City University of London

MSc in Data Science

Project Report

2019

A Sentiment Analysis-Based Extension of Piotroski's F-Score Using a LSTM Neural Network

---

Axil Sudra

Supervised by: Dr Kizito Salako

Date of Submission: 13th November 2019

Declaration

By submitting this work, I declare that this work is entirely my own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the assessment instructions and any other relevant programme and module documentation. In submitting this work I acknowledge that I have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. I also acknowledge that this work will be subject to a variety of checks for academic misconduct.

Signed: *Axil Sudra*

Abstract

This project explores whether the sentiment of company management's discussion and analysis (MD&A) narratives can improve the returns of Piotroski's F-Score investment strategy through the use of a long short-term memory (LSTM) architecture. The Compustat database is used to source financial statement data for NYSE listed companies between January 1, 2011 and September 2, 2019. Through using a unique multi-layer perceptron (MLP) architecture, Piotroski's F-Score methodology is automated in a systematic manner. The SEC EDGAR database is scraped for MD&A narratives contained in various company annual filings. The MD&A narratives are labelled with a sentiment polarity using a context-relevant-dictionary or returns method. Finally, an LSTM architecture is created to classify the sentiment of the MD&A narratives.

Keywords: Sentiment Analysis, Piotroski's F-Score, Multi-layer Perceptron, Long Short-term Memory Networks, Value Investing

Contents

1 Introduction and Objectives

1.1 Formatting and Technical Definitions

This paper frequently refers to column/feature names that are contained in various data structures; for ease of identification these identifiers have been put in **bold** font. Furthermore, functions/modules of Python packages are written in *italic* font. A glossary for technical definitions is provided for ease of interpretation.

1.2 Background, Reasoning and Beneficiaries

The concept of value investing has been in existence for a good part of the last century since Graham and Dodd (2009) advocated the idea of identifying undervalued, although profitable companies with the use of quantitative ratios devised from the characteristics of any company's financial statements. Since then, the masses of promising academic research conducted in the field of value investing has influenced professional investment managers to continue experimenting with value strategies to source alpha (excess returns) in unpredictable market conditions. Furthermore, the research has been instrumental in developing value-orientated benchmarks/measures that can be used to gauge for value within a company (Chan and Lakonishok, 2004). A notable example of such a measure is the book-to-market (B/M) ratio, which measures a particular company's book value relative to its market capitalization. Famous economists and researchers, such as (Fama and French, 1992), have reported that a portfolio constructed of companies with high B/M values outperform so-called glamour portfolios. Glamour portfolios are constructed of glamour stocks, which are stocks that are anticipated to have a growth rate higher than the overall market (Asness et al., 2015).

However, Piotroski (2000) reports that returns of the high B/M based investment strategy can be increased by differentiating financially strong high B/M companies from financially weak high B/M companies. Through implementing this methodology, Piotroski shows that value investors can increase their mean annual returns by at least 7.5%. Furthermore, by forming a portfolio around purchasing shares in financially strong high B/M companies (which are referred to as expected winners) and shorting shares in financially weak high B/M companies (referred to as expected losers), investors could have generated an annual return of 23% between the time period 1976 and 1996. However, Piotroski's methodology or any other high B/M investment strategies does not take account of the value of textual data accompanying a particular high B/M company's financial statements; namely, the management's discussion and analysis narrative (MD&A) in a company's annual filings.

Loughran and Mcdonald (2011) suggest that textual analysis on the content of a company's annual filings could "contribute to an investors ability to understand the impact of the information on stock returns". Manual textual analysis by a human on the contents of many annual filings, which a are usually 100+ pages in length, could be an inefficient way of further filtering and picking which stocks to purchase and short of the financially strong and weak high B/M companies resulting from Piotroski's methodology.

This project proposes that if Piotroski's methodology and the process of gathering the sentiment of a company's management's discussion and analysis narrative (from its corresponding annual filings) could be combined and automated with the use of neural networks, then the returns of such a strategy could possibly surpass the majority of value strategies available today.

Thus, this paper attempts to answer the following research question:

Can the sentiment of a company's management's discussion and analysis narrative via a neural network improve the returns of Piotroski's methodology?

The research conducted in this paper could be informative to academic researchers that expertise in the area of value investing and/or data science, and investment management professionals at any level of their career. More specifically, academic researchers could interpret the results of this report, and identify methods to develop any promising work. Entry-level data scientists would be able to develop their knowledge by attempting to reproduce results and gaining an insight into some techniques used in natural language processing/textual analysis. Investment management professionals would be able to gain some knowledge on how data science techniques can be applied to their industry also.

1.3 Objectives

To answer the research question posed in this paper, several various objectives were required to be completed; these are described below.

- **Implement Piotroski's methodology for some given financial statement data, and compute corresponding returns.** Understand the process behind Piotroski's methodology, implement the methodology in a systematic way, and compute the strategies corresponding returns.
- **Automate Piotroski's methodology with the use of suitable neural network architecture.** Identify and create a suitable neural network architecture that produces the results from Piotroski's methodology.
- **Label the sentiment of some given MD&A narrative data.** Identify suitable methods to label the sentiment of some given MD&A narrative data and conduct a feasibility analysis on the methods.
- **Automate the sentiment analysis of some given MD&A textual data.** Identify and create a suitable neural network architecture that has the sole purpose of predicting the sentiment of any given MD&A narrative data.

