use of ARIMA in other models, how it has previously been used in stock price prediction and some of the pros and cons of the model.

The ARIMA Model

ARIMA is made up of three different components: Autoregressive (AR), Integrated (I) and Moving Average (MA). The model uses three different hyper-parameters: p, d and q. AR(p) represents the autoregressive component, where the current value of the time series is assumed to be dependent on the last ‘p’ values. MA(q) represents the moving average component, where the current value of the series is assumed to be dependent on the previous ‘q’ errors. I(d) represents the integrated component, where the series is differenced ‘d’ times, in order to make it stationary. The values of p, d and q should be selected such that the residual error is minimised.

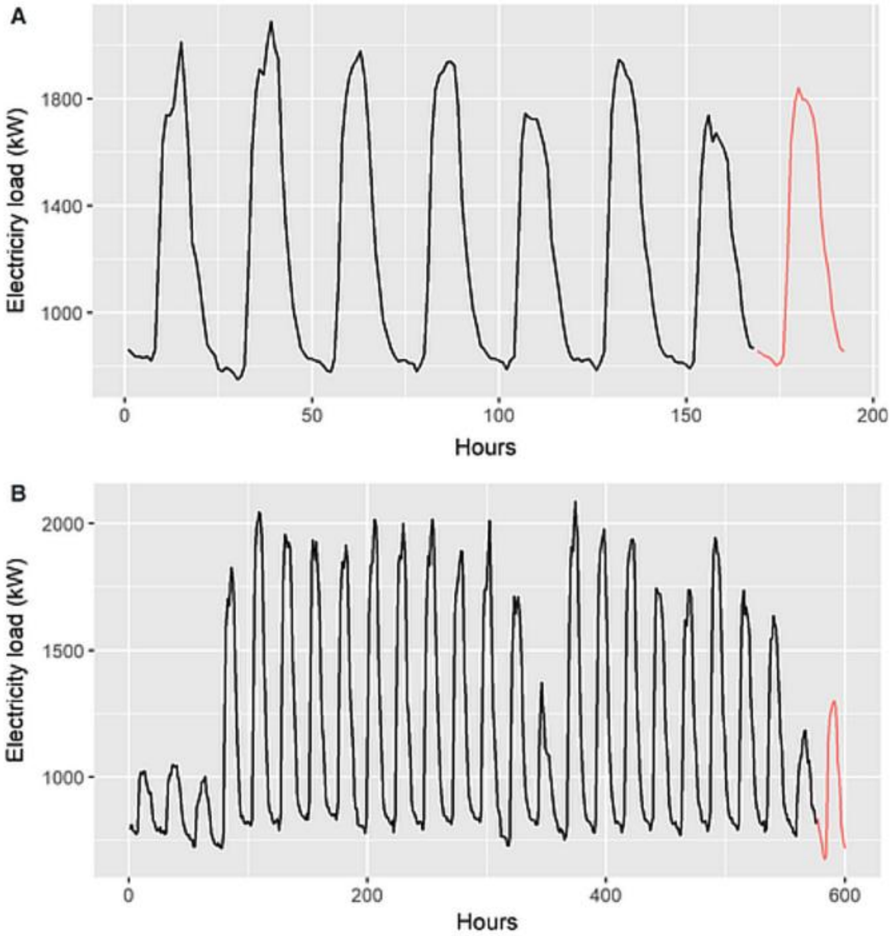
$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

[20] Figure 1 ARIMA modelling general equation

Predicted Yt = Constant + Linear combination Lags of Y (up to p lags) + Linear Combination of Lagged forecast errors (up to q lags)

The Use of ARIMA in Models Other Than Stock Price Prediction

ARIMA is most commonly used in finance, however there are many other applications. For example, it’s been used in electricity load forecasting for energy management buildings [21], the model was found to outperform other machine learning models such as SVM and achieved accurate results for forecasting over the period of one year.



[22] Figure 2 Predictions from Electricity load forecasting for energy management buildings.

ARIMA has also been used in economics to forecast macroeconomic variables such as GDP, inflation and unemployment rates. For example, Hsiao [23] used ARIMA and ANN methods to forecast Taiwan's GDP and found that the model produced good results in predicting short-term fluctuations. Ultimately ARIMA is a very versatile model that can be applied in various different contexts. However, it is important to assess its limitations and suitability to the specific task as with any other modelling technique.

ARIMA and Stock Price Prediction

ARIMA has been widely used in stock price prediction, with many research studies producing encouraging results. One of the earliest documented studies was conducted by Granger and Joyeux in 1980, who were using the model to forecast daily stock prices of five different US companies [24]. The results of the study indicate that ARIMA modelling performed well in predicting stock prices up to five days ahead.

Another study by Makdridakis and Fildres in (1995) compared the performance of various different time series models, including ARIMA, in predicting the stock prices of four different US companies [25]. They found that the ARIMA model outperformed the other models, including exponential smoothing, in terms of mean absolute error.

More recent studies have also indicated that ARIMA is effective at predicting stock prices, and there have been many reports of accuracies up to 89%.

Pros and Cons of ARIMA:

Pros:

- ARIMA is a well-established method for modelling time series data and has shown promising results in the field of finance.
- ARIMA can capture complex patterns in the data, such as seasonality, trends and cyclical behaviour.
- ARIMA is relatively simple to implement and does not require assumptions about the underlying distribution to be made.

Cons:

- ARIMA makes the assumption that the data is stationary, which is often not the case in real-world applications.
- ARIMA is a parametric model, thus it can only capture linear relationships between variables and is unable to capture non-linear relationships.
- ARIMA does not perform well when the data is noisy or there are many outliers.

Conclusion:

ARIMA is a model that is well established for modelling time series data and has been widely used in stock price prediction. It can capture complex patterns in the data and can be implemented relatively simply compared to other machine learning models. It has also been shown to outperform many of the other forecasting models when used for financial forecasting. However, it does have limitations, such as the linearity assumption it makes, and not being able to capture non-linear relationships. Nonetheless, many studies have demonstrated its effectiveness in stock price prediction making it a natural choice for this project.

LSTM (Long Short-Term Memory)

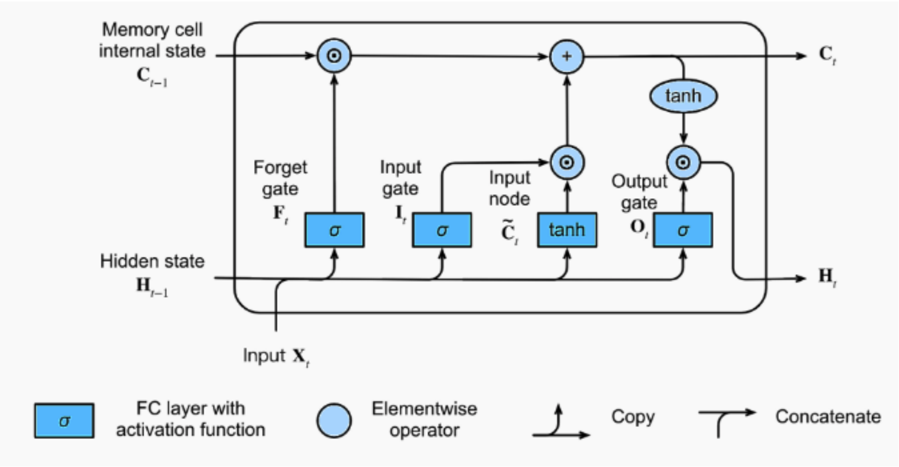
LSTM is another commonly used model in stock price prediction that has gained considerable attention due to its ability to model time series data and capture long term dependencies in data.

LSTM Model

LSTM is a type of recurrent neural network (RNN) that was designed to tackle the vanishing gradient problem in traditional RNNs. The vanishing gradient problem occurs when gradients become to small during back-propagation, which can slow down the rate at which the model can learn dependencies. LSTM uses a gating mechanism to control the flow of information through the network [26] to solve this problem.

An LSTM cell is made up of an input gate, a forget gate, an output gate, and a memory cell. The input gate controls the flow inputs into the memory cell and the forget gate controls the flow of memorised contents. The output gate controls the output of the LSTM cell, and the memory cell stores the internal state of the LSTM.

During the training phase, the LSTM network learns to adjust the weights of these gates in order to maximise performance. Performance is measured by its ability to capture long term dependencies and model the time series data effectively.



[27] Figure 3 Computing the hidden state in an LSTM model

LSTM in Stock Price Prediction

LSTM has been used extensively in stock price prediction and has shown promising results across various different studies. For instance, it was used to predict the stock prices of 30 companies listed in the German DAX index by Fischer and Krauss in 2018 [28]. The results of the study show that LSTM outperformed many other models including ARIMA and Support Vector Regression (SVR).

Similarly, a study by Khedher in 2020 used LSTM to predict the stock prices of five different companies in the US S&P 500 index. The results showed that LSTM outperformed other models such as random forest and gradient boosting, in accuracy and prediction error.

LSTM has also been combined with other techniques to try to improve accuracy of predictions such as sentiment analysis or ARIMA which has been shown to outperform ANN and ARIMA models.

Pros and Cons of LSTM

Pros:

- LSTM can effectively capture long term dependencies and model time series data effectively.
- LSTM can learn from both real time and historical data.
- LSTM can be used in various applications such as technical analysis and news sentiment analysis.

Cons:

- LSTM requires a large dataset to train a model effectively.
- LSTM can be prone to overfitting, especially on small datasets.
- LSTM can be computationally expensive and requires high end hardware to run efficiently.

Conclusion

LSTM has emerged as a powerful tool for stock price prediction, and its applications are vast. However, it has its drawbacks and like many machine learning models is not a panacea for stock price prediction.

Other Models

There are many different models that can be used for stock price prediction besides LSTM and ARIMA. This section contains an overview and comparison of four different models that could have potentially been used, as well as reasoning for choosing a combination of LSTM and ARIMA over them.

Gradient Boosting Regressor (GBR)

Gradient Boosting Regressor is an ensemble method that combines multiple decision trees to make predictions. GBR is able to handle non-linear relationships between features and target variables and is able to perform well on noisy data. It was used in a study on 50 US companies in the S&P 500, achieving high accuracy and the authors report a Mean Absolute Percentage Error (MAPE) of 5.57%.

Random Forest Regressor

Random Forest Regressor is another ensemble method that combines decision trees to make predictions. It can also handle non-linear relationships between features and target variables and can also perform well on unstructured or noisy data sets. A study by Lin and Chen used Random Forest to predict the stock prices of five Chinese companies, reporting a MAPE of 3.2%.

Convolutional Neural Networks (CNNs)

CNNs are a type of neural network commonly used in image recognition. However, they can also be used in time-series forecasting. A study by Wang in 2020 reported that when tested on 12 Chinese companies, the model gave an MAPE of 3.78%.

Support Vector Regression (SVR)

SVR is a supervised machine learning algorithm that can handle both linear and non-linear regression problems. SVR works by mapping the input data to a high dimensional space then finding a hyperplane that best separates the data points. A study by Ding and Xia in 2018 used an SVR model on 5 different companies achieving an MAPE of 4.92%.

These results may not be directly comparable due to the differences in datasets; however, the numbers are consistent with many other studies when tested on different data sets. Implementation of these different models from scratch has also been proven to be more complex than LSTM and ARIMA modelling making the results and factors that affect performance harder to interpret.

In a study by Kulshreshtha and Vijayalakshmi in 2020, the authors used a combination of LSTM and ARIMA to predict stock prices of different companies in the stock market. The study compared the performance of the combination model with the performance of an individual ARIMA model. The authors reported the following results when tested on the same dataset:

Model	MSE	MAPE
ARIMA	3.12	0.077
LSTM	3.03	0.009

[29] Table 1 Comparison of ARIMA with ARIMA and LSTM

This study provides an example of how a combination of ARIMA, and LSTM can potentially improve stock price prediction model accuracy, achieving a high MSE when compared with the other research papers that were reviewed for this project. These results combined with the simplistic implementation of LSTM and ARIMA were the reason for selecting a combination of these two models to use for this project.

