Forecasting US Stock Price Movements Using Convolutional Neural Networks And News Sentiment From GDELT

Sander van Donkelaar
STUDENT NUMBER: 2002765

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE IN COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

dr. H. J. Brighton
Sebastian Olier Jauregui

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
May 2020

Preface

This research project has been conducted as part of the the graduation requirements of the CSAI program at Tilburg University.

Forecasting US Stock Price Movements Using Convolutional Neural Networks And News Sentiment From GDELT

Sander van Donkelaar

Forecasting US Stock Price Movements Using Convolutional Neural Networks and News Sentiment From GDELT

In the recent years, promising developments have taken place in the field of quantitative finance, in which convolutional neural networks have successfully been applied in forecasting stock prices. Although the results seem promising, the models have been trained only by using company data. This research builds upon these findings and investigated whether news sentiment from GDELT can improve the capabilities of Convolutional Neural Networks in forecasting stock movements. In addition, it is checked how this differs between companies from different industries. The results indicate that for some companies, news sentiment from GDELT can improve the predictive performance of the CNN model, although the results can vary greatly per industry and company.

1. Introduction

Predicting the development of future stock prices has received much attention within the scientific and financial domains. Stock prediction is challenging due to the volatile nature of stock prices and the sheer amount of factors that influence them. Although the extent to which one can accurately predict the stock market remains a point of debate, various hedge funds and investors such as David Harding and Harry Markowitz have used quantitative methods to guide their investing (Cookson 2016). The main goal of these methods is finding the critical predictors that can give a clue of a market or stock's future value. In the past decades, there has been a rise in the usage of machine learning and artificial neural networks to forecast stock price movements. Especially in the field of deep learning, new architectures have been developed and employed, which has led to promising developments in this field. Recently, a new approach has been undertaken in modeling financial time series data by using Convolutional Neural Networks (CNN's). Traditionally, CNN's have mainly been used in image classification, where convolutional layers are used to learn hierarchical feature representations of images. However, these models have recently also been successfully used to model financial time-series data (Gunduz, Yaslan, and Cataltepe 2017), (Sezer and Ozbayoglu 2018), (Gudelek, Boluk, and Ozbayoglu 2017)). The results of these studies look promising and indicate that CNN's could have some predictive capabilities in forecasting stock market movements. Although the results seem promising, both studies only used company data and technical indicators as input. The predictive capabilities of other types of variables, such as news sentiment, have not been addressed. Therefore, the current research builds further upon the findings of (Gunduz, Yaslan, and Cataltepe 2017) and (Sezer and Ozbayoglu 2018), as it was investigated to what extent stock price movements can be modeled using CNN's, and whether this can be improved by using news sentiment as a feature in the model.

In order to get the news sentiment, various approaches can be undertaken. For example, the web can be scraped to obtain articles of a given company. One significant implication of this approach is the enormous amount of news sources that are available. Scraping news articles from all these sources to build a data set can be complicated and time-consuming. Besides that, the process of web scraping is a grey area. Although scraping itself is not illegal, commercial use for scraped data is limited (Hirschey 2014), which is not convenient for financial domains. For this reason, other data sources for news sentiment are worth investigating. Therefore, the current study will investigate to what extent the Global Database of Events, Language, and Tone (GDELT) can be used as a source for news sentiment. GDELT can best be described as an incredibly large database containing millions of individual news articles, extracting detailed information on every person, organization, location, number, and theme mentioned in an article, which is updated daily (The GDELT Project 2015). Besides, each article is labelled with sentiment-tones that indicate how positive, neutral, or negative the article is (ibid.). Although GDELT contains an incredible amount of data, its abilities to forecast stock prices have not been addressed very extensively in the academic literature. However, due to the sheer amount of information that is available in GDELT, it is worth investigating to what extent this can help in forecasting stock trend behavior. In addition, it is worth investigating how the predictive capabilities of news sentiment differ across industries. Finding and modeling relevant predictors has a high priority in financial domains. Being able to correctly model trends in stock prices will lead to more efficient algorithms for automatic trading. Mainly due to the fact that nowadays a large part of stock trading is done automatically, it can give a potentially high competitive advantage over other investors, as buying undervalued stocks can lead to great boosts in revenue.

1.1 Research Questions

As noted earlier, the current study will investigate how the predictive capabilities of Convolutional Neural Networks can be improved by using news sentiment extracted from GDELT. Therefore, the overarching research question will read as follows.

RQ "To what extent does news sentiment extracted from GDELT help in forecasting short term stock movements using Convolutional Neural Networks?.

In order to answer this question, a selection of sub-questions have been addressed as well. First of all, it is vital to investigate to what extent short term stock movements without using news sentiment. The goal of this is to get a benchmark to which the model scores with sentiment can be compared. Therefore, the first sub-question reads as follows:

RQ 1.1 "To what extent is it possible to predict short term stock movements, using a Convolutional Neural Network in conjunction with company data and other economic indicators, compared to a baseline?".

Relevant company data and other economic indicators are selected based upon the literature. A clear overview of the selected features and their description is outlined in the appendix. The results are compared to a baseline in order to evaluate the performance. Afterwards, it is checked whether the predictive capabilities improve if news sentiment is added as a predictor, which leads us to the second sub-question:

RQ 1.2 "How can news sentiment extracted from GDELT improve the predictive capabilities of the model in forecasting short term stock movements?".

In order to get a complete overview of how news sentiment can serve as a predictor for future stock prices, the current study will also investigate the extent to which the predictive capabilities of news sentiment vary across different industries. This leads us to the next question:

RQ 1.3 " To what extent do the predictive capabilities of news sentiment in forecasting short term stock movements differ across industries?".

In order to answer this question, fifteen companies from five different industries will be investigated. A clear description of the selected companies can be found in table 1.

2. Related Work

2.1 Brief introduction to Quantitative Finance

For centuries, speculators have tried to find patterns in price fluctuations. In ancient times, early traders from Babylonia recorded prices and dates of crops in order to predict future prices (Zuckerman 2019), a method that can be described as one of the earliest forms of technical analysis. However, the extent to which one can accurately predict the stock market still remains a point of debate. One of the fundamental theories in financial economics is the Efficient Market Hypothesis (EMH), which states that stock prices reflect all information, making it impossible to deviate from or 'beat' the market (Fama 1970). A detailed description of the EMH is out of the scope of this study, but the EMH can be briefly divided into three categories: the strong, semi-strong and weak version. In short, the strong version suggests that stock prices reflect all public and private information. The semi-strong version suggests that it reflects all public information, and the weak version suggests that stock prices reflect all publicly available information, and that past price performance has no relationship with the future. Therefore, any attempts to predict the price based upon past historical data is useless (ibid.). However, although markets appear to be efficient in general, price regularities and even predictable market patterns can appear and persist for short periods (Malkiel 2003).

