University of Warwick

Department of Computer Science

Interim Report

Stock Market Prediction using Sentiment Analysis of News Articles and Stock Brokers Opinions

Author
Jack Wells

Supervisor
Dr. Tanaya Guha

July 16, 2020

Contents

1 Introduction

Stock market prediction is one of the most attractive research topics since the successful prediction can lead to significant profits, which has allowed for glamorisation as in movies such as “The Wolf of Wall Street”, and “The Big Short”. Historically predictive methods are based on the analysis of historic market data, such as stock prices, moving averages and daily returns [1]. However, recent advantages in computing have allowed for the fast processing of a much more of the data, and so the context of the world around the company is now can now to be used to improve the modeling of stock prices.

Stock market prediction is considered one of the most challenging time series prediction due to the volatile nature and the noise the data contains [2]. Over the last few decades machine learning models, such as Support-Vector-Machines (SVM) and Support-Vector-Regression (SVR) have been used in the prediction of time series data, such as the financial data in this project [3] [4] [5] [6] [7]. These methods in these financial literature exploit sentiment signal features, which are inherently limited by not considering factors such as events and news context. Thus does not consider or attempt to model the causality of the stock price change.

How to accurately predict stock price movement is still an open question, the Efficient-Market Hypothesis (EMH) [8] suggests that the stock price of a company reflects all available information and that any price changes are based on newly revealed relevant information, and thus the collection of this additional information will lead to more accurate models.

In this project it is hoped that the problem of not utilising all the data available will be addressed by leveraging deep neural models to extract rich semantic features from news text and from the opinion of stock brokers, it is hoped that this method is competitive with other state-of-the-art approaches demonstrating the effectiveness of Natural Language Processing (NLP) for computational finance. The use of NLP will hopefully be able to capture trends in data that are easily predictable given the information e.g. a change in the law or the private licensing of a product expiring may substantially effect a company, and therefore the stock price. The motivation behind the use of sentiment analysis in particular is as a result of the many papers that have shown significant improvements in when predicting the direction of a price movements using this. These include the use of sentiment in news articles and within other sources of data such as tweets [9] [10] [11]. The main goal of the addition of this data source is to further understand the causality behind the stock price movements, and as such will result in improved accuracy of the models.

2 Background

The area of stock market prediction is a well explored area going back many decades [12], attracting a considerable amount of research, with the methods changing over the years from using SVM to more modern methods such as neural networks. The use of NLP in this area is a more recent development, with most of them focusing on a binary classification, a simplification of the problem of predicting a stocks price (classifying either an increase or decrease in stock price). Some of the most widely studied approaches rely solely on analysing the recent prices and volumes of the traded stock [13] [14] [15] [16] [17] and it is hoped that the model made in this project is competitive with these approaches.

A model with the name Enalyst was introduced in Lavrenko et al and uses similar methods as this project intends to, that being the use of the headline of news articles. Vivek Sehgal et al. [18] and Michal Skuza et al. [19] used the sentiment analysis of texts to develop their models, with Michal Skuza paper using the analysis of tweets. However, all of these methods have limitations including unveiling the rules that may govern the dynamics of the market which makes the model incapable of capturing the impact of recent trends in the overall stock market. However, using the sentiment analysis of just textual data to predict an increase or decrease can't offer an accurate prediction on the exact actual price.

Recent developments in the use of neural networks to learn dense representations of text, have been shown to be effective on a wide range of NLP problems, given enough training data. This provides strong grounds for the exploration of deep-learning based models in the prediction of stock prices. For example Ding et al. [20] has shown that the use of deep-learning representations of event structures yields better accuracy compared to discrete event features. Again this is a crucial reason as to why this project uses NLP, specifically in the use of sentiment analysis of news articles.

2.1 The Problem

The simplest way to look at stock market prediction is the act of trying to determine the future value of a company [12]. Thus the problem in this project can be seen as a way of attempting to calculate the value of a company at a specific time, that may or may not reflect in the stock price of a company [2] for example it's possible for a company to be worth more than the current stock price suggests if the future of that company is bright compared to the opposite in the case of a failing company. Using this information the future stock price should reflect the value of the company.

The problem this project intends to provide some insight into is the development of predictive models to accurately predict the future stock price of a company based on the information that is available. It is believed that the more contextual (textual data, brokers opinions) data that the model is given the better the predictions should be. However, it is fully possible that the increase in data doesn't offer any advantage as the data may add noise, perhaps to an extent that no improvements can be made or that the extra data hiders the model.