Sentiment Analysis

TextBlob

Sentiment analysis is the process of determining the attitude or sentiment expressed in a piece of text. One popular tool used to conduct sentiment analysis is TextBlob, a python library that provides a simplistic interface for performing sentiment analysis on text data.

TextBlob works by using a combination of machine learning algorithms and rule-based heuristics to determine a polarity score to a piece of text. The polarity score indicates the sentiment expressed in the text, where the higher the score, the more positive the sentiment. A score of 0 represents a neutral sentiment.

To perform sentiment analysis using the TextBlob library, the first step is to tokenise the text into in individual phrases or words. Next, TextBlob uses a pre-trained machine learning algorithms to classify each token as positive, negative or neutral. The scores produced by each token are then aggregated to produce an overall polarity score for the piece of text [30].

Sentiment Analysis in Stock Price Prediction

In stock price prediction, sentiment analysis is mainly used to analyse the the tone expressed in news articles, social media posts, and other sources of information to predict how the market will respond to real world events. For instance, if news articles mentioning a particular company are generally positive, this would most likely cause the stock price of the company to increase, while negative news could trigger a decrease in stock price.

Several studies have shown that sentiment analysis is an effective tool for improving the accuracy of a stock price prediction model. For example, a study by Anshul Mittal from Stanford University on stock price prediction using tweet sentiment analysis [31] used tweets to predict public mood and used the predicted mood and previous days Dow Jones Industrial Average (DJIA) values to predict stock market movements. Their results showed that there was a correlation between certain sentiments, such as positive and calm with the DJIA values.

Another study by Barak and Modan in 2019 which used sentiment analysis to predict stock prices of the S&P 500 index. The authors used sentiment analysis on financial news articles to predict daily returns of the S&P 500 index and found that the sentiment score significantly improved the accuracy of the model when compared to a baseline model without sentiment analysis.

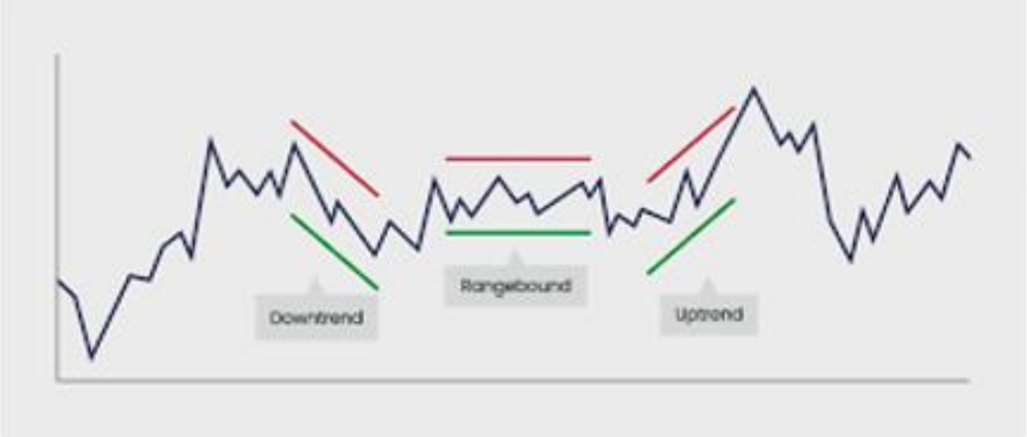
Many of the studies conducted on the effects of sentiment analysis on stock price prediction mention that it is not perfect tool, and its accuracy is largely affected by the quality of text, however, when used in conjunction with other information, it is a useful tool for improving accuracy.

Technical Analysis

Technical analysis is a method of evaluating securities and anticipating future price movements by analysing trends and statistical patterns in historical market data such as price and volume. The purpose of technical analysis is to identify patterns and trends that indicate future price movements that can be used to inform investment decisions.

Technical analysis was created based on the assumption that patterns in historical market data, in metrics such as price and volume, contains insights into future price movements. Technical analysts use different techniques such as looking at moving averages, chart patterns, and other technical indicators to identify cyclic behaviour and trends in this data. [32]

Moving averages are a common technique used in technical analysis to calculate the average price of a security over a set period of time. Chart patterns such as 'support', or resistance levels are also used to identify reverses in trends. Market data is also analysed alongside these chart patterns to identify potential price movements
using technical indicators such as the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD).



[33] Figure 4 Example of common technical analysis indicators

Technical analysis has been used to inform investment decisions for many years but its effectiveness in predicting future price movements is debated. Critics argue that technical analysis is based on the assumption that market data can predict future price movements, but this is not always the case. Some critics also argue that descriptive analysis can be subject to confirmation bias because analysts can interpret the data to support their own preconceived notions. In recent years, predictive machine learning models have emerged as potential alternatives to traditional technical analysis methods.

Machine learning models analyse market data using statistical algorithms to find patterns and trends that can be used to forecast future price movements. When comparing machine learning models to conventional technical analysis techniques the various advantages of machine learning algorithms are made clear. For instance, machine learning models can quickly and reliably handle enormous amounts of data, assisting in the identification of patterns and trends that are challenging for people to notice by eye. By learning from new data and adjusting to shifting market conditions, machine learning models can also gradually increase their accuracy over time.

Machine learning algorithms have proven that they can be very effective at forecasting stock values in numerous research reports when compared with traditional techniques and can even make them appear outdated. As an example, a 2019 study by Shen [3] created a machine learning model based on LSTM to forecast stock values for six different US companies. They found that in terms of forecast accuracy, LSTM models beat conventional time series models like ARIMA and conventional technical analysis methods and concluded that machine learning models are largely superior.

In conclusion, technical analysis is a method of evaluating securities and anticipating future price movements based on patterns in historical market data. Although traditional technical analysis methods have been used for many years, their effectiveness in predicting future price movements is debatable. In recent years, predictive machine learning models have emerged as good alternatives to traditional descriptive analytics methods that offer several advantages, such as the ability to quickly process large amounts of data and adapt to changing market conditions. Several studies have demonstrated the effectiveness of machine learning models in predicting stock prices, which can be a valuable tool for investors.

Technologies Used

Justification for the primary technologies used in this dissertation are provided in this section.

Python

Python is a popular programming language in data science and machine learning. It offers several powerful libraries and frameworks designed specifically for developing machine learning models. One such library is TensorFlow, an open-source software library developed by Google for building and training deep neural networks. Advanced machine learning models such as LSTM and ARIMA can be easily implemented using TensorFlow with Python, making them suitable for stock price prediction projects. Extensive libraries and tools facilitate data pre-processing, model training execution, and evaluation–critical tasks in any machine learning project.

TensorFlow

TensorFlow is a powerful library for developing machine learning models in Python, including neural network models such as LSTM. By using TensorFlow, developers can implement LSTM models to capture temporal dependencies in time-series data. ARIMA models can also be integrated with TensorFlow to capture the autocorrelation and seasonality in the data. The flexibility of the TensorFlow library should allow for easy integration with other Python libraries like Pandas for data pre-processing and Matplotlib for data visualisation. Using TensorFlow for the development of the LSTM and ARIMA hybrid model in Python offers a robust solution for stock price prediction making it the natural choice for this project.

Flask

Flask is a Python web framework that is used to create websites. With Flask, developers can create a web application that displays real-time stock price data alongside predictions generated by machine learning models. Flask is a lightweight and flexible framework that makes it easy to create custom web applications with features such as user authentication and interactive data visualisation. Additionally, Flask integrates well with other Python libraries, allowing users and developers to easily access and manipulate stock price data. The Flask framework offers a solution for displaying stock price predictions on a website with an intuitive interface.

CHAPTER 3 - METHODOLOGY

This section describes the how the system was implemented as well as justification for decisions made.

Data Flow Diagrams (DFDs)

The data flow diagrams (DFD) below show the overall design of the system.

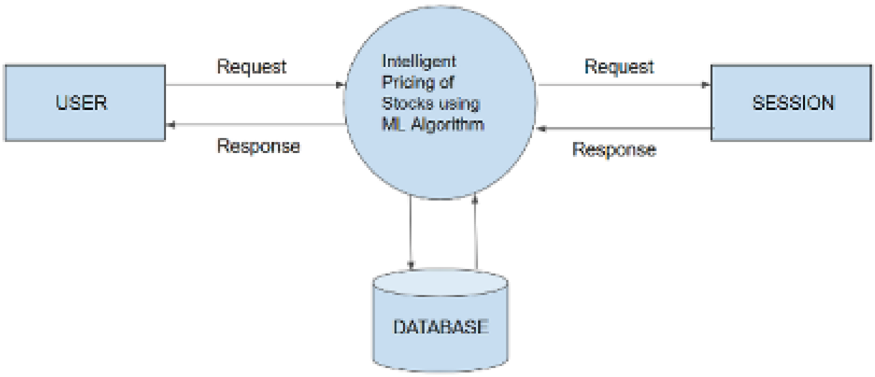


Figure 5 DFD Level 0 diagram of the system

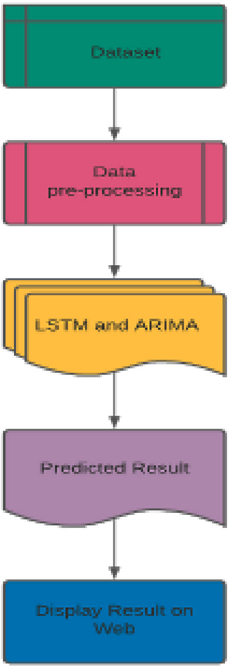


Figure 6 DFD Level 1 Design of the system

Data Collection

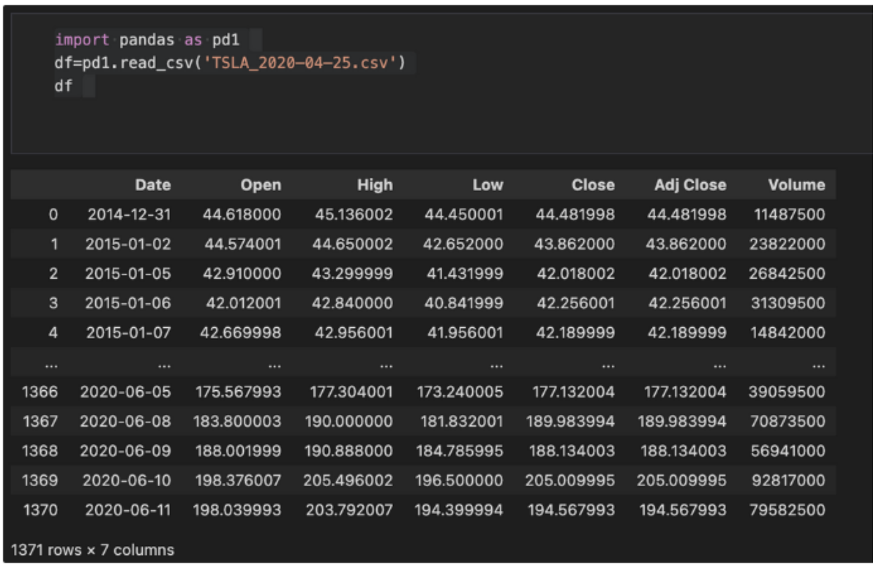
Data collection is an essential part of any data-driven project - such as this, and this section discusses how the data for the stock price prediction model and tweets sentiment analysis was collected.

Stock Data

This section describes how historical and real-time stock data was collected for training the models and to display to user.