With the rise of behavioral finance, new light has been shed on the topic. One of the critical assumptions of the efficient market hypothesis is that players in the market act rational and have access to all available information. Yet, often investors often do not take all available information into account when making a decision (Comlekci and A.Ozer 2018). As a consequence, investment behavior is not always entirely rational. Therefore, markets are subject to the cognitive biases of those who invest in it (**?**). As a consequence, investors often make irrational decisions, eventually being the driving force behind market anomalies. Therefore, stock prices are influenced by various psychological, cognitive, and emotional factors, ultimately distracting the stock market from being effective (ibid.). When a market is inefficient, stocks do not accurately represent its true value, which results in under- or overvalued stocks. As a consequence, market anomalies can occur in which a situation occurs where the movement of stock prices can be predicted to some degree (Schwert 2003). For the past decades, various of these market anomalies have been found in the academic literature. For example, periodic anomalies have been found in which fluctuations of stock prices perform differently in certain periods, such as certain months or after holidays (Keim 1985); (Agrawal and Tandon 1994);). Other examples include that of (Dimson, Marsh, and

Staunton 2002), where it was found that companies with a higher dividend yield have greater annualized stock returns (2002). Moreover, it was found that when stocks that have been performing well over a three to twelve-month period, they will continue to do so in the subsequent months (Jegadeesh and Titman 2011). In addition, in a research conducted by Barbaris et al.it is demonstrated that news has the power to alter the behavior of investors in markets up to 12 months (1998). As a result, good news will result in positive returns and overvalued stocks and vice versa. These effects can remain for an extended period before returning back to their average values (Barberis, Shleifer, and Vishny 1997). Therefore, it is evident that markets are not always fully efficient. Market anomalies are evidence of systematic patterns in stock prices. The main goal in stock price prediction is finding these patterns in order to get an idea of a stock's future value.

2.2 Machine learning in Quantitative Finance

In the past decades, machine learning models have proven to be very powerful in extracting hidden relationships in large amounts of data. As a result, there has been a significant increase in using machine learning models to predict stock prices. Modeling financial data is challenging, mainly due to its temporal nature and the high number of variables that influence it. Because stock prices are time series data, particular methodologies need to be adopted to allow for time series analysis. Recent studies have shown that artificial neural networks can be powerful in predicting future stock prices and tend to outperform traditional methods, such as autoregressive models (Adebiyi, Adewumi, and Ayo 2014). In a study conducted by Kim Kim, a feature fusion long short-term memory-convolutional neural network (LSTM-CNN) is created, which combined different representations of the same stock data. In this way, the model was able to predict short term stock prices with an average RMSE of 0.098 (Kim and Kim 2019). One point of criticism is that the study only took company data into account. The predictive capabilities of other variables have not been investigated. Moreover, in a study by (Nguyen and Yoon 2019), a new framework is proposed in which LSTM networks are trained on stock data from several companies altogether. Afterward, the networks are fine-tuned by using small amounts of data from a target stock. In this way, data from multiple companies is used simultaneously to train a model to solve the problem of limited data, as each year has only 240 trading days. The model had an average prediction accuracy of 57.97% in predicting the movement of stock prices for the next 3 three days, which is a satisfactory result given the volatile nature of stock prices.

2.3 Convolutional Neural Networks in Quantitative Finance

Although convolutional networks are primarily known for their image classification capabilities, it is shown that Convolutional Neural Networks can also be used to model various types of multivariate time-series data. Examples of this include predicting energy consumption (Lara-Benítez et al. 2020); someone's sleeping stage (Yildirim, Baloglu, and Acharya 2019), or predicting stock prices. Usually, time series data is modelled with LSTM networks, but recent research indicates that convolutional neural network can be more efficient and do not suffer from vanishing or exploding gradients (Mittelman 2015). A more detailed description of the inner workings of convolutional neural networks is found in section 3.3.1. as noted earlier, the study will build further upon the findings of (Gunduz, Yaslan, and Cataltepe 2017) and (Sezer and Ozbayoglu

2018) where financial time series are converted into 2D images and fed into a CNN in order to extract any temporal relationships between variables, which can give a clue about a stock's future value. In the study of (Gunduz, Yaslan, and Cataltepe 2017), the hourly direction of 100 Stocks on the Borsa Istanbul Stock Market was predicted, by using historical stock price data and several types of technical indicators. This data was transformed into a 2D matrix, which is fed as an image to the CNN. The feature set consisted of different indicators, price, and temporal information selected using Chi-Square selection, which measures the dependence between a feature and the class label. Afterward, the features were ordered before they were presented as inputs to the CNN. The results were promising, as the best performing model had a mean MA F-Measure rate of 0.563, outperforming other baseline classifiers Naive (0.465) and Random (0.499) in terms of mean MA F-Measure rates.

Moreover, in a study conducted by (Sezer and Ozbayoglu 2018) a similar modeling approach was undertaken. Similarly to the study of (Gunduz, Yaslan, and Cataltepe 2017), financial time series was converted into 2-D images, and for each image, 15 different technical indicators were selected and calculated up to a 15 day period. As a result, 15x15 sized 2-D images were generated. Instead of having binary labels, each image is labeled as Buy, Sell or Hold, eventually causing the model to learn a trading strategy. The results indicate that the approach performs very well against other baseline models over long periods of out-of-sample test periods, with an average return of 12.83%, which was up to 3 x times as high as the baseline. Besides, in a study conducted by (Gudelek, Boluk, and Ozbayoglu 2017), Exchange Traded Funds were predicted using a 2D Convolutional Neural Network that was fed 28 x 28 images, which represented 28 features at 28 different time steps. The features consisted of technical indicators at different time intervals. Because 2D convolution processes learn relationships between pixel regions, features were clustered using agglomerative hierarchical clustering. Afterward, the images were fed into a Convolutional Neural Network. The network consisted of 2 two convolutional layers of 32 and 64 filters and 3x3 kernel sizes. These layers were followed by a (4x4) pooling layer and two fully connected Dense Layers. The results indicated that the next day's prices could be predicted with 72% accuracy. However, it should be taken into account that this is achieved by predicting the prices of Exchange Traded funds (ETF's) in order to avoid the high volatility of stock markets. In general, the results of these studies seem promising, yet neither takes the predictive capabilities of other types of variables into account. Therefore, it is worth investigating whether news sentiment can help forecast stock prices using Convolutional Neural Networks.

2.4 News Sentiment

Research suggests that there is a correlation between news sentiment and stock returns (Barberis, Shleifer, and Vishny 1997). Moreover, several studies have investigated the predictive power of news sentiment in forecasting stock prices. In a study by (Pagolu et al. 2016), it is examined to what extent sentiment extracted from Twitter tweets is correlated with the fluctuation of stock prices. The results showed a strong correlation between the two variables. Although the focus of the current research will not be on sentiment extracted from Twitter posts, it indicates that sentiment, in general, can be a significant predictor for future stock prices. In a research conducted by (Vanstone, Gepp, and Harris 2019), autoregressive neural networks were trained on historical stock data and sentiments extracted from Bloomberg news articles and Twitter posts to anticipate future stock prices. It was found that, although the results varied per company, sentiment extracted from news articles and Twitter posts, in general, significantly improve the quality of stock price predictions. Another interesting note is that when the models are trained to predict only the direction of the next movement (up or down), instead of a continuous value, no significant difference between the baseline model and sentiment model was found. A study conducted by (Audrino, Sigrist, and Ballinari 2018) is in line with these results. By using a predictive regression model and an extensive dataset of daily economic and sentiment variables, it was found that sentiment can predict the future volatility of stock prices for a two-week horizon. This effect was also present when controlling for other economic variables. Besides, research from the Federal Reserve Bank of San Fransisco indicated that market sentiment, together with momentum, serves as a robust predictor for the return on the Standard Poor's (SP) 500 stock index over the next month (Langsin and Tubbs 2020)

2.5 News Sentiment from GDELT

The number of academic studies that investigated the capabilities of GDELT to model stock prices is minimal. However, the results of the studies that did look promising. For example, in a study conducted by (Alamro, McCarren, and Al-Rasheed 2019), closing prices of the Saudi Stock Market Index were predicted by incorporating sentiment tones from GDELT, which were filtered for Saudi Arabian News. Together with social media attention and historical price data, the sentiment tones were fed into several multivariate models. A Long-Short Term Memory network had the highest performance among all models, which was able to give very accurate forecasts with a Mean Absolute Error of 0.59, comparing to the mean value of the stock index, which was 7413. Although the current research is focused on US companies, these findings indicate that sentiment from GDELT can be used to forecast stock prices.