The exact task to be solved will be, given the data of a companies broker opinions, news article titles and the financial data (that includes the daily high, low, close, open, adjusted close, and volume), these will be used to build data points with a rolling window approach. Then these data points will then be to predict the direction of the price change some days in the future. (In this report 10 days will be used to create each of the data point, and then from the last day used in the data point the closing value on that day will be compared to the closing price 10 days in the future).

2.2 Differences Between this Project and Other Work

One of the major differences that this project has compared to others is how the sentiment is derived from the news article headlines. It is intend that multiple news articles will be used together as this will hopefully give a better context of the company. This project will also use the BERT model [21] (Bidirectional Encoder Representations from Transformers) created by Google in 2018, which has been pre-trained on English Wikipedia and BookCorpus. This is a state-of-the-art model that can relate the content of many sentences together, in this project it will be used to relate the meaning of many news articles together, for instance the ten most recent or all articles that have been published in the last month.

Another key difference the use brokers opinions to help predict the future price, to my knowledge this hasn't been explored in other papers and research before. This, and the combining of multiple different sources of data i.e. the brokers opinions, news articles and financial data is something that is often not done. Most papers focus on one of these or maybe two but not three or more. The use of more of these data sources is hoped to improve the accuracy.

3 Methodology

The primary goal of this project is to predict the future price of a stock, however this is a particularly difficult goal so the problem will be simplified by only attempting to predict the direction of the price movement. The reason exact price predicting is so difficult is due to the volatility of the stock markets, as they are dependent on a huge number of factors.

This project begins with using just the stock data that is the high, low, close, ect. on a specific day, to attempt to predict the future price change. It will then go on to incorporate other features, these are the brokers opinions and news articles. News articles will be used although not the entire article or even the abstract only the headline, this is because the headlines have been shown in many pieces of previous work to work better [1]. This could be because they are more compact and often give a better overview of the sentiment of the text. Secondly, it allows for faster and easier processing of the text.

In this project the data is modeled for both the individual company and all the companies combined together. A rolling window approach will be used to make each of the data points, all scaling will be done on the data points within this window. For example 10 days will be used in one window and the scaling will be done on these 10 days only.

3.1 Metrics

For predicting the direction of the price change (increasing, decreasing) accuracy will be used as a metric. This is because the data set is a near 50/50 split when looking at the direction of price movements, over all the companies. For individual companies data sets are only ever a few percent off a perfect split, and so accuracy is a valid metric to use.

3.2 Data

To acquire a large sample of data to be able to come to the best conclusions possible the data will come from all the companies in the entire FTSE 100. These are the 100 largest companies (that are listed on the stock exchange) in the United Kingdom, these will hopefully have many news articles and brokers opinions written about them. However, one company has data that has many occasions that the price doesn't change (3i), and another has only been listed for 70 days. As such these two companies will not be used in this project.

The data has been collected between 1^{st} January 2014 and 27^{th} March 2020. There are three sources of data for this project these include Yahoo finance, that offers a API that allows the download of the financial information of each company over a period of time. This data includes the closing price, high price, low price, volume traded, adjusted close and the opening price of each company for each day the stock markets are open, and an example can be seen below:

Table 1: Stock price data, with no pre-processing

Date	Open	High	Low	Close	Adj Close	Volume
2014-01-02	765.5	785.5	760.5	783.5	707.394714	2448870
2014-01-03	780.5	782.5	769.5	776.5	701.074646	1086984
2014-01-06	776.0	792.5	772.5	790.0	713.263306	1704991
2014-01-07	790.0	807.5	790.0	804.0	725.903442	1812159
2014-01-08	805.0	811.5	789.0	790.0	713.263306	1172443

The second source of data was a website [22] that contained brokers opinions on the stock these are given in statements such as "buy" or "sell". However, there are also more ambiguous categories such as "house stock". So that the model can learn better, some of the noise is removed by forming five categories out of the the statements which show the relative "goodness" of the statement; these are buy, outperform, hold, underperform, and sell.

The third source of data in my project was another website [23], used to get news articles about the companies over the last 6 years. This was done with the use of web scrapers.

There is a fourth source of data in this project is an openly available data set that contains sentiment labeled news articles. These will be used to train the BERT model and also used to do some initial testing of other sentiment analysing methods and models.