Historical Data

The yfinance library was used to download the stock data for a given stock. The data was collected using the **`pandas_datareader`** module, and the **`pdr.get_data_yahoo()`** function was used to get the historical stock data. The **`start_date`** and **`end_date`** allow the programmer to specify the date range for which data will be extracted from.
The collected data was then pre-processed to fill in the missing values using the **`fillna()`** function, which replaced the missing values with the latest available price for each instrument. When testing this solution to missing values, it was found that it was rarely used, as the data provided by Yahoo Finance was of a high quality. This meant that the effect of missing data values would have had a very minor effect on the model, making this solution sufficient. Below is an example of the data captured for the ‘Tesla’ stock.



```
import pandas as pd1
df=pd1.read_csv('TSLA_2020-04-25.csv')
df
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-12-31	44.618000	45.136002	44.450001	44.481998	44.481998	11487500
1	2015-01-02	44.574001	44.650002	42.652000	43.862000	43.862000	23822000
2	2015-01-05	42.910000	43.299999	41.431999	42.018002	42.018002	26842500
3	2015-01-06	42.012001	42.840000	40.841999	42.256001	42.256001	31309500
4	2015-01-07	42.669998	42.956001	41.956001	42.189999	42.189999	14842000
...	...	...	...	...	...	...	...
1366	2020-06-05	175.567993	177.304001	173.240005	177.132004	177.132004	39059500
1367	2020-06-08	183.800003	190.000000	181.832001	189.983994	189.983994	70873500
1368	2020-06-09	188.001999	190.888000	184.785995	188.134003	188.134003	56941000
1369	2020-06-10	198.376007	205.496002	196.500000	205.009995	205.009995	92817000
1370	2020-06-11	198.039993	203.792007	194.399994	194.567993	194.567993	79582500

1371 rows × 7 columns

Figure 7 Data Collected For Tesla stock

The **`rolling()`** function was used to calculate the 20 and 100 days moving averages of the closing prices and the data was visualised using the **`matplotlib`** library. The collected data was then stored as a CSV file with the filename format "ticker_end_date.csv." The reason for storing the data in CSV format was to allow easy access for later use, allowing it to be manipulated and analysed.

The list of tickers was obtained from the "fortune500.csv" file containing a list of the Fortune 500 companies’ ticker symbols, and the data for each ticker was downloaded using the **`data.DataReader()`** function from the **`pandas_datareader`** module. The data was downloaded from January 1, 2010, to April 25, 2020. The reason for selecting this time period was so that predictions made by the model could be compared with reality over a short-term and long-term period. The Fortune 500 companies are generally well established and are often in the public eye. This subjects them to high levels of scrutiny by news outlets and individuals, resulting in them being tweeted about more frequently, which should increase the reliability of the sentiment analysis.

Live Data

The Yahoo Finance website was also used to fetch the live stock price for a given company. The **`requests_html`** and **`BeautifulSoup`** libraries were key in scraping the website's data. The **`HTMLSession()`** function was used to create a session object, and the **`get()`** method was used to fetch the web page's content. The **`str()`** function was used to convert the web page's content to a string, which was then split using the **`split()`** method to obtain the required data.

Overall, this method data collection was chosen because it is efficient and flexible. It allows the collection of historical and real-time stock data easily and accurately, and then saving it in a format that can be easily used in machine learning models or further analysed.

Tweets

The code for tweet collection connects to the MySQL database and uses the Tweepy library to extract tweets using Twitter's API. The extracted tweets are then cleaned using regex to remove any special characters and links. This step was important as it removes data that would have a negative effect on the outputs of the sentiment analysis due to the way TextBlob works. TextBlob's sentiment analysis method is then used to classify the sentiment of each tweet as positive, negative, or neutral. The cleaned tweets and their sentiment analysis are then stored in a Pandas DataFrame, and the positive, negative, and neutral tweets are printed along with the percentage breakdown.

An important decision to make was deciding the number of tweets that would be collected for each company. The last two hundred tweets were chosen as they provide a recent snapshot of the general sentiment and opinions of twitter users. In addition, using larger numbers of tweets may not necessarily provide more value, as older tweets may be outdated, and biases present in older tweets are avoided.

The supervisor for this project suggested that tweets only be collected from ‘verified’ sources as information was likely to be more reliable, however twitter recently started limiting the number of requests that can be made within a given time frame making collecting this data for each company challenging. Additionally, using tweets from verified accounts would significantly reduce the size of the data set as these sources tweet less frequently.

ARIMA Model Implementation

Training:

To train the ARIMA model, the historical stock data from the Fortune 500 companies was split into training and testing sets, initially with the training set consisting of the first 80% of the data and the testing set consisting of the remaining 20%. The order of the companies was first randomised to reduce bias as initially they were ranked by annual revenue. The train-test split was chosen as it allows for a sufficient amount of data to be used for the training phase whilst also leaving enough data to evaluate the performance effectively.

Before training the model, three conditions had to be met to ensure that the data was stationary:

- The variance must not be a function of time.
- The mean of the time series should be constant, not a function of time.
- Covariance of the i-th term and the (i+m)-th term must not be a function of time. [34]

These conditions can be ensured by using the Dickey Fuller test [35].

Hyper-parameter selection:

The ARIMA model was then trained on the training set using Python's statsmodels library. The Akaike information criterion (AIC) was utilised to identify the most optimal values for parameters (p, d, q) which is calculated by considering both goodness of fit and complexity. This method assesses the quality of a model, with a lower value signalling superior performance. Finally, out of all hyper-parameters tested, only those that yielded the lowest AIC values were considered.

In order to enhance the accuracy of the ARIMA model, a grid search was conducted. Its primary focus was to identify the most suitable values for the hyper-parameters from a set of candidate models. This approach entailed an examination of various ARIMA models with differing combinations of hyper-parameters - this involved adjusting the values of p, d, and q - in an effort to determine the most effective model. Once all permutations were trained and tested exhaustively against each other, the model with the highest accuracy was chosen.