3. Experimental Setup

As noted earlier, it is investigated to what extent news sentiment from GDELT can help in forecasting short term stock movements using Convolutional Neural Networks. In order to answer the question, a 2D CNN was created which predicts whether the closing price of the next day is either higher or lower than the previous closing price. The network is trained on 2D matrices, in which rows represent features and columns represent time-steps. This 2D matrix can best be described as an image that is an abstract representation of a certain time window. Instead of extracting spatial relationships between pixel regions in the image, the CNN can extract temporal relationships between regions of features and time-steps. This allows it to extract patterns which can give a clue about the stock's future direction. After the model was created, it was trained on economic and- company data. Afterwards, it was checked whether the predictive capabilities of the model improved when sentiment variables were added as a features in the image. In short, the entire experiment can best be divided into three steps: extracting the data, transforming the data and modeling the data.

3.1 Extracting the dataset

A total of fifteen companies from the SP 500 were investigated, ranging from five different sectors, including healthcare, banking, technology, oil energy, and airlines. A clear overview of the selected companies can be found in table 1.

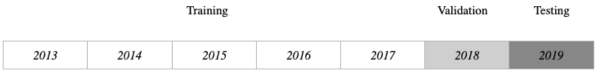
The time span of the current research is April 1, 2013, to December 31, 2019. The current study does not take weekends into account, as stock exchanges are closed on

Table 1: Overview of all selected companies!

Oil & Energy	Technology	Banking	Airlines	Healthcare
Exxon Mobil	Amazon	JP Morgan	Delta Airlines	Allergan
Chevron	Apple	Goldman Sachs	Southwest Airlines	AbbVie
Devon	Microsoft	Morgan Stanley	American Airlines	Gilead Sciences

these dates. The data will be split into three sets: a training, validation, and testing set. All data up to 2017 was used for training. Data from 2018 was used for validation, and data from 2019 was used for testing.

Figure 1: Train, Validation and Test subsets



As noted in the introduction, stock prices are affected by many factors. Therefore, various types of variables were used to model the stock prices. A clear overview of all variables and its description can be found in the appendix. In short, the variables can be grouped into three categories: company data, economic indicators and sentiment.

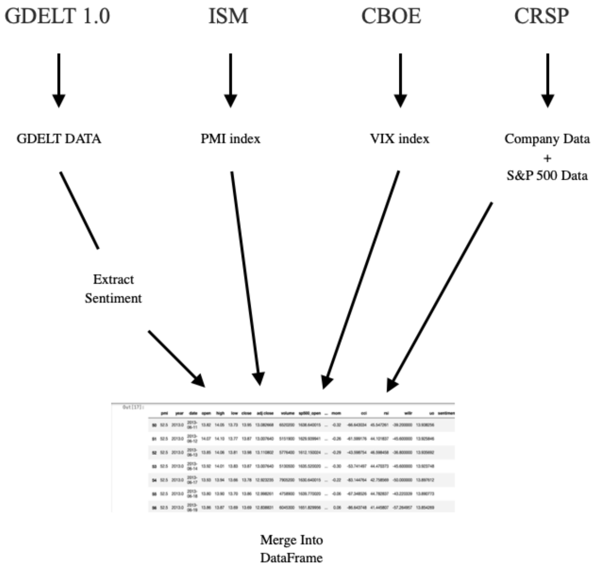
Company Data: The company data consists of open, high, low, and closing prices, as well as trading volume and dividend pay-out. The company data was extracted from the Center for Research in Security Prices.

Economic Indicators: Besides company data, economic data was also used. These consist of the open, high, low, and closing prices of the SP 500 and other relevant economic indicators. These include the ISM Purchasing Managers Index (PMI index) and the CBOE VIX index. The PMI index can be described as a diffusion index that represents whether market conditions will increase, remain the same or decrease, according to senior executives of more than 400 companies in 19 industries. Therefore, the PMI will be included as a variable as it can indicate expected trends in the US economy. The PMI data was extracted from the ISM website. The CBOE volatility index is a statistic that represents the market's expectation of 30-day forward-looking volatility. Therefore, it indicates the level of risk in the market. Volatilities are routinely reported by financial news services and closely tied to investor sentiment (Corrado and Thomas W. Miller 2005), which can give a clue about the movement of stock prices. Therefore, the VIX index was also included in the current study. VIX data was downloaded from the CBOE website.

Sentiment The main focus of the current study is sentiment extracted from GDELT. In the current study, the Global Knowledge Graph from GDELT 1.0 was used, which allows one to extract all news data of a given day (The GDELT Project 2015). In order to extract the data from GDELT, the GDELT Python library is used together with the Pandas library (McKinney et al. 2010). For any date up to April 1, 2013, the library returns DataFrame, which consists of articles written on that given day. In order to filter the DataFrame, the column 'ORGANIZATIONS' was used. This column shows all organizations which are mentioned in the respective article. For each day, the data was filtered by checking whether the company's name was mentioned in the 'ORGANIZATIONS' column. The filtered data was appended to a new DataFrame containing all articles in which the company was mentioned, ordered by date. Afterward, mean sentiments were extracted for each given day by using the ’TONE’ column, which contains a comma-delimited list of six core emotional dimensions. The first value in the list contains the average sentiment tone of the article. Afterward, the average sentiment is obtained by adding all sentiment values and dividing it by the total number of articles in which the company is mentioned on that given day. Moreover, a variable ’result count’ was created, which counts how often a company is mentioned on GDELT for each day. Besides the company sentiment, country-level sentiment is also used. This reflects the average sentiment of all news articles on a given day, filtered on location. This is done by using the column ’LOCATION’, which consists of a list of all locations found in the text. This column is used to filter the DataFrame on all articles from the US. This approach is based upon the study of (Alamro, McCarren, and Al-Rasheed 2019), more of which can be found in section 2.