3.3 Data Pre-processing

The stock price data will be scale, this means the data from multiple companies can be combine even if they have large differences in the average stock price. A statistical transform is used that models the data with a Gaussian distribution, with mean zero. Combine all the data together it is hoped that the models will improve as there will be more data available to learn from. This will hopefully benefit the LSTM model as its particularly data hungry.

The brokers data had to be pre-processed, this included turning the brokers opinions into the five categories as mentioned above. This data was one-hot-encoded as can be seen below:

Table 2: The brokers opinion data after initial processing.

Date	Buy	Outperform	Hold	Underperform	Sell
2014-01-13	0	0	1	0	0
2014-01-15	0	0	0	1	0
2014-01-31	0	0	0	1	0
2014-02-03	1	0	0	0	0
2014-02-05	0	0	1	0	0
2014-02-11	1	0	0	0	0
2014-02-19	1	0	0	0	0
2014-02-20	1	0	0	0	0

However, the brokers opinions are a long term prediction and so therefore it would make sense to use the same opinion for multiple days. This has been done by rolling each of the first time an opinion was seen for 100 days the stock markets where open. 100 days of trading was chosen as this is a reasonable length of time that the opinion would remain relevant for (in real time this is approximately 120 days). This is shown below:

Table 3: Brokers data after being extended for 100 days, this is the data for WPP.

	Buy	Outperform	Hold	Underperform	Sell
1573	2.0	1.0	1.0	0.0	0.0
1574	2.0	1.0	1.0	0.0	0.0
1575	2.0	1.0	1.0	0.0	0.0
1576	2.0	1.0	1.0	0.0	0.0
1577	2.0	1.0	1.0	0.0	0.0

This is then combined with the stock price data that a has been scaled, as described above.

The text has had the classic NLP pre-processing that includes stop word removal, these are words that don't carry much information such as "The", being removed. The words have also been stemmed this aims to reduce the vocabulary and increase word overlapping. Stemming reduces words such as "consulate", "consulting", "consultant" all become the word "consultant". Some of the words in the sentences that have been collected such as the word "ftse" and words highly related to it such as "ftse 100" have been changed into the word "index", this has a similar meaning but will have a better word embedding. Also any mention or abbreviation of a company name will be changed into the single word "company". This is because it's not expected that a specific company name will offer any additional information that can be used. However, when using the word "company", a better prediction model will hopefully be produced. In later parts of this project other pre-processing methods where required to be used to make the text data usable by the pre-trained BERT model these include adding special start, stop and end of sentence characters to the text. Sentences where then also tokenized, this allows a computer to understand the word that is being given. In all of the models word embedding where then used, these have the ability to carry contextual information that a word may have. For instance words with similar meaning have similar in meaning such as "good" and "great" have similar word embedding. The sentences are then padded with zeros to make them all the same length. An example for a sentence can be seen below:

Original: according finnair technical service measure due employment situation

Tokenized: ['according', 'finn', '##air', 'technical', 'service', 'measure', 'due', 'employment', 'situation']

Token IDs: [2429, 9303, 11215, 4087, 2326, 5468, 2349, 6107, 3663]

4 Modelling

Four different types of models are used to model the data on these are a logistic regression, SVM, XGBoost, and an LSTM. The hope is that the more complex LSTM model out performs the other models. These models have been chosen as the SVM and logistic regression are often used in the prediction of stock prices in similar research papers. The use of the XGBoost model was done as this is a type of decision tree that has performed very well on many kaggel tasks and occasionally in similar tasks to the in this project.

These models are trained on the individual companies, these are company specific models. The models use 1000 data points to model train the model on and the remaining for testing on this is approximately 531 data points. All the data is then combine all, this is done in a similar way to the individual example. That is by taking the first 1000 data samples (4 years of data) and testing on all data points after that which is about 531 data points (2 years of data) for each of the companies.

4.1 LSTMs and BERT Model

The LSTM (Long Short Term Memory networks) model used for the prediction has the following architecture:

Table 4: LSTM architecture.

Layer (type)	Output shape	Parameter number
LSTM	None, 1, 180	259920
LSTM	None, 1, 90	97560
LSTM	None, 30	14520
Dense	None, 1	31
Total parameters :	372,031	
Trainable parameters :	372,031	

The LSTM is again often used in stock prediction tasks, it is a recurrent neural network that makes use of memory modules that have the ability to remember parts of previous input data. This will hopefully give this model the ability to perform better than the other models. It is also often used on time-series data analysis tasks.