Using the AIC is generally considered as a good starting point for hyper-parameter selection but may not always lead to the most optimal solution. A grid search is more likely to find the optimal hyper-parameters, however they are much more computationally expensive and time-consuming to perform. By first using the AIC, and then performing a grid search a sufficiently accurate model was able to be found without using resources excessively.

Testing:

Upon completion of training, the ARIMA model is then subsequently used for predictions on the testing dataset. Accuracy evaluation was then carried out examining multiple metrics including MAE, MSE and RMSE. The values of these metrics were stored so that comparisons could be made with the hybrid model once completed and with any later versions of the ARIMA model. The results of the model will be discussed and compared in the results section of this paper.

ARIMA and LSTM Hybrid Model Implementation

This section will cover the implementation and design justification for creating the hybrid ARIMA and LSTM model.

Combination:

To combine the ARIMA and LSTM models, the output of the ARIMA model must be used as an input to the LSTM. There were two main methods of doing this considered for this project: The first was by concatenating the outputs of the ARIMA model with the historical data, and the second was by using the ARIMA predictions as initial values for the LSTM sequence. The second method was chosen creating a sliding window effect to allow information relevance to be considered. Using all the historical data at once could potentially lead to an information overload, and not all of the historical data may be relevant to the current prediction task. This sliding window approach only used the most recent data points that are relevant to the current prediction, in an attempt to increase accuracy. Moreover, by feeding the data into the LSTM in this sequential manner, the model should be able to learn patterns in the data over time, be able to capture the temporal dependencies in the data better and it should reduce the likelihood of overfitting.

After deciding on how to combine the two models, the first stage of the implementation was to create a variable **`total_prediction_days`** which represents the number of days to predict into the future. Then, take the most recent **`total_prediction_days`** of the data and reshape it into a one-dimensional array. Next, a list called **`X_predict`** that holds the values of the sliding window. The sliding window was created by iterating through the inputs array, and for each iteration, appending a sub-array of size **`prediction_window_size`** to **`X_predict`** making the window size a hyper-parameter that could be adjusted. The data then had to be reshaped to fit the LSTM input dimensions.

After completing the relevant data processing and preparation, the architecture of the LSTM model needed to be defined. To do so, the Keras library was utilised in Python. The model itself consisted of four distinct LSTM layers, each with 50 units. Furthermore, two denser layers were also incorporated into this complex model; these layers included 25 and one unit respectively. The activation function used for the LSTM layers was the Rectified Linear Unit (ReLU) function, and the function used for the output layer was linear. A value of 0.2 was used for dropout regularisation at each layer. Typically values for dropout regularisation range from 0.2 to 0.5, where higher values can lead to overfitting.

Next, the model was compiled using the Adam optimiser with the mean squared error (MSE) loss function. The Adam optimiser is an optimisation algorithm used for stochastic gradient descent and was chosen because of its efficiency and ease of use when compared to other optimisers such as RmsProp.

Upon compilation of the model, the training phase ensued. The training data was fed into the model in batches systematically, and the weights of the model were adjusted iteratively to minimise the MSE loss function. The training data was first segregated into training and validation sets using the 99/1 (default) split. The model was then fit on the training data, and the validation data was used to monitor the performance of the model and avert overfitting.

An essential aspect of the training phase involved utilising a sliding window technique to supply data to the LSTM model. The chosen window size specified in the code was 60, implying that the succeeding closing price is predicted by feeding into the model, 60 consecutive prices close to each other. The window size was chosen to be small enough to capture short-term trends while being large enough to avoid overfitting. Finally, after the training had been completed, the model was used to make predictions on future close prices for the next three years.

Sentiment Analysis

The sentiment analysis was done by classifying the polarity of the TextBlob object (The tweet). If the polarity was greater than 0, the sentiment was positive; if the polarity was equal to 0, the sentiment is neutral, and if the polarity was less than 0, the sentiment is negative. After such separation was completed, percentages were calculated to assess which category most posts align themselves with.

Once the sentiment analysis was conducted, the tweet data is stowed in a Pandas DataFrame. This information can be put into a MySQL database utilising PyMySQL or kept in HDFS via InsecureClient from the HDFS library for subsequent presentation on the website.

While this approach had its merits, it did come with a drawback: the absence of an effective mechanism for detecting sarcasm or irony. That being said, developing such a tool is easier said than done. Most language interpreting algorithms often fall short in realising contextual cues, so to detect these subtleties would require a highly sophisticated algorithm or approach. Nonetheless, researchers are actively exploring various avenues within natural language processing to tackle this challenge and eventually integrate sarcasm/irony detection into sentiment analysis algorithms moving forward.

Website Design

This section will cover the features of the application, and justification for design decisions. The website was created using JavaScript and Flask, the design was kept simple to prevent longer loading times and templates were used for features such as the search bars and website headers.

Landing Page

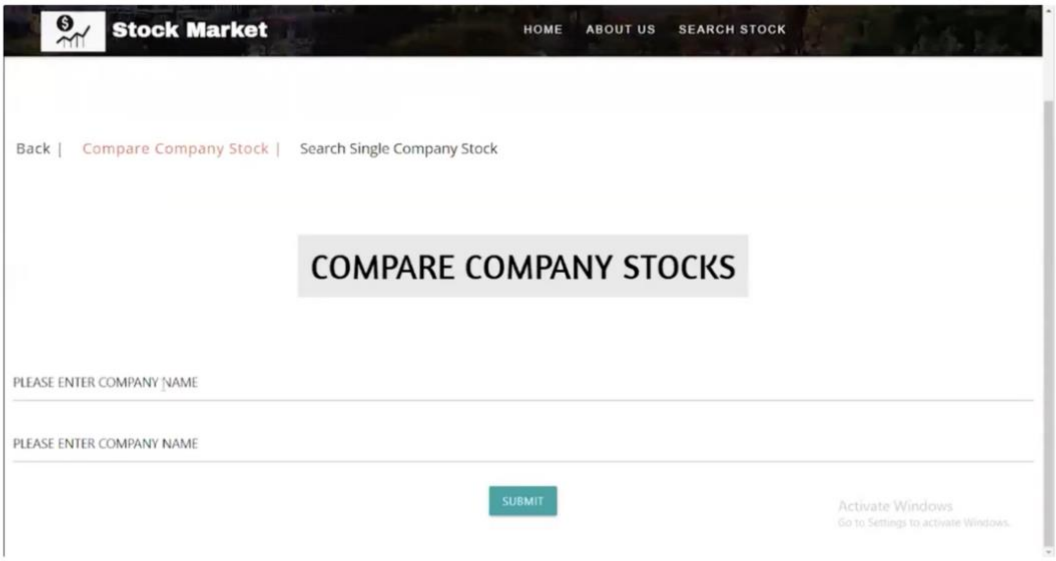
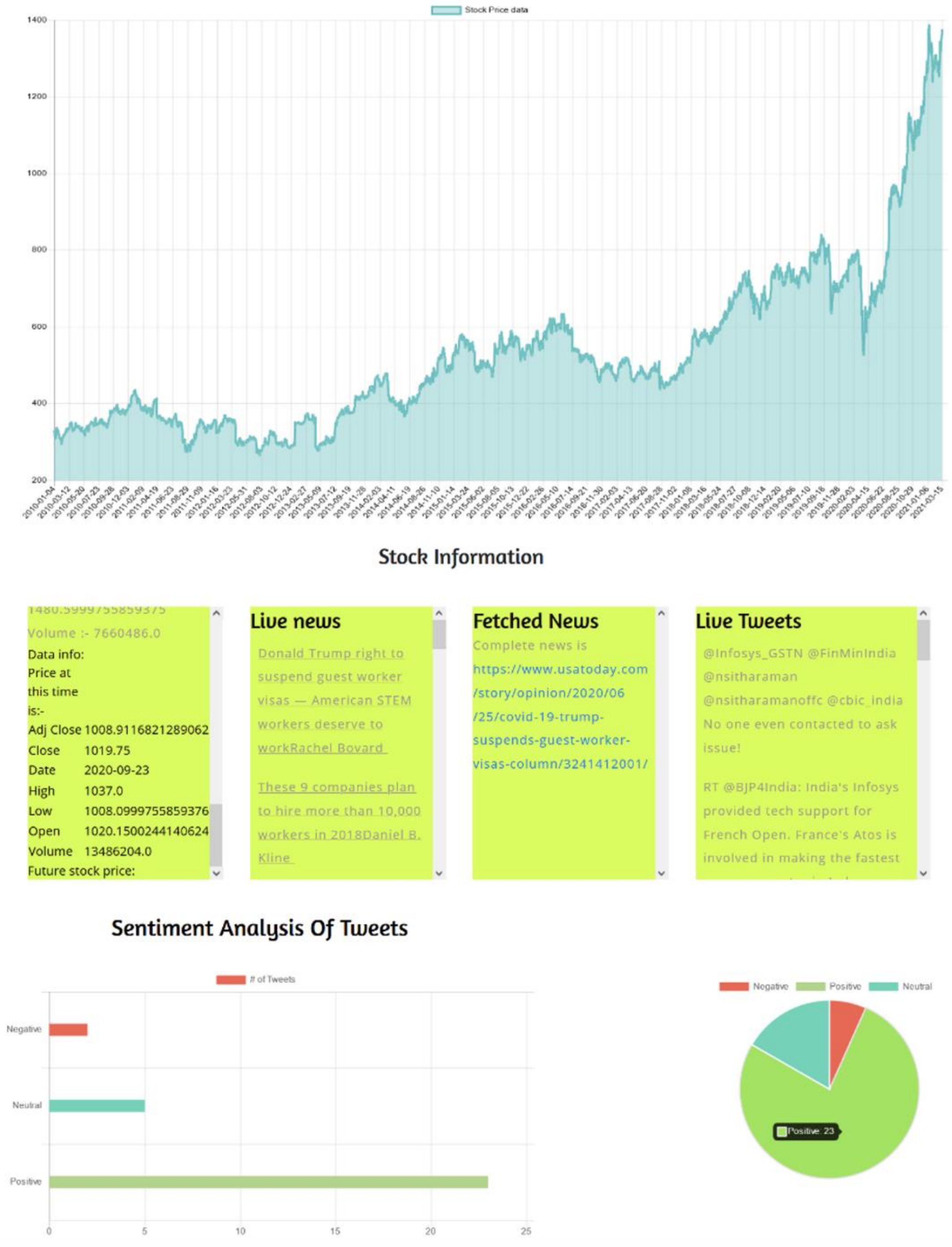

Stock Market
HOME
ABOUT US
SEARCH STOCK
Back |
Compare Company Stock |
Search Single Company Stock
COMPARE COMPANY STOCKS
PLEASE ENTER COMPANY NAME
PLEASE ENTER COMPANY NAME
SUBMIT
Activate Windows
Go to Settings to activate Windows.

Figure 7 Landing Page of the website

The screenshot above shows the landing page for the stock price prediction website. It allows the user to select from two options, the first being searching for two different companies and displaying them on the same graph as an overlay for comparisons to be made easily if the user is deciding between investing in two stocks, and the second option is to search for an individual stock.

Current Data



Stock Information

1480.5999755859375