3.2 Data Transformation

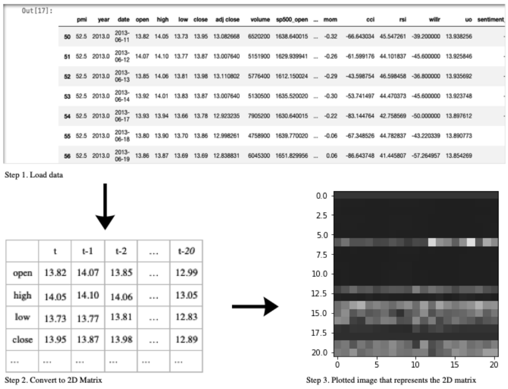
After all data is extracted, a Python script was written which automatically imports all files and merges it into one DataFrame. This DataFrame is be used for further pre-processing.



```
Out[17]:
     pmi    year   date        open   high   low    close  adj.close  volume   sp500_open   ...  mom    cci         rsi        willr       uo         sentimen
50   52.5   2013.0 2013-06-11  13.82  14.05  13.73  13.95  13.082668  6520200  1638.640015  ...  -0.32  -66.643034  45.547261  -39.200000  13.938256
51   52.5   2013.0 2013-06-12  14.07  14.10  13.77  13.87  13.007640  5151900  1629.939941  ...  -0.26  -61.599176  44.101837  -45.600000  13.925846
52   52.5   2013.0 2013-06-13  13.85  14.06  13.81  13.98  13.110802  5776400  1612.150024  ...  -0.29  -43.598754  46.598458  -36.800000  13.935692
53   52.5   2013.0 2013-06-14  13.92  14.01  13.83  13.87  13.007640  5130500  1635.520020  ...  -0.30  -53.741497  44.470373  -45.600000  13.923748
54   52.5   2013.0 2013-06-17  13.93  13.94  13.66  13.78  12.923235  7905200  1630.640015  ...  -0.22  -83.144764  42.758569  -50.000000  13.897612
55   52.5   2013.0 2013-06-18  13.80  13.90  13.70  13.86  12.998261  4758900  1639.770020  ...  -0.06  -67.348526  44.782837  -43.223339  13.890773
56   52.5   2013.0 2013-06-19  13.86  13.87  13.69  13.69  12.838831  6045300  1651.829956  ...   0.06  -86.643748  41.445807  -57.264957  13.854269
```

Figure 2: Merging the data

After the data was downloaded, it was pre-processed. This mainly consists of removing NaN values and normalizing the data into a range between 0 and 1. Unfortunately, not all companies were mentioned in GDELT each day, which resulted in NaN values. Therefore, NaN values in the sentiment column are linearly interpolated. Afterward, all data is normalized between the range (0,1) using the MinMaxScaler from the sklearn library (Pedregosa et al. 2011). In the current study, CNN’s are used to learn temporal relationships between regions, in which the regions do not correspond to specific features of an image, but to time-windows. Therefore, the sequential data will be transformed into a 2D Matrix, where rows represent features and columns represent time steps. This process will be repeated for each day. An overview of this transformation can be found in figure 3.



```
Out[17]:
      pmi   year    date  open   high   low  close  adj close   volume  sp500_open  ...   mom        cci        rsi       willr         uo  sentiment_
50   52.5 2013.0  2013-  13.82  14.05 13.73 13.95  13.082668  6520200  1638.640015 ... -0.32  -66.643034  45.547261  -39.200000  13.938256
                  06-11
51   52.5 2013.0  2013-  14.07  14.10 13.77 13.87  13.007640  5151900  1629.939941 ... -0.26  -61.599176  44.101837  -45.600000  13.925846
                  06-12
52   52.5 2013.0  2013-  13.85  14.06 13.81 13.98  13.110802  5776400  1612.150024 ... -0.29  -43.598754  46.598458  -36.800000  13.935692
                  06-13
53   52.5 2013.0  2013-  13.92  14.01 13.83 13.87  13.007640  5130500  1635.520020 ... -0.30  -53.741497  44.470373  -45.600000  13.923748
                  06-14
54   52.5 2013.0  2013-  13.93  13.94 13.66 13.78  12.923235  7905200  1630.640015 ... -0.22  -83.144764  42.758569  -50.000000  13.897612
                  06-17
55   52.5 2013.0  2013-  13.80  13.90 13.70 13.86  12.998261  4758900  1639.770020 ... -0.06  -67.348526  44.782837  -43.220339  13.890773
                  06-18
56   52.5 2013.0  2013-  13.86  13.87 13.69 13.69  12.838831  6045300  1651.829956 ...  0.06  -86.643748  41.445807  -57.264957  13.854269
                  06-19
```

Step 1. Load data

	t	t-1	t-2	...	t-*20*
open	13.82	14.07	13.85	...	12.99
high	14.05	14.10	14.06	...	13.05
low	13.73	13.77	13.81	...	12.83
close	13.95	13.87	13.98	...	12.89
...	...	...	...	...	...

Step 2. Convert to 2D Matrix

Step 3. Plotted image that represents the 2D matrix

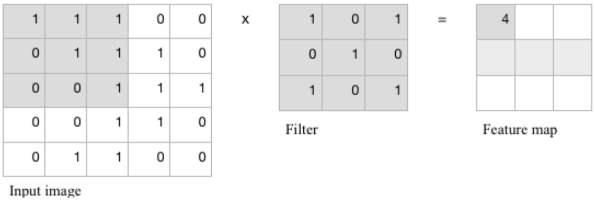
Figure 3: Transformation of the data

In the current research, data from the past 20 days were used to make the matrices. In total, there were 14 variables used from the company- and economic data, which resulted in a matrix of following shape: 14 x 20. Afterwards, the sentiment variables were added as features, which consisted of a total of 3 variables, resulting in 17 x 20 matrices. The target label will be the the movement direction of a stock for the next day. More specifically, the target labels represent whether the closing price of a given stock are either higher or lower the next day. Therefore, the problem best can be described as a binary classification task, as it aims to capture the direction future movements, instead of the exact price. The reason that price movements were chosen, instead of exact values, is because of the fact that exact values can easily lead to false interpretations of the results. This because of the fact that stock prices tend to fluctuate with small values. Therefore, if a model just predicts a value very close to the closing price of the previous day, it usually will have a very low absolute error. However, it will not have any predictive value, as it does not say anything about the actual direction of the stock price. To summarise, the current research will transform sequential data of a company into a 2D matrix. This results in an image which represents an abstract representation of a certain time window, which is fed into a convolutional neural network in order to learn temporal relations between regions of variables that can give a clue about a stock's future value. The corresponding label will be the closing price of a stock the next day.

3.3 Implementation

After the data has been normalized and transformed into images, the data needs to be modeled. The model was created are created in Python by using the Keras library, a library that allows for the creation of sophisticated deep learning models with relative ease (Chollet et al. 2015). The results of the models are plotted using the Python libary Matplotlib cite(Hunter 2007)

3.3.1 Convolutional Neural Networks. Convolutional neural networks are most famous for their image recognition capabilities, in which they learn a hierarchy of feature representations that allow the model to identify key features, which can separate the input into different classes (although they can also be used for regression) (Chollet 2003). A basic convolutional Neural Network usually consists of two key components: a convolutional layer and a pooling layer. In the convolution layer, images are transformed into a feature map by using so-called filters. A filter can best be seen as a matrix representing a connection between a patch of the input layer and a neuron in the hidden layer. The filter slides over the input matrix and applies element-wise matrix multiplication between the filter matrix and the patch of the input image, both of the same dimensions. Afterward, the outputs of this multiplication are summed, and bias is added, resulting in a single value. This process is repeated for all patches in the input layer, resulting in a hidden layer representing a 'feature map' (ibid.). A visualization of the convolution process can be found in figure 1.