LSTM's have the ability to relate data points from before a specified time point allows for the improved results. The use of bidirectional nodes allows for the ability for the model to look forward and backwards in the data. This makes LSTM's very good at understanding text as they can look at words before and after the current target word, that the word context is trying to be learned, as the BERT model does.

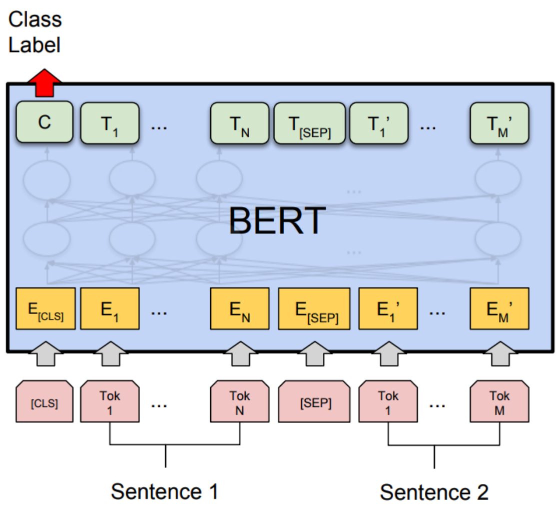

Figure 1: Visual representation of the BERT model.

The BERT model in particular is being used due to its state-of-the-art performance on tasks such as sentiment analysis and other NLP tasks like information retrieval, and summarising. As can be seen in the figure 1 it has the ability to understand the sentiment of multiple news articles at the same time. The BERT model has specialised transformer layers that convert the tokenized words and interpret them to perform so well on these tasks. The BERT model does also require the input of a mask, these can be show in the image as $E_{[CLS]}, E_1, \ldots$ where E_1 represents actual words and the other $(E_{[CLS]})$ special characters that indicate the start, end, and padding spaces within sentences. The model then uses transforms the words and then uses this to produce a sentiment label for the sentence based on the training data.

5 Results

The data that is modelling will take 10 days of data to model 10 days into the future, this will be for all of the models.

5.1 Financial Data Only

To begin only the stock price data will be used to train the models, the results are shown in the following table, the average accuracy that is generated by each of the companies and the variance in the accuracy for all 98 companies.

Table 5: Using only the stock price data and making new models for each company.

Model	Mean Accuracy (%)	Variance
SVM	51.28%	0.14%
Logistic reg	50.73%	0.11%
XBGoost	50.55%	0.08%
LSTM	50.11%	0.14%

Looking at the top 3 and bottom 3 results we see:

Table 6: Top 3 and bottom 3 models results created using only stock price data.

Company	SVM	Logistic Regression	XBGoost	LSTM	Average
EXPN.L	63.20%	60.50%	54.76%	57.45%	58.98%
JD.L	60.32%	60.32%	55.30%	51.35%	56.82%
SMT.L	56.55%	53.50%	56.91%	57.27%	56.06%
IMB.L	44.70%	43.27%	46.32%	46.14%	45.11%
ANTO.L	44.34%	48.47%	47.58%	38.60%	44.75%
GSK.L	43.27%	42.73%	43.44%	43.81%	43.31%

As we can see this model performs only slightly better than just guessing. Further investigating into building a combined model will be done as this is hoped to improve the results.

Combining All the Data

When combing all the data there are approximately 200,000 data points this does cause some issues in the time it takes for the SVM to train, and as such it wont be tested in this subsection.

The results of using the combined data to predict the results gives:

Table 7: Combining All the Stock Prices into one Model.

Model	Accuracy (%)
Logistic Regression	51.56%
XGBoost	50.91%
LSTM	51.58%

We can see that combining all the data does improve the results for all of the models when compared to building individual models for each of the companies.

Although the performance of this model is not satisfactory to be reliably used, so the project goes on to try to improve the results. This is done by adding the opinions of stock brokers.

5.2 Financial and Brokers Opinions

The addition of the brokers opinion is aimed to improve the results by trying to understand the causality behind the price change better. Using both the stock price and brokers data, the results are shown below:

Table 8: Using the stock price data and brokers opinions, new models for each company.

Model	Mean Accuracy (%)	Variance
SVM	50.65%	0.19%
Logistic Regression	50.29%	0.15%
XBGoost	50.09%	0.13%
LSTM	51.26%	0.23%

Again looking at the top and bottom three companies results.

Table 9: Top and bottom 3 performing models using brokers opinion and stock price data.