1.4 Methods for Completing Objectives

The methods for completing the objectives outlined in section 1.3 are discussed below.

- **Implement Piotroski's methodology for some given financial statement data, and compute corresponding returns.** This was achieved by reviewing Piotroski's research paper (2000) in the literature survey (section 2.1.2) and coherently implementing his methodology. The returns of the methodology were computed using data from the Compustat database; excess return was the metric of choice to assess the performance of various portfolio formations.

- **Automate Piotroski's methodology with the use of suitable neural network architecture.** A multi-layer perceptron (MLP) architecture was created to predict the F-Score of some given financial measures for a particular company. It was identified that Piotroski's scoring functions which produced the F-Score value were discontinuous, and thus an MLP was built for each one. The evaluation metric for the overall MLP architecture was a confusion matrix and its corresponding measures (e.g., recall).

- **Label the sentiment of some given MD&A narrative data.** A word count was conducted on any given MD&A narrative for a particular company using the positive and negative word lists from the Loughran and Mcdonald (2011) financial dictionary. Once the word counts for positive and negative words had been obtained, a sentiment scoring formula and corresponding polarity criteria were created to assign any MD&A narrative data a sentiment. Furthermore, a feasibility study was completed on this labelling method by calculating the excess returns of particular portfolio formations. In case the method was deemed infeasible, the MD&A textual data of any particular company would be labelled with a sentiment corresponding to its returns (i.e., if a company's stock price increased over a certain period, its MD&A narrative would be labelled with a positive sentiment).

- **Automate the sentiment analysis of some given MD&A textual data.** A long short-term memory (LSTM) neural network was implemented to read a sequence of numerical MD&A narrative vectors and predict a corresponding sentiment (i.e., positive or negative). As with the MLP, the evaluation metric for the LSTM architecture was a confusion matrix and its corresponding measures (e.g., recall).

1.5 Work Plan

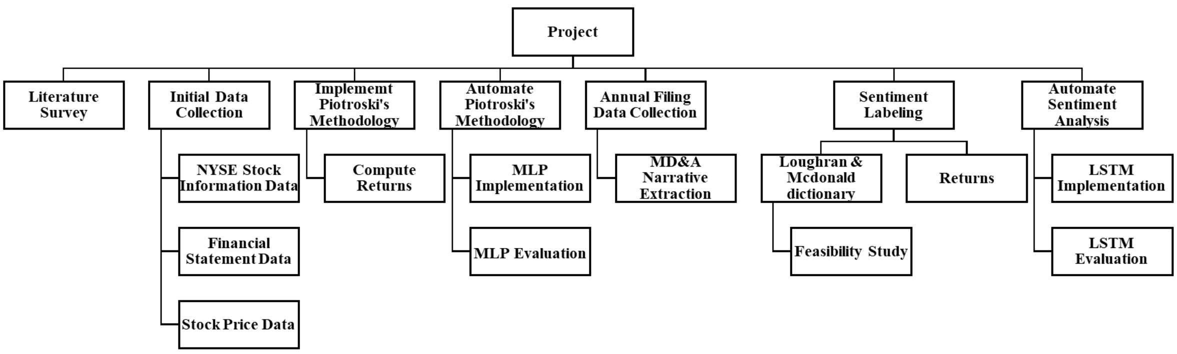

Figure 1: Work plan for the project

The work plan shown in figure 1 was broken down into development stages. The initial data collection and literature survey was completed concurrently. The implementation of Piotroski's methodology was completed in isolation to get a better understanding of the strategy's behaviour. Data collection of the stock returns data and implementation of the MLP were completed in unison. The annual filing data collection stage of the work plan was the longest stage of the project as specific HTML patterns had to be obtained to extract the MD&A narrative of each company at once. There was also a significant amount of manual data processing. The sentiment labelling stage was the second-longest stage of the project as various returns had to be analyzed for the feasibility study. The implementation and evaluation of the LSTM network were not too demanding and thus was completed in a few days. The report was completed throughout the time allocated to different parts of the work plan.

Note that the project was interrupted midway due to unforeseen circumstances which halted work for almost one month and a half. An extension was applied for and granted until November 13, 2019.

1.6 Changes in Goals and Methods

The main goal of this project did not change significantly; instead of developing a sentiment analysis system based on polarity scores, a system based on polarity values (i.e., positive and negative) was developed in addition to implementing Piotroski's methodology using an MLP neural network. All other goals and methods set in the proposal were relatively kept the same. Note that a more meaningful title was given to this project from the title proposed in the original project proposal.

1.7 Structure of Report

Section 1 provides the background to the project and clearly defines a research question to be answered. A breakdown of how the research question was to be answered is provided via objective and method descriptions.