$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & & \\ & & \\ & & \end{bmatrix}$$

Figure 4: The convolution process

Afterward, the values are passed to the pooling layer, aiming to reduce dimensionality and spatial variance (Chollet 2003). It does this by pooling together values from its input and transforming it into an output with a lower number of dimensions. An example of a pooling operation is MaxPooling, in which the maximum value of each block is taken as an output value.. Finally, a fully connected dense layer is created at the end of the network to output a prediction. In a binary classification task, it will have a sigmoid function that will output a probability that the instance is of a certain class

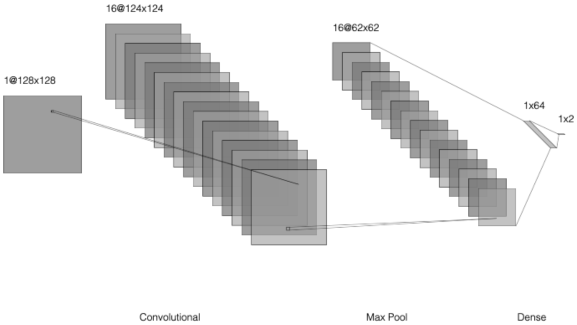


Figure 5: Example of a CNN model with 16 filters

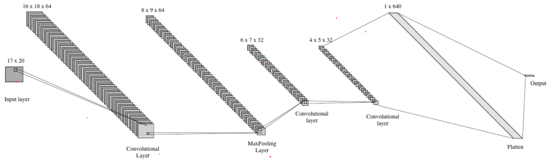
In the current study, a CNN was created which in total consisted of a total of 7-layers. The layers consisted of three convolutional layers, a MaxPooling layer, and a Dropout layer. Since our dataset only consisted of around 1500 instances per company, the number of layers was kept relatively small in order to reduce the number of parameters to avoid over-fitting. In addition, Dropout was used, as this can improve the model's abilities to generalize (Cai et al., 2019). This because the network becomes less sensitive to noise in the data by randomly dropping some nodes. This is very convenient for modeling stock data, as stock data is very noisy by nature. The optimal number of layers and parameters were obtained by trying different configurations on a subset of the companies. These parameters were tested by using the validation set. Other techniques such as Batch Normalization were also considered but were not used, as they did not increase the model's performance while testing on the validation set. The architecture of the used model can be found in table 2.

Table 2: CNN architecture used

Layer Type	Parameters
Convolution	Filters: 64 Kernels: 3x3
Max Pooling	Pooling size: 2,2
Dropout	0.2
Convolution	Filters: 32 Kernels: 3x3
Convolution	Filters: 32 Kernels: 3x3
Flatten	
Dense	2

An important note is that the dimensions of the model change when different sets of variables are used. This because there are fewer features when only the economic data and company data are used, which results in smaller matrices as input. This smaller input also translates to smaller filter dimensions. For training, the model was trained for 50 epochs with a batch size of 16. After experimentation, with different training lengths, it was found that the models tended to overfit when they were trained for a longer time. The optimizer that was used in the model was Adam with a learning rate of 0.001.

Figure 6: the CNN architecture used in the current study



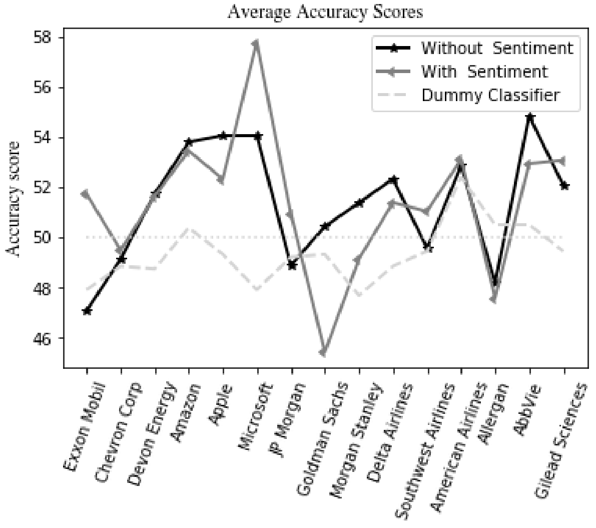
3.4 Evaluation Criteria

The primary evaluation metric in the current study is accuracy. Each final model is trained and tested a total of five times. This because the accuracy scores tend to deviate around a certain value. This is mainly the result of the randomness within parameters of deep learning architectures. As a result, the performance can differ significantly from the initialized weights of different random states. Another reason can be that different nodes are dropped out in the dropout layer. Therefore, training the model one time might give a false indication of how well the model performs. Therefore, each model has been trained and tested five times. The scores were averaged and reported together with the standard deviations. This indicates how well the model performs on average and how the accuracy scores are spread. In this way, it shows how robust the model is to randomness. Because stock prices are the result of very complex mechanisms that are, on their turn, the results of interactions between countless variables, it is nearly impossible to predict stock prices with high accuracy. Therefore, any value above 50% can be considered a good result. The main goal of predicting stock prices is to be slightly better than chance, as that already is enough to gain a competitive advantage over other investors while trading. Besides, the model was also evaluated by using precision, recall, and f1 scores. These scores are averaged over the total number of trails. Also, the results of the model will be compared against a pseudo-random dummy classifier. This baseline classifier assumes that whatever happens today will happen tomorrow and serves as a benchmark to compare the results.

3.5 Results

The current experimental study is carried out on 15 stocks, all listed on the SP 500 in which the closing price of the next day is predicted using Convolutional Neural Networks. A clear overview of the results can be found in the Appendix. In general, the results seem to be mixed.

Figure 7: Average accuracy scores per company



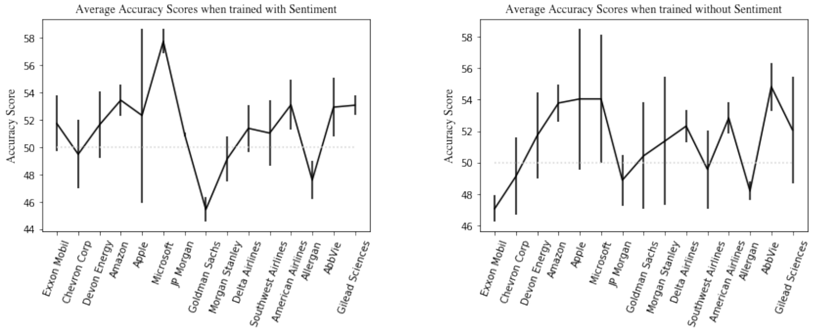
On a company level, the results vary significantly for each company. In general, the model seems to outperform the dummy classifier in terms of accuracy. Only for the companies Goldman Sachs and Allergan did the dummy classifier outperform the model. For 5 out of 15 companies, the predictive capabilities improved when the model was trained with sentiment. Especially for Microsoft and Exxon Mobil, remarkable differences can be found in terms of accuracy. It seems that sentiment improved the model's forecast abilities with around 4 %, where the accuracy scores for Microsoft is the highest of all scores. Smaller increases in performance can be found for Gilead

Table 3: Overview of accuracy scores per industry.