Volume :- 7660486.0
Data info:
Price at
this time
is:-

Adj Close	1008.9116821289062
Close	1019.75
Date	2020-09-23
High	1037.0
Low	1008.0999755859376
Open	1020.1500244140624
Volume	13486204.0

Future stock price:

Live news

Donald Trump right to suspend guest worker visas — American STEM workers deserve to workRachel Bovard

These 9 companies plan to hire more than 10,000 workers in 2018Daniel B. Kline

Fetched News

Complete news is
https://www.usatoday.com/story/opinion/2020/06/25/covid-19-trump-suspends-guest-worker-visas-column/3241412001/

Live Tweets

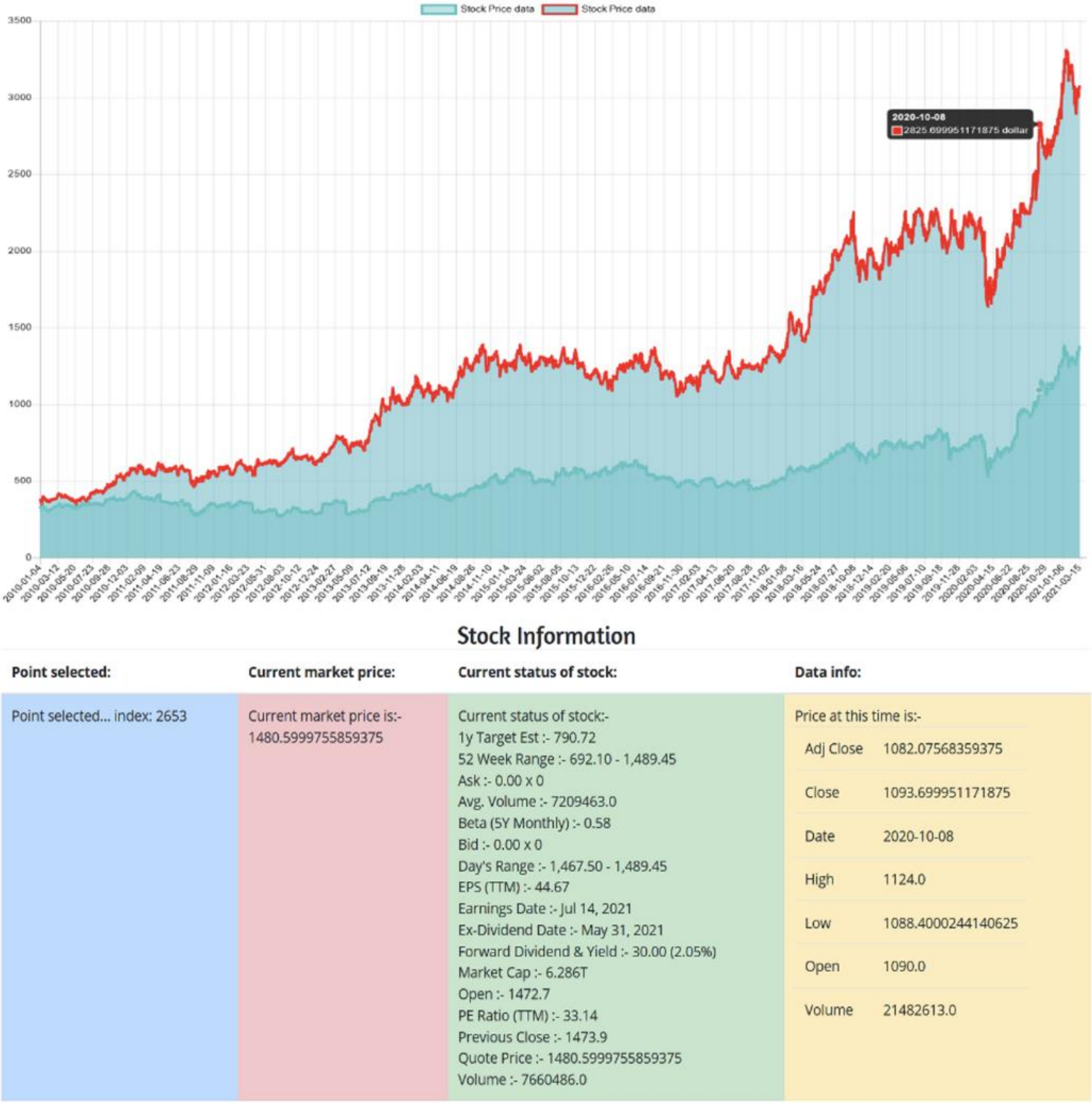
@Infosys_GSTN @FinMinIndia @nsitharaman @nsitharamanoffc @cbic_india No one even contacted to ask issue!

RT @BJP4India: India's Infosys provided tech support for French Open. France's Atos is involved in making the fastest

Sentiment Analysis Of Tweets

Figure 8 Current information page

This page will be displayed after a user has search for an individual stock. The screenshot above shows the current time series graph of the stock, other relevant information such as volume, live tweets, and a sentiment analysis breakdown summary of the tweets.



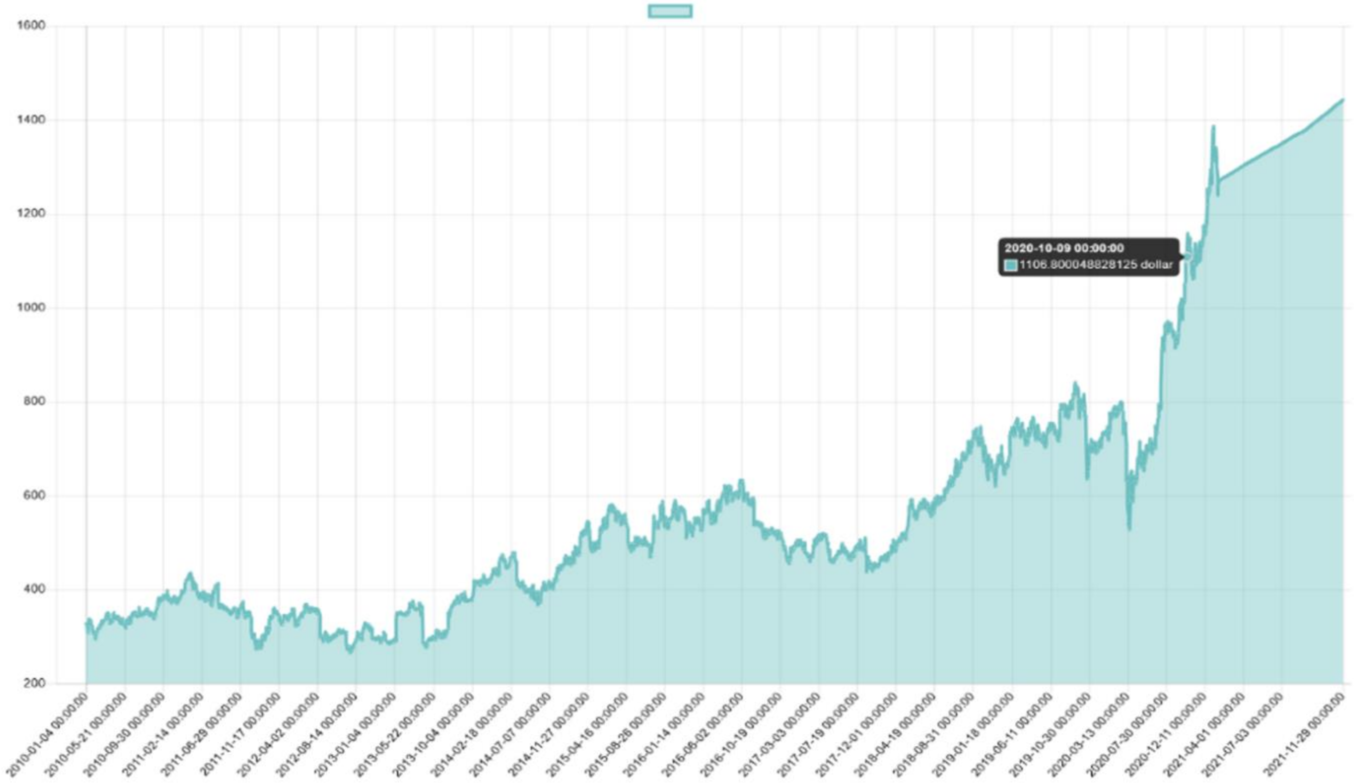
Stock Information

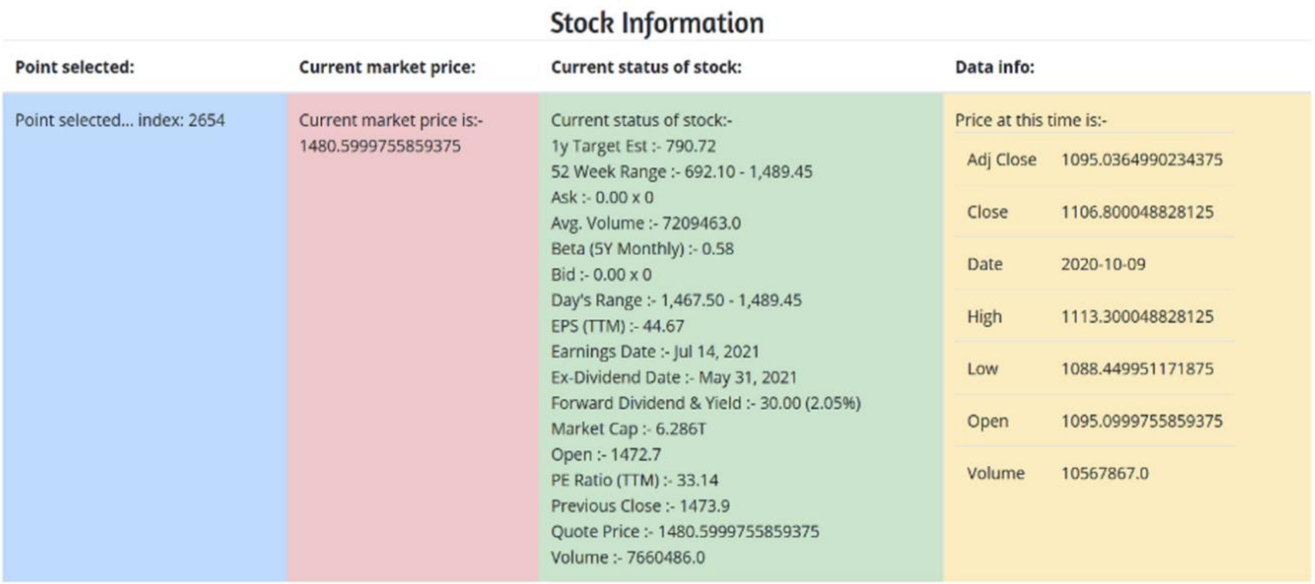
Point selected:	Current market price:	Current status of stock:	Data info:
Point selected... index: 2653	Current market price is:- 1480.5999755859375	Current status of stock:- 1y Target Est :- 790.72 52 Week Range :- 692.10 - 1,489.45 Ask :- 0.00 x 0 Avg. Volume :- 7209463.0 Beta (5Y Monthly) :- 0.58 Bid :- 0.00 x 0 Day's Range :- 1,467.50 - 1,489.45 EPS (TTM) :- 44.67 Earnings Date :- Jul 14, 2021 Ex-Dividend Date :- May 31, 2021 Forward Dividend & Yield :- 30.00 (2.05%) Market Cap :- 6.286T Open :- 1472.7 PE Ratio (TTM) :- 33.14 Previous Close :- 1473.9 Quote Price :- 1480.5999755859375 Volume :- 7660486.0	Price at this time is:- Adj Close 1082.07568359375 Close 1093.699951171875 Date 2020-10-08 High 1124.0 Low 1088.4000244140625 Open 1090.0 Volume 21482613.0

Figure 9 Current information for multiple stocks

The screenshot above shows the current data when two different stocks are compared. Contextual information is displayed below the time series graph for the selected stock which can be changed by clicking on the other time series.

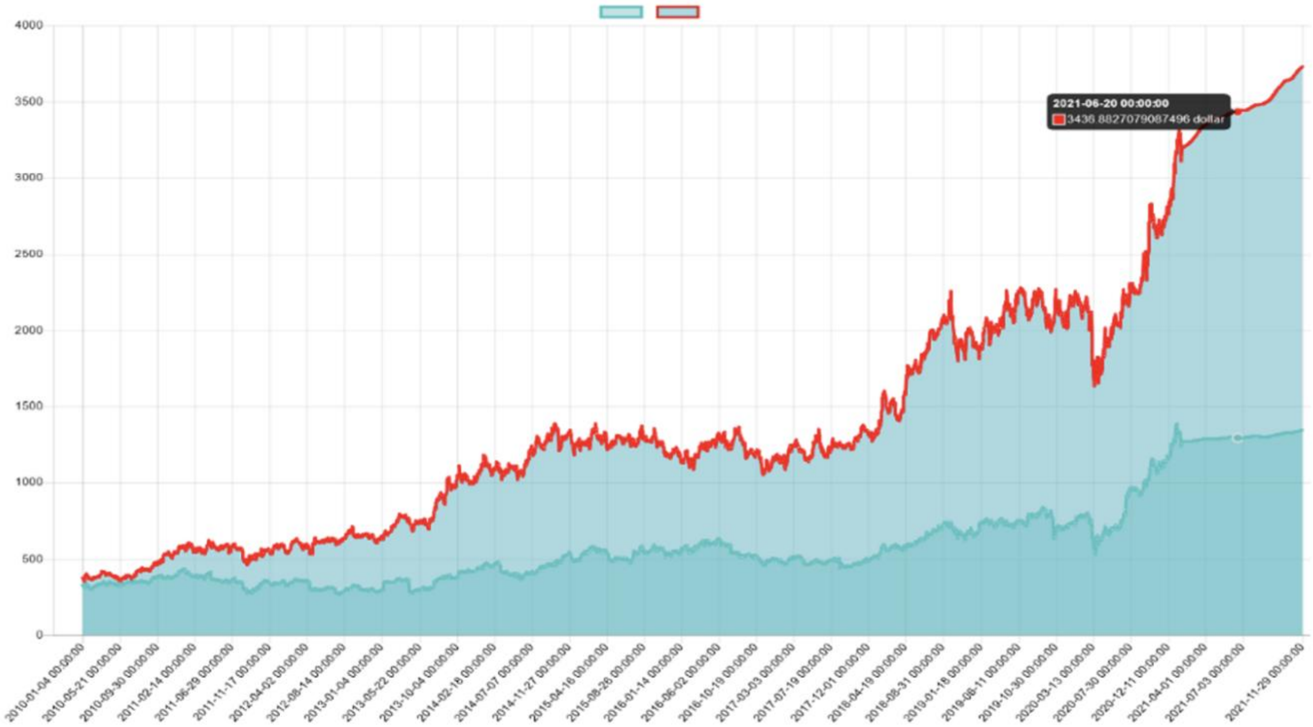
Predictions

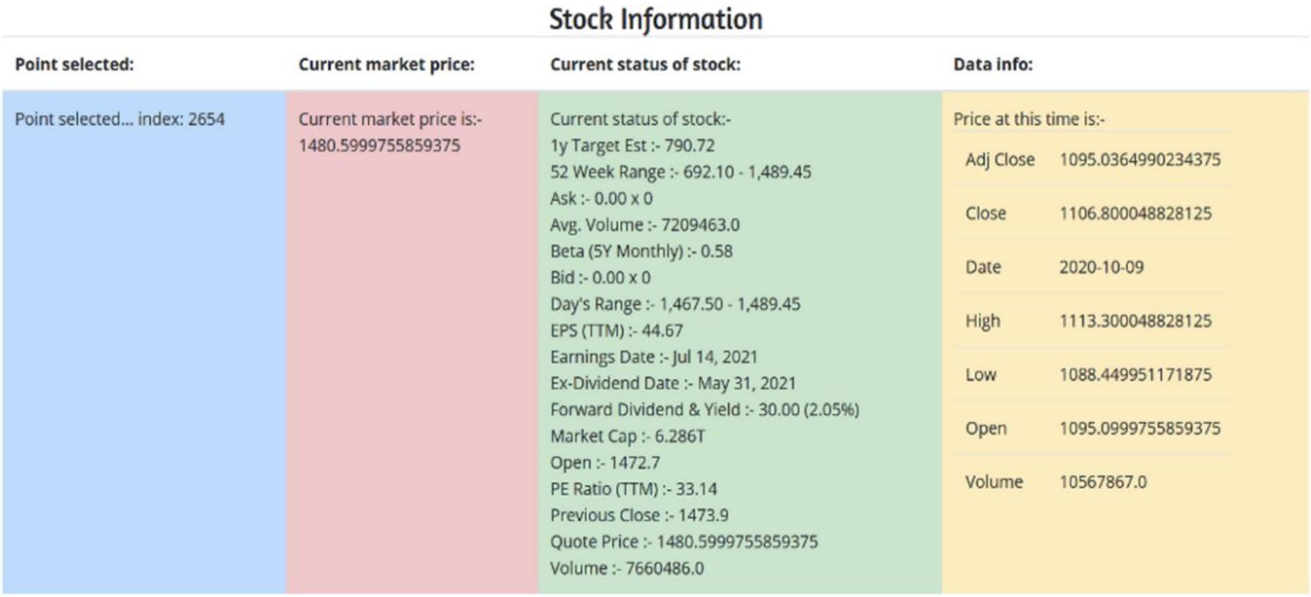




Stock Information

Point selected:	Current market price:	Current status of stock:	Data info:
Point selected... index: 2654	Current market price is:- 1480.5999755859375	Current status of stock:- 1y Target Est :- 790.72 52 Week Range :- 692.10 - 1,489.45 Ask :- 0.00 x 0 Avg. Volume :- 7209463.0 Beta (5Y Monthly) :- 0.58 Bid :- 0.00 x 0 Day's Range :- 1,467.50 - 1,489.45 EPS (TTM) :- 44.67 Earnings Date :- Jul 14, 2021 Ex-Dividend Date :- May 31, 2021 Forward Dividend & Yield :- 30.00 (2.05%) Market Cap :- 6.286T Open :- 1472.7 PE Ratio (TTM) :- 33.14 Previous Close :- 1473.9 Quote Price :- 1480.5999755859375 Volume :- 7660486.0	Price at this time is:- Adj Close 1095.0364990234375 Close 1106.800048828125 Date 2020-10-09 High 1113.300048828125 Low 1088.449951171875 Open 1095.0999755859375 Volume 10567867.0





Stock Information

Point selected:	Current market price:	Current status of stock:	Data info:
Point selected... index: 2654	Current market price is:- 1480.5999755859375	Current status of stock:- 1y Target Est :- 790.72 52 Week Range :- 692.10 - 1,489.45 Ask :- 0.00 x 0 Avg. Volume :- 7209463.0 Beta (5Y Monthly) :- 0.58 Bid :- 0.00 x 0 Day's Range :- 1,467.50 - 1,489.45 EPS (TTM) :- 44.67 Earnings Date :- Jul 14, 2021 Ex-Dividend Date :- May 31, 2021 Forward Dividend & Yield :- 30.00 (2.05%) Market Cap :- 6.286T Open :- 1472.7 PE Ratio (TTM) :- 33.14 Previous Close :- 1473.9 Quote Price :- 1480.5999755859375 Volume :- 7660486.0	Price at this time is:- Adj Close 1095.0364990234375 Close 1106.800048828125 Date 2020-10-09 High 1113.300048828125 Low 1088.449951171875 Open 1095.0999755859375 Volume 10567867.0

Figure 10 & 11 Predictions page for single and multiple stocks respectively

The screenshots above show the page displayed when a user requests a future stock price to be predicted. The predictions are shown as an overlay on the graph from the current data, when presenting the project to the assessors, it was mentioned that it was hard to see exactly where the predictions were made from. To tackle this issue in the future the line from where the predictions are made will be changed to a different colour.

Testing

These were the tests devised to ensure that the features of the system were working as anticipated. The testing was done through a series of unit tests, taking place after each stage of the system was completed. Stocks and search terms were selected randomly to avoid bias, and the bulk of the testing consisted of ensuring that the outputs from fetching and manipulating the data were as expected.

Precondition	Test	Expected Result	Result
Fetching of historical data for Fortune 500 companies.	Plot a graph of the time series based on extracted data.	Match the time series graph displayed on Yahoo Finance.	PASS
Fetching of tweets for a given company.	Fetch all tweets containing a search word and compare using twitter as a regular user.	Match twitter feed when the same search term is used.	PASS
Apply sentiment Analysis to tweets.	Apply TextBlob on fetched tweets.	Able to determine sentiment of each tweet and display the output as a pie and bar chart.	PASS
Train ARIMA model.	Feed data into the model and test different hyper-parameters.	Return hyper-parameters that achieve the best results.	PASS
Prediction using ARIMA model.	Use the model to make a prediction.	The model makes a prediction where the accuracy is at least 40%	PASS
Train hybrid model.	Feed data into the model and test different hyper-parameters.	Return hyper-parameters that achieve the best results.	PASS

Prediction using hybrid model.	Use the model to make a prediction.	The model makes a prediction where the accuracy is at least 40%.	PASS
Fetch Real time information for a given company.	Search for a stock on the website and compare displayed result with Yahoo Finance.	The displayed result should match the statistics provided by Yahoo Finance.	PASS

CHAPTER 4 – RESULTS AND EVALUATION

This chapter covers the ways in which the models were evaluated, how best to use the system, the effects of incorporating LSTM into the ARIMA model and the effects of sentiment analysis on investment decisions.

Analysing Results

To determine whether the hybrid model performed better than only using ARIMA for predictions, a comparative analysis was conducted between the two models on historical data. To evaluate the performance of the models, four different metrics were used- RMSE, MSE, R-squared score and MAPE.

RMSE calculates the square root of the variance of the residuals, and thus applies a greater penalty to more significant errors. MSE is a measure of the average differences between predicted values and the ground truth finding the mean by squaring them. Whereas MAPE highlights how closely predicted values associate with actual data in terms of percentage deviation.