Company	SVM	Logistic Regression	XBGoost	LSTM	Average
LSE.L	60.50%	58.71%	63.20%	64.63 %	61.76%
AVV.L	62.66%	57.27%	58.53%	63.91%	60.59%
SGRO.L	60.86%	57.09%	59.07%	60.68%	59.43%
INF.L	45.60%	47.94%	40.57%	48.29%	45.60%
MGGT.L	42.37%	45.42%	41.29%	48.83 %	44.48%
IMB.L	43.27%	45.24%	42.19%	42.91%	43.40%

Initially, looking at the results we see that they are slightly worse when compared to the individual only financial data, for all models other than the LSTM. However, the best models do out perform those only using the financial data.

5.2.1 Using Only the Brokers Data

Now to see how good using only the broker data is:

Table 10: Only using Brokers opinion data to make models for each company.

Model	Mean Accuracy (%)	Variance
SVM	50.73%	0.24%
Logistic Regression	50.54%	0.27%
XBGoost	49.25%	0.20%
LSTM	51.41%	0.28%

Using only the brokers opinions does offer some improvement compared when using both the financial and brokers data together. And is comparable to using only the financial data.

Table 11: Top and bottom 3 performing models when using only the brokers opinion data.

Company	SVM	Logistic Regression	XBGoost	LSTM	Average
LSE.L	62.66%	62.66%	62.12%	64.63 %	63.02%
AVV.L	62.66%	64.63 %	57.09%	67.33 %	62.93%
HLMA.L	58.71%	53.50%	55.83%	67.86%	58.98%
VOD.L	44.52%	43.63%	45.24%	41.65%	43.76%
IMB.L	40.57%	43.99%	42.19%	46.32%	43.27%
RDSA.L	43.63%	43.45 %	40.57%	44.70%	43.09 %

Combining All the Data

Table 12: Combining all the data using broker and stock price data.

Model	Accuracy (%)
Logistic Regression	51.51%
XGBoost	50.81%
LSTM	51.58%

Table 13: Combining all the brokers opinion data of all the companies.

Model	Accuracy (%)
Logistic Regression	51.49%
XGBoost	51.64%
LSTM	51.90%

When looking at these we see that we have our best results so far when only looking at the brokers opinions and training on the combined data set.

5.3 Analysis of Results

The results on the whole do show that there are some improvements when using the extra brokers data source. However, this doesn't compare to some of the current results in other papers that are about 57% accuracy when using news articles. The limitations of using just the brokers opinion and stock data can be seen in more recent times. For instance in recent times due to coronavirus, many companies have had large drops in their stock price this has resulted in the last month or so of data being anomalous, as the price dropped "unexpectedly". This big price drop was unexpected in the data has been seen so far, when in reality quite foreseeable. Using news to build a better intuition of the price seems to be a good idea in creating better models, the articles should be able to offer a better context to the problem, and thus increased accuracy.

There are two ways to do this:

- Build model that takes text and stock price data together.
- Build a separate model to interpret the text and then feed this into another model to predict the future price.

In this project the second approach is used as combining multiple different types of data such as numbers and text is very difficult and it will be very hard to interpret the text in that kind of model.

5.4 News Article Analysis

Starting this project testing whether pre-trained sentiment analysers available from python packages such as NLTK are able to work on this task. These models are particularly good at getting sentiment from normal text, however when testing on a set of labeled financial news articles the accuracy was only 60% this is far too low to be used and to expect good results with. So to improve this an LSTM model that has had the weights initialised by the GloVe word embedding was used. It was trained on a data set of sentiment pre-labeled financial news articles, however this only gave a 53% accuracy. A better approach was needed, this was achieved with the use of the BERT model, this is performing well. Trained the model with a 71% training, 7.1% validation and 21.8% test split, a 85% accuracy was achieved on this task. This is an excellent result, especially when considering that even humans struggle to understand the sentiment (8 people where given these articles and only 66% or more of the people agreed on the sentiment). Using the model on the news articles that where collected in this project to acquire the sentiment of the articles, these being positive, negative and neutral. When looking at the labels produce by the BERT model the sentiments are particularly good, for instance a random sample of the data is shown below.

Table 14: Example of news articles after pre-processing and sentiment analysis.