	CNN with sentiment		CNN without sentiment		Baseline
	Accuracy	Std.	Accuracy	Std.	Accuracy
Oil & Energy	50.94	2.32	49.30	2.01	48.5
Technology	54.50	2.81	53.96	3.25	49.20
Banking	48.50	0.89	50.23	3.03	48.74
Airlines	51.84	1.99	51.58	1.49	50.22
Healthcare	51.19	1.41	51.70	1.83	50.15

Sciences, JP Morgan, and Southwest Airlines. On average, the model that is trained with sentiment had the best scores. Another thing worth mentioning is the low performance of the sentiment model for Goldman Sachs. With a score of only 46, the model scores far below chance when it was trained with sentiment. When the model was trained without sentiment, the accuracy score becomes slightly above chance. Moreover, when the model is trained without sentiment, the scores are remarkably better for Morgan Stanley, Delta Airlines, and AbbVie in terms of accuracy. For the other companies, the results are more or less the same.

Figure 8: Accuracy scores plotted with the standard deviation.



The error bars in figure 8 represent the standard deviation of the predictions, which indicates how the amount of variation between the predictions. An interesting note is that the average standard deviation seems to be lower when the model was trained with sentiment, compared to when it was trained without sentiment. One remarkable finding is the high standard deviation for Apple, which indicates an extensive spread in accuracy scores. Besides, there is a big difference in terms of standard deviation for Microsoft. The prediction scores of Microsoft has a very low standard deviation when it was trained on sentiment. However, the standard deviation increases when the model was trained without sentiment. In general, it seems that, except for Apple, the accuracy scores have more variability when the model is trained without sentiment compared to when it is trained with sentiment.

On an industry level, clear differences can be found. It seems that the movement of closing prices in the technology industry can be better predicted than other industries. In three out of five industries, the average scores improve when the model is trained with sentiment. However, it should be noted that these scores lie relatively close together. Only in the oil and energy sector, a difference higher than 1% can be found. In addition, the model shows a decrease in accuracy when it is trained with sentiment for the banking sector. The difference is almost 2%, which is fairly high compared to other sectors.

In Appendix A, a detailed overview can be found of the average scores, together with the averaged F1 scores, recall scores, and precision scores. When looking at the precision scores, big differences can be found. It seems that when the model is trained on sentiment, the precision scores for Allergan, Goldman Sachs and Morgan Stanley drop significantly, resulting in precision scores under 0.1, which can be described as a very low score. On the contrary, very high precision scores can be found for Gilead Sciences, Abbvie, and Microsft. When the models are trained without sentiment, the average precision seems to be significantly higher for Amazon and JP Morgan. However, it significantly decreases for Chevron and Gilead Sciences with around 40 %.

4. Discussion

Combing back to the first research question, one can say that it is possible to a certain extent to predict short term stock movements with company data and economic indicators. The average accuracy results are slightly above chance, while the dummy classifier performs slightly below chance. For our second research question, it seems that that the predictive capabilities of news sentiment vary greatly per company. It seems that in some cases, news sentiment has some predictive power in forecasting stock prices. Especially in the case with Microsoft and Exxon Mobil, there is a big increase in accuracy scores. The increase in accuracy scores of Microsoft might partly be the result due to the fact that it was more often mentioned in GDELT. For example, the GDELT data, which was extracted for Microsoft, contained over more than 1.4 million rows. This is a high amount compared to the data of other companies, such as Apple, that only contained around 400 thousand rows. As a result, the higher amount of data might result in more representative averages of the news sentiment all articles around a given company, which may partly explain these differences. However, this does not explain the differences found for Exxon Mobil, as Exxon Mobil got mentioned 200.000 times, which is less than Chevron, which is mentioned around 250.00 times.

When looking at the average, it seems that sentiment does make the model slightly better in terms of accuracy scores. In addition, it seems that the sentiment does make the model a bit more robust, as the lower standard deviation indicates that the accuracy scores are less spread around a certain average. This can also be seen with Gilead Sciences, where the standard deviation is over three times as small. However, it should be noted that each average score only consists of five trials. Therefore, the results still might not represent the true accuracy scores and can partly be due to chance. Due to the fact that the data is time-series data and therefore not shuffled, the variation in the accuracy scores can't be the result of different sets of the train and test data. Therefore, the smaller standard deviations in terms of accuracy indicate that sentiment makes the model a bit more robust against randomness in weight initialization and regularization and improves the model's capabilities to fit the data to some extent.

When Looking at different industries, it seems that sentiment increases the predictive power of the model in the Oil Energy sector by almost 2 percent. This difference increase is mainly due to the increase in accuracy for Exxon Mobil. For other industries, smaller differences can be found. However, these differences might be too small to make a solid conclusion, as the accuracy scores tended to deviate. An interesting note is a big decrease in accuracy in the Banking Sector. Besides for accuracy scores, the model performs significantly worse in terms of precision, which is especially the case with Morgan Stanley and Goldman Sachs. The low precision indicates that the models are not good in predicting a positive class, which means that in almost all cases, when the network predicted that the price went up, the price actually went down the next day. However, these low precision values disappear when the model is trained without sentiment. It could be the case that the sentiment data for these companies is noisy, which causes the model to find fewer patterns in the data. This noisiness might be caused by the fact that the sentiment extraction algorithm is not specifically tuned for financial articles, causing the sentiment tones to be labeled differently. The effects of this might be bigger for companies that operate only within the financial domains, decreasing its predictive value. However, it could also be the case that the sentiment data results in the need for a different model architecture. This because of the fact that the same model is used for all companies, while it might not be optimal to fit the data for each company. Another thing worth mentioning is the low accuracy score for Allergan. This is by far the worst scoring model in terms of Accuracy, Precision, Recall, and F1 Score, both when it is trained with and without sentiment. In Appendix B, the training accuracy is plotted and compared to that of Amazon. One can clearly see differences in terms of training accuracy. It seems that Allergan does not converge towards a solution, which indicates that the data is too complicated, or a different CNN architecture is needed with different types of layers or parameters.

4.1 Limitations

One major limitation is that it should definitely be taken into account is that most accuracy scores are not very robust. As noted earlier, the average accuracy for all companies is 51.40 when the models are trained with sentiment and 51.35 when they are trained without sentiment. However, the standard deviation is 1.88 and 2.35, respectively. To illustrate, while Apple had an average accuracy score of 52.32, it was found that its lowest testing score was 43%, and its highest score was 57.76 percent. These differences were obtained only by retraining the same model. Therefore, the scores vary to a great extent. Especially in the field of stock prediction, it is important to have a robust score over time as these small differences in accuracy scores can potentially lead to huge losses while algorithmic trading, making the model useless in practice.

Besides the variation in scores, the GDELT data is limited as well. One problem with GDELT is the amount of noise in the data. For example, not all articles are of equal importance for predicting stock prices, as an article on a small website that mentions Apple briefly while reviewing the latest music on Itunes has less predictive value than an article in the Financial Times, where Apple's latest quarterly figures are discussed. However, currently, all articles were weighted equally when the average sentiment tones were extracted. In addition, it was the case sometimes that companies were mentioned as an organization in GDELT, while the article actually was written about something else, and the company was only briefly mentioned. In the current research, limited filtering was done. It might be the case that filtering the data on certain themes or sources might improve the predictive capabilities of GDELT to a great extent. Besides filtering, it might be worth it to also check whether affiliated companies are mentioned. For example, Apple also owns the companies Shazam, Beats Electronics, and AuthenTec. News around these companies might also affect Apple's stock price. Besides, it is not entirely clear whether the effects are due to company sentiment or country sentiment, or both. Therefore, more future research is needed, which takes these limitations into account.