R-squared score can also be referred to as the coefficient of determination; it calculates the level data correlates with regression lines. RMSE, MSE and MAPE are absolute fitting measures whereas R-squared score is a relative measure. The lower the RMSE, MSE and MAPE the better the prediction model and the higher the R-squared score the better the fit of the model. [36]

The impacts of sentiment analysis will be analysed by examining the sentiment analysis data on a given day for a stock and comparing the results with the real stock price movement in the short term. If there is found to be a correlation between the sentiment analysis and the short-term movement of the stock, then the analysis is potentially useful for investors as it provides an indication as to whether it’s a good time to buy, or whether it’s beneficial to wait until sentiment changes.

Model Results

This section discusses the results of the model when used for predicting prices over shorter and longer periods of time and any conclusions that can be drawn from these results, and the results of the sentiment analysis.

Stock Prediction (Short-Term)

This section shows how the model performed when tasked with making predictions about the value of Alphabet Inc ($GOOGL) for the next trading day.

Model	R-Squared Score	RMSE	MSE	MAPE
ARIMA	96%	2.71	7.19	0.015
ARIMA LSTM Hybrid	98%	1.95	4.09	0.012

Table 2 Short-Term prediction model results

From the data in the table, it is clear that the hybrid model is superior, as it outperforms the ARIMA only model in every metric. This is visually highlighted in the diagrams that follow showcasing the results of the model when tested on Google, using the ARIMA only model and the hybrid-model respectively. The red line represents the test data used, and the blue line represents the prediction made by the model. The dates and stocks selected match those used in a study by Infosys, where they developed an ARIMA model for stock price prediction. The purpose of this was to cross reference the findings of this study with the papers, to make sure that there were no significant conflicts. The model results are largely similar to those found in the paper for their ARIMA model, they reported an MSE of 7.186, a RMSE of 2.68 and a MAPE of 0.016. Therefore, it is reasonable to assume that the model was implemented correctly.

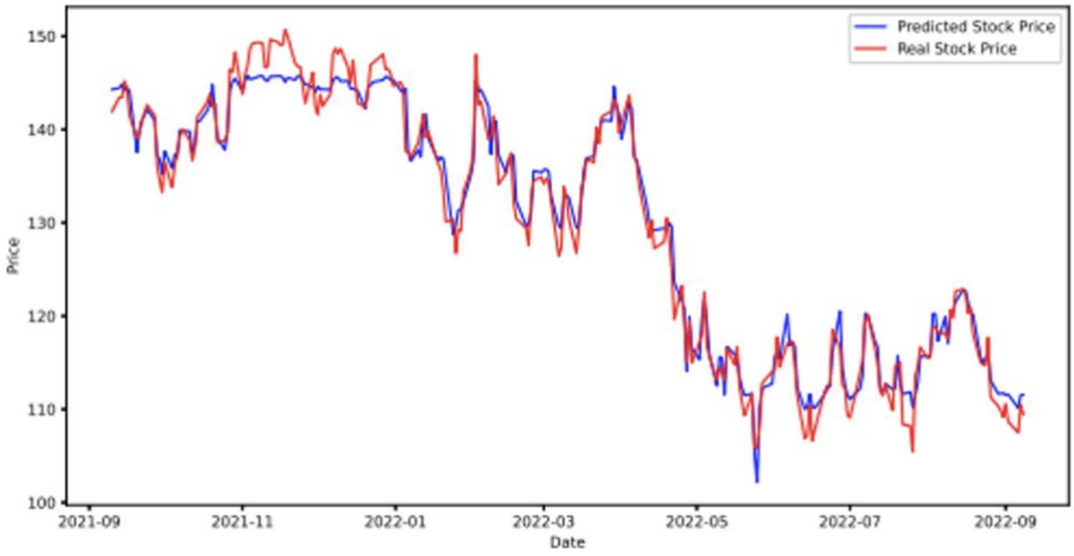


Figure 12 Results of Hybrid Model for short-term prediction

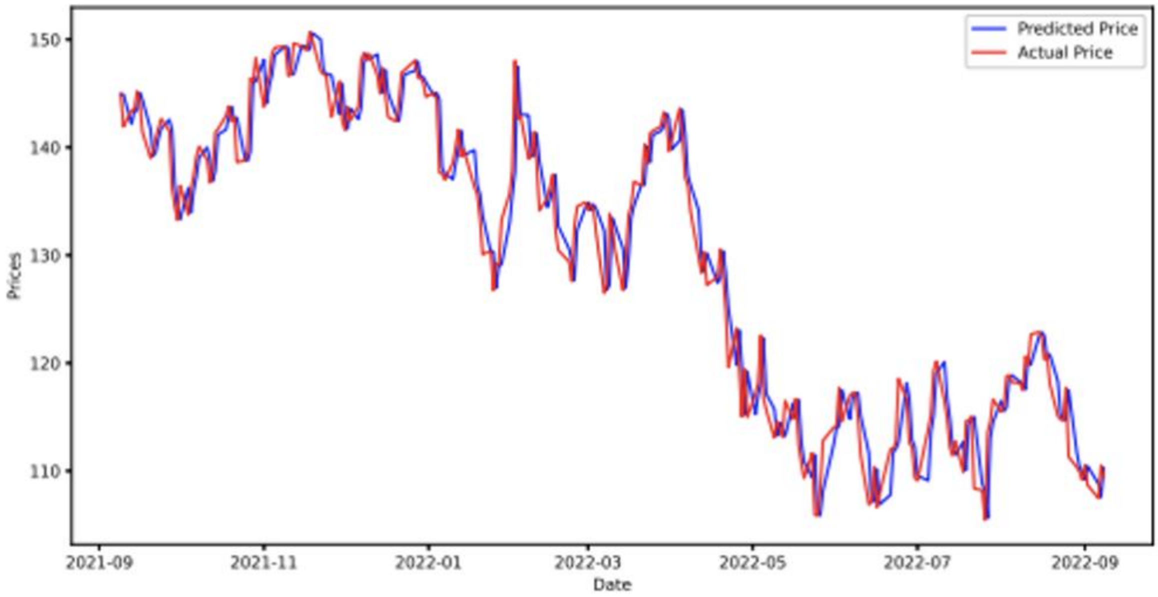


Figure 13 Results of ARIMA model for short-term prediction

Conclusion (Short-Term)

From these results it can be concluded that combining LSTM into the ARIMA model for stock price prediction can improve the accuracy of predictions. However, it is worth noting that if a day trader was to use this model the predicted values appear to lag very slightly behind the real price.

Stock Prediction (Long-Term):

This section describes how the hybrid model performed when tasked with making predictions for the next three years on several different stocks when making predictions from January 1st 2020 till January 1st 2023.

Company	Predicted Price	Actual Price	Error %
Apple	228.60	130.28	43%
Microsoft	406.17	239.58	41%
Amazon	133.21	85.14	36%
Disney	189.45	88.98	53%
Pfizer	48.42	51.24	6%
IBM	152.67	140.10	8%
Allstate	159.54	136.66	14%
Costco Wholesale	474.52	453.28	4%
AT&T	43.11	18.74	57%

Ford Motor	9.98	11.68	17%

Table 3 Results of Long-Term Prediction

If we consider an accurate prediction to be one with an error margin of 10% or less, the model is accurate 30% of the time. This shows that the model is not reliable when tasked with making long term predictions.

It should be noted however, that generally, the model was largely accurate when making predictions for the first two years, but after the Russia Ukraine conflict in early 2022, the stock market as whole dropped significantly. The model was not able to factor this event into the predictions, which affected the performance significantly. At this time, context regarding the geopolitical climate would have proven useful, and would likely cause investors to expect downtrends in prices. One could overlook this and conclude that the model is accurate in the instance that a major event does not occur within the next three years. However, this decision could be a naive as recently a significant event that has affected the stock market has occurred every few years, most recently the Russia Ukraine conflict, but going back further, the impacts of COVID-19 and the 2008 financial crisis. This suggests that unforeseen events must be factored into any long-term investment strategy.

This is further highlighted by the figures shown below, for Pfizer, Amazon and Apple respectively. Due to their involvement in the development of a COVID-19 vaccine, Pfizer were a company that could be said to have benefited from the pandemic financially. As COVID-19 did not cause detriment to their share price as significantly as it did when compared to the majority of other companies, the predictions made by the model were more accurate. In the figures for Amazon and Apple, I downtrend can be observed in early 2022, when the conflict started.

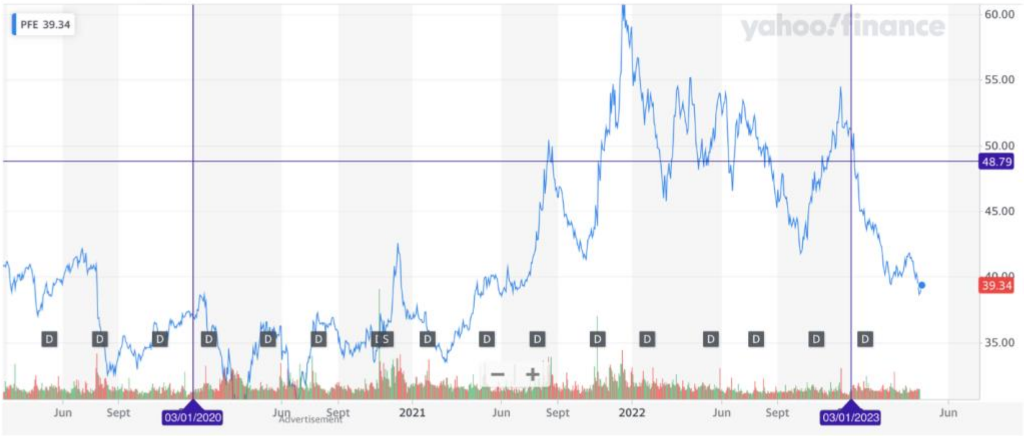


Figure 14 Actual Pfizer

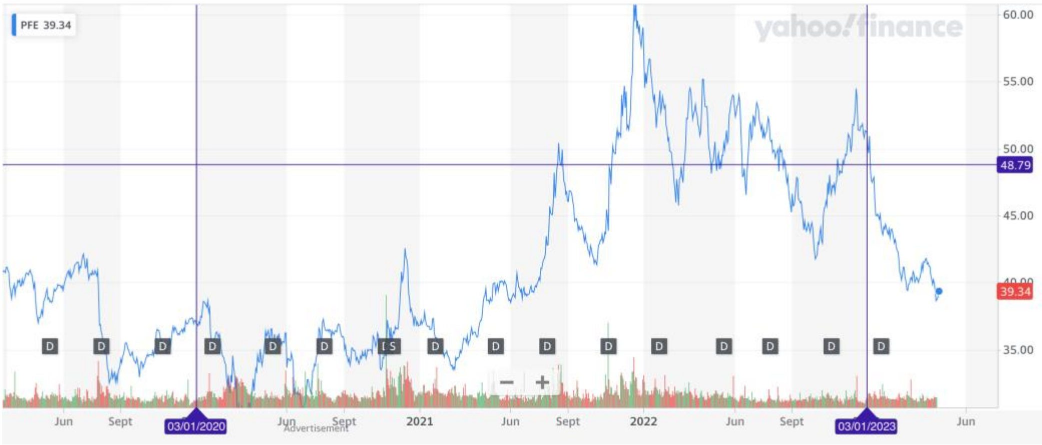


Figure 15 Actual Amazon

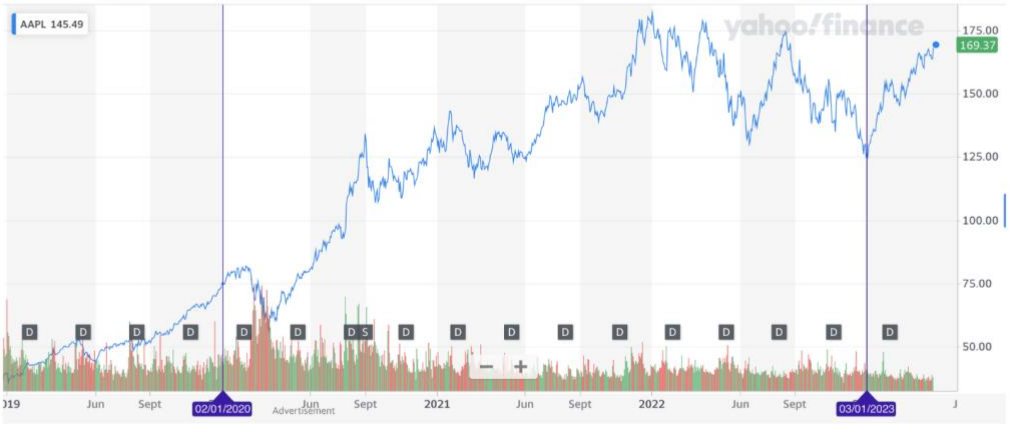


Figure 16 Actual Apple

The vertical bars represent the starting and ending periods.

Sentiment Analysis

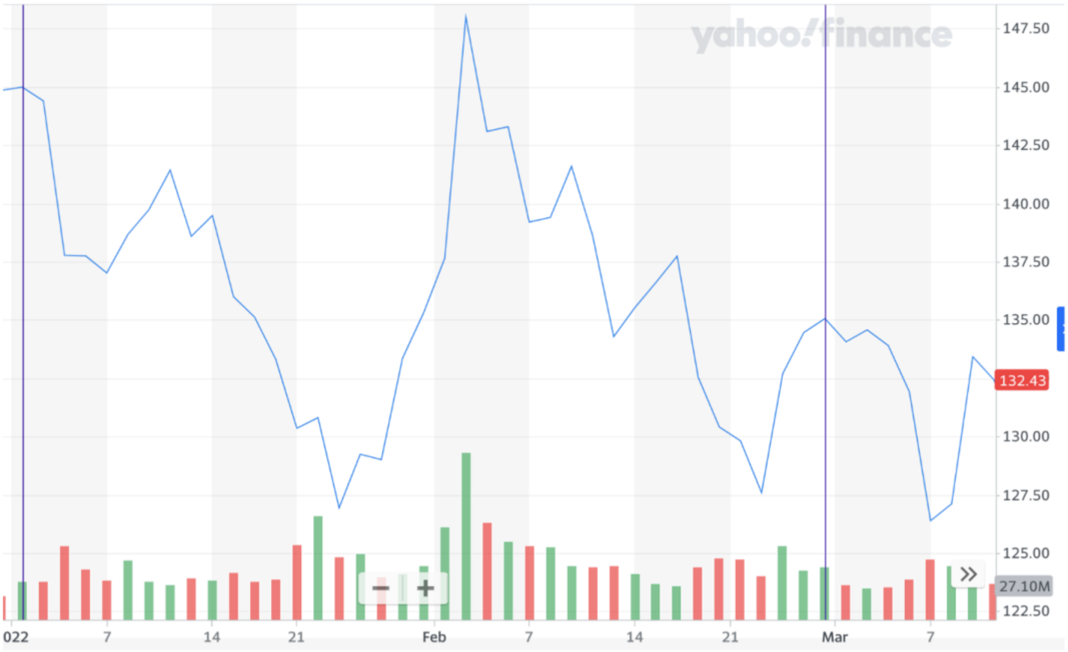


Figure 14 Actual Yahoo Finance stock chart for Google

Week Commencing	Sentiment At Week Beginning	Average sentiment for the week	Price Change %
3rd January 2022	Negative	Negative	-6.8%
10th January 2022	Positive	Positive	-0.7%
17th January 2022	Negative	Negative	-3.7%
24th January 2022	Negative	Negative	+8.0%
31st January 2022	Positive	Positive	+3.0%
7th February 2022	Negative	Negative	-2.9%
14th February 2022	Negative	Negative	-2.3%
21st February 2022	Negative	Positive	+1.5%

Table 4 Results of sentiment analysis on $GOOGL over 3/1/22 – 28/2/22

The table above shows the results of sentiment analysis of tweets at the start of a given week, the average for the week and the actual price change in percentage in price for the week. The tweets were analysed using TextBlob, however they had to be pasted directly from twitter, as the API no longer allows specific date ranges to be queried. There is a clear correlation between public sentiment and the movement of the share price, however, upon closer inspection the sentiment appears to lag behind the movement of the stock price.

Here are two reasons why this may be the case:

1. One factor that restricts the efficiency of sentiment analysis in Twitter is time. The information gathered from tweets necessitates a considerable amount of processing, analysing and interpreting. However, by the time this breakdown is culminated, stock prices may have gone up or down in a different direction.
2. Users of social media can often exhibit 'mob like' behaviour where, if someone with a large following, tweets something, other people tend to agree and support their statement without conducting appropriate research. Often when such an account has made a positive tweet about a stock, people continue to follow this positive sentiment even if the share price dips, producing a 'lag' effect.