Price change (%)	Sentence	Predicted Sentiment
-1.1	forget cash isa id rather buy company share price yield	neutral
-8.503	coronavirus pandemic gut hotel stay demand data	negative
-0.878	britain company raise fiscal year profit outlook	positive
-2.58	big oil billion fund back new cement engine technology	positive

This suggests that it will be possible to use this data source effectively to improve the accuracy of the current models. As has been done in many other papers which has this project attempts to reproduce.

6 Future of the Project

The main area of work that needs to be done is to take the data from BERT models sentiment predicted news articles and use them to improve the models. It can either done by using the exact predicted sentiment i.e. "Positive" or by using the vector the model produces that suggests how much of each sentiment the news article is.

Additionally, the performance of the models at this point isn't satisfactory, and as such an attempt to manually build new features from the stock price data such as the gradient of the closing price over the last few days may be used. Also making three categories to predict instead of just increase and decrease and introducing a third class, neutral that has a percentage change in price less than 5% for instance. This may improve the models abilities in finding good stocks to buy, which obviously relates to the overall use of this model. It also may also give better performance as this will hopefully reduce problems caused by volatility in the data.

6.1 Appraisal and Reflection

Currently the project is progressing well, and on schedule or even slightly ahead of schedule. Their have been some minor issues within the sentiment analysis part of this project, those being a lack of performance, however after introducing the BERT model it has been possible to accurately identify the sentiment of financial news articles. Another issue that was discovered in the presentation was an issue with scaling, but this has been resolved over the last few weeks.

6.2 Time Management

The time-line for this project ranges from 01/01/2020 to 01/09/2020. Various stages have been set out in order to monitor the progress in this project and allow for the right level depth in this project as to keep up with the time-line. For instance there is no need to develop a LSTM model with many layers if doing so will offer little benefit to the accuracy and may cause time pressure on more important parts of the project such as the NLP modeling. As it stands this project will be completed on time and there maybe time for more experimentation with different approaches.

References

[1] Liu and Huicheng, “Leveraging financial news for stock trend prediction with attention-based recurrent neural network,” Nov 2018.

[2] B. Wang, H. Huang, and X. Wang, “A novel text mining approach to financial time series forecasting,” Dec 2011.

[3] S. P. D. Basak and D. C. Patranabis, “Support vector regression,” 2017.

[4] E. O. J. P. B. S. M. A. Hearst, S. T. Dumais, “Support vector machines,” 1998.

[5] G. F. A. N. Refenes, A. Zapranis, “Stock performance modeling using neural networks: a comparative study with regression models, neural networks,” 1994.

[6] S. P. S. P. Das, “Support vector machines for prediction of futures prices in indian stock market.”

[7] C.-C. C. C.-J. Lu, T.-S. Lee, “Financial time series forecasting using independent component analysis and support vector regression, decision support systems,” 2009.

[8] B. G. Malkiel, “The efficient market hypothesis and its critics,” 2003.

[9] M. Kanakaraj and R. M. R. Guddeti, “Performance analysis of ensemble methods on twitter sentiment analysis using nlp techniques,” in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pp. 169–170, 2015.

[10] S. A. Phand and J. A. Phand, “Twitter sentiment classification using stanford nlp,” in *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, pp. 1–5, 2017.

[11] M. R. Vargas, B. S. L. P. de Lima, and A. G. Evsukoff, “Deep learning for stock market prediction from financial news articles,” in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 60–65, 2017.

[12] K. M. B. G. Malkiel, “A random walk down wall street,” 1985.

[13] E. Schoneburg, “Stock price prediction using neural networks: A project report,” 1990.

[14] V. Akgiray, “Conditional heteroscedasticity in time series of stock returns: Evidence and forecasts,” 1989.

[15] A. B. A. T. D. M. Gocken, M. Oz¸calıcı, “Integrating meta-heuristics and artificial neural networks for improved stock price prediction,” 2016.

[16] C. K. A. A. A. Adebiyi, A. O. Adewumi, “Comparison of arima and artificial neural networks models for stock price prediction,” 2014.

[17] H. A. K.-J. Kim, “Simultaneous optimization of artificial neural networks for financial forecasting,” 2012.

[18] C. S. V. Sehgal, "Sops: stock prediction using web sentiment," 2007.

[19] A. R. M. Skuza, "Sentiment analysis of twitter data within big data distributed environment for stock prediction," 2015.

[20] T. L. J. D. X. Ding, Y. Zhang, "Deep learning for event-driven stock prediction.," 2015.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. cite arxiv:1810.04805Comment: 13 pages.

[22] "www.lse.co.uk."

[23] "uk.investing.com."