In addition, one major limitation of the current study is that weekends are not taken into account in the research, which can have major implications for the validity and predictive capabilities of the sentiment variables. The reason for this was that the alternative was to interpolate company data in the weekends. However, this would result in a lot of artificial data, which would trouble the predictive capabilities of the model. However, not using all the sentiment data from the weekends results in a lot of information loss. Therefore, it is definitely worth investigating if there are other, better-suited methods to solve this.

Another limitation was the fact that the data was limited. Due to the fact that GDELT's GKG graph only ranges back to April 1, 2013, there were only around 1644 data points. Given the stochastic nature of stock prices, this might be too little data for the model to be able to generalize well. An alternative could be to fit the model again, but this time using intraday stock data, resulting in smaller time intervals, similar to (Gunduz, Yaslan, and Cataltepe 2017) . Another way to approach this problem is that of (Nguyen and Yoon 2019), in which data from multiple stocks was combined to train the model. Afterward, the model was fine-tuned with a small amount of data from the target stock. Although a LSTM model was used for this approach, it is worth investigating whether a similar approach can be used with Convolutional Neural Networks. However, the importance of a network structure increases as the sample size becomes smaller (D'souza, Huang, and Yeh 2020). Due to the fact that there were some difficulties while downloading and the data, there was not much time available to spend on tuning the parameters and trying out different architectures, such as experimenting with the learning rate, optimizer, optimal number of filters or the number of convolutional layers. As a result, the current models might not fit the data well enough to draw solid conclusions. Similarly, another limitation of the current study is that one model is used for all companies. The primary reason for this was the fact that there was simply not enough time to retrain the model for each company with different configurations. However, this can have major implications, as each company is different. Therefore, one architecture might be optimal for one company but does not fit the data of another company. Each company has different data, resulting in different data distributions.

Finally, another thing worth mentioning is that variables have not been clustered on correlation, but roughly grouped based on the category they belonged. It would have definitely been a better approach would have been to cluster the variables based on feature correlation, as a CNN learns spatial relationships between pixel regions. It has been mentioned in the literature that clustering improves the performance of the CNN model (Gunduz, Yaslan, and Cataltepe 2017), (Sezer and Ozbayoglu 2018) . Besides clustering the variables using feature correlation, more time could have been spent on feature selection, as some features might not add any value to the model, which could distort its predictive capabilities. Similarly, because the CNN learns relationships between regions of pixels, only one variable that reflects the sentiment for a company might be too small to make a large difference. Therefore, more types of sentiment variables should be incorporated into the model.

5. Conclusion

The current study investigated to what extent news sentiment from GDELT can improve the capabilities of Convolutional Neural Networks in forecasting stock price movements. In addition, it was checked how the predictive capabilities of news sentiment from GDELT differ across different industries. To investigate this further, a total of fifteen companies were analyzed, ranging from 5 different industries. Subsequently, it was checked to what extent stock movements could be predicted using a 7-layered CNN, which was trained on company data and economic data. The results were compared to a dummy classifier. Afterward, it was checked whether the accuracy scores of the model improved when sentiment variables were added as features. The results indicate that sentiment from GDELT can improve the predictive capabilities of convolutional neural networks, although it should be noted that the results vary greatly per company and industry. Moreover, interesting results were found in terms of variance with respect to the accuracy scores. The results indicated that for some companies, re-training models with the same parameters could drastically change its performance, indicating that the model is not robust, whereas for other companies, the results remained more or less the same, indicating that these models are more robust. The mixed results can partly be attributed to the complex nature of stock prices, but also to limitations in the experimental set-up. Therefore, future research is needed that takes these limitations into account.

6. Self-Reflection

Personally, writing the thesis was a very valuable experience, in which I have learned a lot. One of the main things I learned is that predicting stock prices is incredibly hard and frustrating. Besides learning about python, quantitative finance and deep learning, I also learned a lot about conducting academic research in general. At first, I thought too easy over the topic. After I had written the script to extract the data, I started out with modeling. The first results seemed to be very good, causing me to relax and underestimate the other work that still needed to be done. When the deadline was approaching I noticed some errors in the code, causing a lot of stress. Eventually, I did not have enough time left to do everything I wanted. The main lesson I learned is that projects such as writhing a thesis are easily underestimated, but can take up a huge amount of time. Therefore, it is important not underestimate your topic and be precise in planning. This means that you have to start early and be critical your own on work. Something I will definitely take into account while writing my next thesis.

7. Acknowledgements

I want to thank my supervisor, dr. H. J. Brighton for the feedback and valuable insights during our weekly online meetings. Moreover I'd like to thank my family, friends and girlfriend for feedback and support.

References

Adebiyi, Ayodele Ariyo, Aderemi Oluyinka Adewumi, and Charles Korede Ayo. 2014. Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, 2014:1–7.

Agrawal, Anup and Kishore Tandon. 1994. Anomalies or illusions? evidence from stock markets in eighteen countries. *Journal of International Money and Finance*, 13(1):83–106.

Alamro, Rawan, Andrew McCarren, and Amal Al-Rasheed. 2019. Predicting saudi stock market index by incorporating GDELT using multivariate time series modelling. In *Communications in Computer and Information Science*. Springer International Publishing, pages 317–328.

Audrino, Francesco, Fabio Sigrist, and Daniele Ballinari. 2018. The impact of sentiment and attention measures on stock market volatility. *SSRN Electronic Journal*.

Barberis, Nicholas, Andrei Shleifer, and Robert W. Vishny. 1997. A Model of Investor Sentiment. NBER Working Papers 5926, National Bureau of Economic Research, Inc.

Chollet, François et al. 2015. Keras. https://keras.io.

Chollet, François. 2003. *Deep learning with Python*. Manning Publications Co.

Comlekci, I. and A.Ozer. 2018. Behavioral finance models, anomalies, and factors affecting investor psychology. In *Contributions to Economics*. Springer International Publishing, pages 309–330.

Cookson, Clive. 2016. Winton capital's david harding on making millions through maths.

Corrado, Charles J. and Jr. Thomas W. Miller. 2005. The forecast quality of CBOE implied volatility indexes. *Journal of Futures Markets*, 25(4):339–373.

Dimson, Elroy, Paul Marsh, and Mike Staunton. 2002. *Triumph of the Optimists*. Ofxord University Press.

D'souza, Rhett N., Po-Yao Huang, and Fang-Cheng Yeh. 2020. Structural analysis and optimization of convolutional neural networks with a small sample size. *Scientific Reports*, 10(1).

Fama, Eugene F. 1970. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383.

Gudelek, M. Ugur, S. Arda Boluk, and A. Murat Ozbayoglu. 2017. A deep learning based stock trading model with 2-d CNN trend detection. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE.