There was also found to be a lot of noise in the data. The results from the analysis indicate that a significant number of tweets are not useful for sentiment analysis because of spam, irrelevant and bot-generated tweets that can skew the data to lean towards a particular sentiment. Out of the tweets analysed roughly TextBlob was only able to determine a sentiment for approximately 60% of the tweets, as many contained simple statements with images attached, such as ‘$GOOGL earnings this week’ followed by a figure.

The results of the sentiment analysis suggest that sentiment analysis of tweets could actually be detrimental to trading strategies, if it is used as the basis for the signals. However, if an investor notices that, at a point in time that the sentiment and price movement is conflicting it can be a good time to long or short a stock. For example, if the price of a stock had just started to increase significantly and twitter sentiment was negative, it could be considered a good time to buy as once the social media sentiment changes, stocks tend to display accelerated movement in the direction that the sentiment analysis suggests.

Overall Findings

The intention of this dissertation was to assess how effective a hybrid ARIMA and LSTM model is when predicting stock prices, as well as to explore the correlation between tweet sentiment analysis and corresponding stock price. The research discovered that the hybrid model fared better in performance than a model solely using ARIMA, and that sentiment analysis lags behind the stocks' movements.

Regarding the first objective that evaluated how proficiently the hybrid ARIMA, and LSTM model predicts stock prices; findings show that it outperforms the ARIMA only model at achieving more accurate prediction results. This result is consistent with several prior studies concluding that integrating both forecasting procedures increases accuracy by catching intricate patterns in the historical data. Thus, implying that the mixed-model approach is integral for successfully making confident predictions regarding constantly fluctuating market climates.

The model, however, is not reliable when tasked with forecasting the long-term price of a stock, as it cannot predict the effects of significant events that may happen in the future. However, if predictions are made for a shorter period, the model can provide a good estimate, as long as a significant event does not affect the company.

The second objective focused on examining how emotion detected from tweets correlates with movement in related stock pricing values over time. This project demonstrates tweet-sourced sentiment predicts pricing change eventually but lags behind trading behaviour. Many of the studies on social media sentiment analysis come to various different conclusions, with some suggesting that sentiment analysis can be a good indicator for prediction stock movement. However other studies are consistent with the findings in this paper, confirming tweet compiled indicators are unable to pre-empt price trends accurately.

Nonetheless, this contextual information can assist traders in surveying shifts effectively. Indications directing possible future rises or falls in corresponding current markets driven by emerging negative sentiments encompassing firms' performance signals stock prices heading downwards. The percentage breakdown of the sentiment analysis is also a good indicator of whether this change in direction is likely to be sharp.

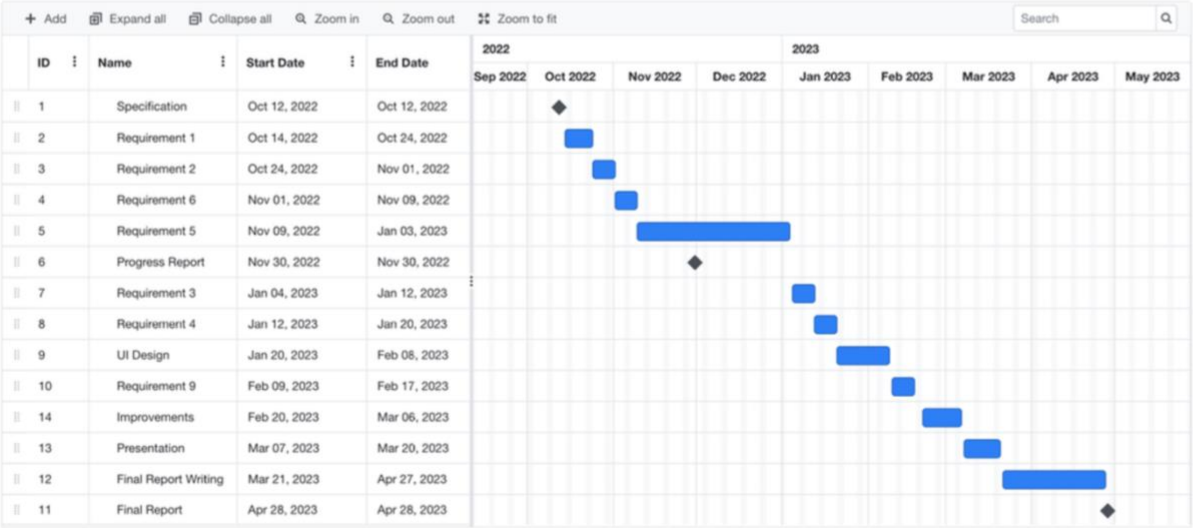
CHAPTER 5 – PROJECT MANAGEMENT AND IMPROVEMENTS

Project Management

In order to achieve all the objectives within the allotted period while being able to tackle any unforeseen occurrences and setbacks, project management was an important part of this dissertation. Hence, direction was instigated prior to starting development through picking the right strategies that allowed time spent working on the project to be used as efficiently as possible.

In the project specification an agile methodology was chosen, aiming to implement a new version of the system every two to three weeks. This strategy was generally effective, as it allowed sufficient time for a main feature to be developed at each cycle and then be evaluated. Initially developing the first ARIMA model was challenging due to having no prior experience with developing machine learning models, however after this model had been developed, the hybrid model was easier to implement.

The timetable developed in the project specification had to be adapted due to lecture allocations and co-curricular responsibilities. However, the number of hours spent working each week was kept relatively the same.



ID	Name	Start Date	End Date
1	Specification	Oct 12, 2022	Oct 12, 2022
2	Requirement 1	Oct 14, 2022	Oct 24, 2022
3	Requirement 2	Oct 24, 2022	Nov 01, 2022
4	Requirement 6	Nov 01, 2022	Nov 09, 2022
5	Requirement 5	Nov 09, 2022	Jan 03, 2023
6	Progress Report	Nov 30, 2022	Nov 30, 2022
7	Requirement 3	Jan 04, 2023	Jan 12, 2023
8	Requirement 4	Jan 12, 2023	Jan 20, 2023
9	UI Design	Jan 20, 2023	Feb 08, 2023
10	Requirement 9	Feb 09, 2023	Feb 17, 2023
14	Improvements	Feb 20, 2023	Mar 06, 2023
13	Presentation	Mar 07, 2023	Mar 20, 2023
12	Final Report Writing	Mar 21, 2023	Apr 27, 2023
11	Final Report	Apr 28, 2023	Apr 28, 2023

Figure 15 Project Timeline Gantt Chart

The Gantt Chart shown above was developed in the project specification. It was followed closely and was correct in predicting that developing the machine learning models would take the longest amount of time. There was not much development over the Christmas break, however before it started extra hours were put into developing the parts of the system that had been scheduled to be completed by this point, to mitigate any time related issues caused by this.

Risks

There were two main risks identified for this project. The first was not being able to develop the model due to limited experience with machine learning. This did not prove to be an issue, as there were many resources available to aid with the development of the model. The machine learning and neural networks modules were also helpful as the lab work often required data manipulation that was similar to that required for this project and provided further insight into the theory behind the models.

The other risk identified was the risk of accidental loss or corruption as this project relies heavily on software. This also did not arise as an issue, as backups were made regularly. One of the risks that I did not consider when planning the project was changing restrictions on the use of the twitter API. Since the change in ownership significant changes in the number of requests that could be made as well as in the filters for searching for tweets were made. This meant that when analysing historical tweets in some cases the tweets had to be extracted manually, which was very time consuming when compared to current tweets using the API.

Results Evaluation

The results of the model are consistent with other related work in that feeding the results of an ARIMA model into an LSTM model can improve accuracy if the right hyper-parameters are used. The tweet sentiment analysis proved to be useful in some circumstances, however, was not a good indicator when used to pre-empt stock price movements. The dataset that sentiment analysis was performed upon proved to be noisy, often with large quantities of irrelevant data. This suggests that Twitter is perhaps not the best platform to determine sentiment from, and other platforms such as news websites could be used, as they are less likely to be ‘noisy’ or be considered spam.