Gunduz, Hakan, Yusuf Yaslan, and Zehra Cataltepe. 2017. Intraday prediction of borsa istanbul using convolutional neural networks and feature correlations. *Knowledge-Based Systems*, 137:138–148.

Hirschey, Jeffrey. 2014. Symbiotic relationships: Pragmatic acceptance of data scraping. *SSRN Electronic Journal*.

Hunter, J. D. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Jegadeesh, Narasimhan and Sheridan Titman. 2011. Momentum. *SSRN Electronic Journal*.

Keim, D. 1985. Dividend yields and stock returns: Implications of abnormal january returns. *Journal of Financial Economics*, 14(3):473–489.

Kim, Taewook and Ha Young Kim. 2019. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PLOS ONE*, 14(2):e0212320.

Langsin, K.J. and M Tubbs. 2020. Using sentiment and momentum to predict stock returns. *Research from the Federal Reserve Bank of San Francisco.*

Lara-Benítez, Pedro, Manuel Carranza-García, José M. Luna-Romera, and José C. Riquelme. 2020. Temporal convolutional networks applied to energy-related time series forecasting.

Malkiel, Burton G. 2003. The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, 17(1):59–82.

McKinney, Wes et al. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56, Austin, TX.

Mittelman, Roni. 2015. Time-series modeling with undecimated fully convolutional neural networks.

Nguyen, Thi-Thu and Seokhoon Yoon. 2019. A novel approach to short-term stock price movement prediction using transfer learning. *Applied Sciences*, 9(22):4745.

Pagolu, Venkata Sasank, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi. 2016. Sentiment analysis of twitter data for predicting stock market movements. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System*

(SCOPES), IEEE.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Schwert, G. William. 2003. *Anomalies and Market Efficiency*. Handbook of Economics and Finance.

Sezer, Omer Berat and Ahmet Murat Ozbayoglu. 2018. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70:525–538.

The GDELT Project. 2015. Gdelt 2.0 global knowledge graph codebook (v2.1). Retrieved from: https://blog.gdeltproject.org/gdelt-2-0-our-global-world-in-realtime/.

Vanstone, Bruce James, Adrian Gepp, and Geoff Harris. 2019. Do news and sentiment play a role in stock price prediction? *Applied Intelligence*, 49(11):3815–3820.

Yildirim, Ozal, Ulas Baloglu, and U Acharya. 2019. A deep learning model for automated sleep stages classification using PSG signals. *International Journal of Environmental Research and Public Health*, 16(4):599.

Zuckerman, G. 2019. *The man who solved the market*. New York: Portfolio/Penguin.

Appendix A: Used Features

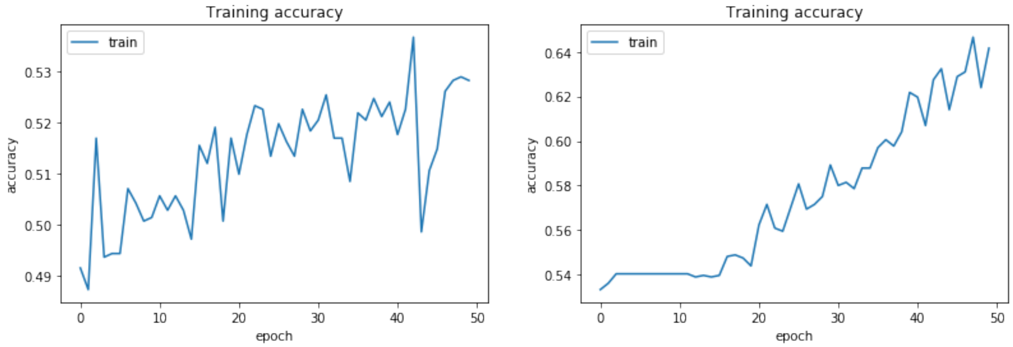
Company Data	Description	Category
Open	Open price on the stock exchange	Company Data
High	Highest price on the stock exchange	
Low	Lowest price on the stock exchange	
Close	Closing price on the stock exchange	
Dividends	Dividends per share	
Adjusted Volatility	Percentual change of a stocks closing price adjusted to the percentual change of the s&p 500	
S&P 500 Open	Opening price of the S&P 500 that day	Economic Indicators
S&P 500 High	Highest price of the S&P 500 that day	
S&P 500 Low	Lowest price of the S&P 500 that day	
S&P 500 Close	Closing price of the S&P 500 that day	
S&P 500 Volatility	Percentual change of the closing price compared compared to the previous day	
VIX Close	Closing value of the VIX index	
PMI	ISM Purchasing Managers Index (PMI index)	
Mean Sentiment	Mean sentiment of all articles where a company where a company is mentioned	Sentiment Data
Result Count	How often a company is mentioned on a given day	
US sentiment	Mean sentiment of all US articles	

Appendix B: Results

	CNN with sentiment					CNN without sentiment					Baseline
	Accuracy	std.	F1	Recall	Precision	Accuracy	std.	F1	Recall	Precision	Accuracy
Exxon Mobil	51.72	2.04	0.38	**0.61**	0.62	47.07	0.84	0.48	0.47	0.51	47.91
Chevron	49.48	2.52	0.60	0.51	0.74	49.13	2.47	0.32	0.40	0.34	48.85
Devon Energy	51.63	2.42	0.51	0.51	0.55	51.72	2.72	0.27	0.51	0.20	48.74
Amazon	53.44	1.16	0.62	0.54	0.73	53.79	1.20	0.67	0.54	0.89	50.38
Apple	52.32	**6.37**	0.63	0.55	0.78	54.05	4.48	0.58	0.58	0.74	49.32
Microsoft	**57.75**	0.9	**0.72**	0.59	**0.94**	54.05	4.08	0.66	0.58	0.80	47.91
JP Morgan	50.95	0.13	0.38	0.52	0.34	48.88	1.63	0.58	0.49	0.73	49.21
Goldman Sachs	45.43	0.9	**0.13**	0.513	**0.08**	50.43	3.39	0.42	0.59	0.356	49.32
Morgan Stanley	49.13	1.65	**0.11**	0.47	**0.06**	51.37	4.06	0.313	0.38	0.38	47.68
Delta Airlines	51.38	1.73	0.61	0.53	0.72	52.32	1.00	0.68	0.53	0.96	48.85
Southwest Airlines	51.03	2.42	0.26	0.53	0.22	49.57	2.48	0.35	0.37	0.46	49.44
American Airlines	53.10	1.83	0.34	0.50	0.30	52.84	1.0	0.33	0.38	0.38	52.38
Allergan	47.58	1.38	**0.06**	**0.08**	**0.05**	48.19	0.6	**0.04**	**0.17**	**0.02**	50.5
AbbVie	52.93	2.17	0.64	0.54	0.79	**54.83**	1.52	0.69	0.55	**0.84**	50.5
Gilead Sciences	53.07	0.7	0.62	0.56	0.81	52.07	3.37	0.38	0.65	0.41	49.44
	Mean										
	51.40	1.88	0.44	0.50	0.52	51.35	2.32	0.45	0.48	0.53	49.44

Appendix C: Plotted training accuracy

Figure 1: Plotted training accuracy



(a) Training Accuracy of Allergan

(b) Training accuracy of Amazon