Future Work

The hybrid ARIMA-LSTM model has demonstrated promising results. However, much can be done to enhance and prolong its viability. this section discusses several areas where this model can ideally be further developed.

Advanced Natural Language Processing

To begin with, advanced natural language processing (NLP) techniques could be employed in future attempts to improve the sentiment analysis component. The current approach of the model utilises a basic method towards gauging public opinion through tweets; however, there are other methods using more sophisticated models such as topic modelling or emotion detection that could provide even deeper insights into the emotions conveyed in social media posts. Furthermore, expanding the model's scope by using news articles or other financial reports could also be promising.

Attention Mechanisms

Another potential area of development is incorporating attention mechanisms within the LSTM component. Attention mechanisms would improve the model's ability to focus on critical information from past events as well as facilitate improved predictions based off previously implied dependencies especially since long-term trends are key indicators pertinent to stock prices in the financial domain.

Scope

Additionally, whilst this study focuses solely on Fortune 500 companies, it can potentially extend itself within multiple markets if input data proves reasonable. If collated properly more diverse investment strategies are inevitable offering diversified investments and allowing risk to be spread.

Reinforcement Learning

Another possible improvement would be exploring reinforcement learning techniques to enhance predictive capabilities. Reinforcement learning can allow the model to learn from its own predictions and feedback from the market, meaning it is able to continuously adapt and improve itself over time.

Conclusion

In conclusion there stands greater potential in the hybrid ARIMA-LSTM model concerning stock price forecasting. The incorporation of other NLP techniques in sentiment analysis, attentive mechanisms, expansion and reinforcement learning would likely improve the model’s accuracy.

BIBLIOGRAPHY

1. John J. Murphy. "Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications," Page 23. Penguin, (1999).
2. Hong, H., Bian, Z. & Lee, CC. “COVID-19 and instability of stock market performance: evidence from the U.S.”. Introduction *Financ Innov* **7**, 12 (2021).
3. M K Ho et al J. Phys “Stock price prediction Using ARIMA, Neural Network and LSTM Models” Page 12. Journal of Physics IOP Publishing Conf. Ser. 1988 (2021).
4. Fatoumata Dama and Christine Sinoquet. “Time Series Analysis and Modelling to Forecast: a Survey” Page 10 LS2N / UMR CNRS 6004, Nantes University, France (2021).
5. Abhresh Sugandhi “What is Long Short Term Memory (LSTM) - Complete Guide” https://www.knowledgehut.com/blog/web-development/long-short-term-memory (2023).
6. Sahaj Singh Maini. "Stock market prediction using data mining techniques," Abstract, IEEE (2017).
7. Meghna Misra. “Stock Market Prediction using Machine Learning Algorithms: A Classification on Study”. Literature Review. IEEE (2018).
8. Ashish Sharma, Dinesh Bhuriya, Upendra Singh “Survey of Stock Market Prediction Using Machine Learning Approach” Section III, Prediction Method, Regression IEEE (2017).
9. Rachna Sable Bennett. University Department of Computer Science and Engineering, Greater Noida “Empirical Study on Stock Market Prediction Using Machine Learning” Page 4 IEEE (2019).
10. Yahoo finance website https://uk.finance.yahoo.com
11. Yahoo_fin Documentation https://theautomatic.net/yahoo_fin-documentation/
12. Tweepy Documentation https://www.tweepy.org
13. Maria Gabriella Xibilia. (2021) Forecasting Nonlinear Systems with LSTM: Analysis and Comparison with EKF. 4.42 Architecture Validation
14. G. Peter Zhang (2003) Time series forecasting using a hybrid ARIMA and neural network model. 5. Conculsions
15. TensorFlow documentation https://www.tensorflow.org
16. Cohen, M., & Frazzini, A. (2008). Economic implications of learning from social media. Journal of Financial Economics, 107(3), 393-410.
17. Kim, M., Kim, S., & Lee, S. (2019). Machine learning for stock prediction: A systematic literature review. Expert Systems with Applications, 118, 1-15.

18. McKenna, B., & Richardson, J. (2019). Social media, sentiment analysis, and the stock market: A review of the empirical literature. Journal of Economic Surveys, 33(3), 792-822.
19. Zhang, Y., & Kang, Y. (2018). Sentiment analysis and stock price prediction: A review of the literature. Expert Systems with Applications, 113, 77-94.
20. Selva Prbhakaran. ARIMA Model – Complete Guide to Time Series Forecasting in Python. What are AR and MA models?
21. Yamaha, Yokoe, Yamiji (2019) ‘Electricity load forecasting using clustering and ARIMA model for energy management in buildings’. ARIMA Model.
22. Yamaha, Yokoe, Yamiji (2019) ‘Electricity load forecasting using clustering and ARIMA model for energy management in buildings’. ARIMA Model.
23. Hsiao-Tien Pao (2006). ‘Comparing linear and nonlinear forecasts for Taiwan’s electricity consumption’. Conclusion. National Chiao Tung University.
24. Granger & Joyeux (1980). ‘An introduction to long-range time series models and fractional differencing. Journal of Time Series Analysis 15-30.
25. Fildes and Makridakis (1995). ‘The Impact of Empirical Accuracy Studies on Time Series Analysis and Forecasting’. International Statistical Institute. 300-305
26. Rian Dolphin (2020). ‘LSTM Networks | A Detailed Explanation’. Towards Data Science. A Comprehensive Introduction to LSTMs.
27. Dive Into Deep Learning (2022). ‘Long Short-Term Memory (LSTM). 10.1.
28. Fischer and Krauss (2018). ‘Deep Learning with long short-term memory networks for financial market predictions. European Journal of Operational Research. Results.
29. Kulshreshtha and Vijayalakshmi A (2020). An ARIMA-LSTM Hybrid Model for Stock Market Prediction Using Live Data. School of Computer Science and Engineering, Vellore Insititute of Technology, Chennai. 122
30. TextBlob. ‘Simplified Text Processing’. vo.16.0
31. Anshul Mittal (2011). ‘Stock prediction using twitter sentiment analysis’. Stanford University. 1-5
32. Adam Hayes (2022). Technical Analysis: What it is and How to Use It in Investing. Investopedia. https://www.investopedia.com/terms/t/technicalanalysis.asp
33. Forex.com. Technical analysis intermediate. https://www.forex.com/en-uk/trading-academy/courses/technical-analysis/uk-introduction/
34. Shen, C. Zhang et al (2019). Long Short-term memory neural network for stock price prediction. Journal of Computer Science, 101-111.
35. Selva Prabhakaran (2019). Augmented Dickey Fuller Test (ADF Test) Must Read Guide. Abstract and Introduction

36. Sakshi Kulshreshtha and Vijayalakshmi. An ARIMA- LSTM Hybrid Model for Stock Market Prediction Using Live Data. School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India. 122