Section 2 provides context surrounding value investing and more specifically Piotroski's methodology. An in-depth review of the procedures involved in conducting sentiment analysis are given, and theory on MLP and LSTM networks is explored.

Section 3 explains the steps involved in producing the work that has been completed; this section gives an in-detail explanation of how components of the project were constructed.

Section 4 provides an overview of the results that were produced from the methods described in section 3.

Section 5 discusses the results in comparison to the project's objectives.

Section 6 evaluates the products of the project and recommends improvements and areas of future work.

2 Context

2.1 Value Investing

2.1.1 The Concept of Value

The foundations of value investing date back to 1934 when Columbia Business School professors Benjamin Graham and David Dodd published ‘Security Analysis: Principles and Technique’, a book which is considered as the “bible of value investing” (Graham and Dodd, 2009). During and before the great depression, investing was primarily speculative. Graham and Dodd began to experiment with a company’s financial statements to determine value, a investment method known as fundamental information analysis or simply fundamental analysis (Lev and Thiagarajan, 1993), and devised the idea of ‘intrinsic value’ - a subjective notion of true value justified by the facts, e.g., the assets, earnings, dividends, definite prospects, irrespective to the company’s share price. In recent years, intrinsic value has been quantified by different formulas, the most common being the present value of a company’s expected future cash flows (Lee et al., 1999) (also known as the discounted cash flow model - DCF) which is given by:

$$\mathrm{DCF} = \frac{CF_1}{(1+r)} + \frac{CF_2}{(1+r)^2} + \frac{CF_3}{(1+r)^3} + \cdots + \frac{CF_n}{(1+r)^n} \tag{2.1.1}$$

where CF_n is the cash flows in period n and r is the discount rate. In successfully determining the intrinsic value of a company, Graham and Dodd advocated the important point that defines value investing today - **purchase securities (e.g., stocks) that are below their intrinsic value and sell securities that are above their intrinsic value**. They also called for investors to maximize their ‘margin of safety’, which is a ratio given by:

$$\mathrm{MoS} = \frac{IV - MV}{IV} \tag{2.1.2}$$

where IV is the company’s intrinsic value and MV is the company’s security market price. In other words, when the market price of a company’s security is considerably below its intrinsic value, the difference is the margin of safety. The margin of safety removes the need for an investor to predict the future as it accounts for most misjudgments; Graham (2003) states: “The function of the margin of safety is, in essence, that of rendering unnecessary an accurate estimate of the future. If the margin is a large one, then it is enough to assume that future earnings will not fall far below those of the past for an investor to feel sufficiently protected against the vicissitudes of time.” However, employing a margin of safety does not result in a successful investment as the idea of intrinsic value is highly subjective.

As well as searching for securities with a significant margin of safety, Graham and Dodd (2009) proposed the use of diverse financial measures to determine the potential future strength of a company. These included:

- Price-To-Book Ratio (P/B) - A financial valuation metric which measures the market value of a company relative to its book value. As a formula, it is given by

$$\text{P/B} = \frac{\text{Market Value per Share}}{\text{Book Value per Share}} \tag{2.1.3}$$

where Market Value per Share is simply equal to a company's stock price and Book Value per Share is the difference between a company's total assets and total liabilities (net assets) divided by its total number of shares outstanding. From equation (2.1.3), a P/B > 1 indicates that a company is overvalued and a P/B < 1 indicates that a company is undervalued.

- Price-To-Earnings Ratio (P/E) - A financial metric which measures the market value of a company relative to its earnings. As a formula, it is given by

$$\text{P/E} = \frac{\text{Market Value per Share}}{\text{Earnings per Share}} \tag{2.1.4}$$

where Market Value per Share is equal to a company's stock price (as mentioned previously) and Earnings per Share is a company's profit divided by its total number of common shares outstanding. From (2.1.4), a high P/E can indicate a company's stock is overvalued and a low P/E can indicate a company's stock is undervalued. Previous research studies (Williamson, 1971; Basu, 1977) suggest the stock of companies with a low P/E tend to outperform those with a high P/E.

- Dividend Yields - A financial metric which measures a company's annual dividend relative to its share price. As a formula, it is given by

$$\text{DY} = \frac{\text{Annual Dividend}}{\text{Share Price}} \tag{2.1.5}$$

A high DY can be the result of high dividend returns relative to a company's share price, or a falling share price. On the other hand, a low DY can be the result of lower dividend returns relative to a company's share price, or a rising share price.

Graham (2003) also advocated the use of qualitative analysis when investing, although in limited quantity; qualitative analysis is based on subjective judgments, and therefore can lead to speculation. Factors such as analyzing the rationale behind management decisions or management structure constitute qualitative methods in investing. Readers should note that value investment strategies are generally quantitative.

2.1.2 Piotroski's F-Score

Since Graham and Dodd's findings, many investment strategies have been proposed that build upon the idea of value. A notable example of such a strategy is Piotroski's F-Score which was developed by Stanford Business School professor Joseph D. Piotroski (2000) in a research paper titled "Value Investing: The Use of Historical Financial Statement Information to Separate Winners from Losers".

Piotroski's F-Score focuses on companies with a high book-to-market (B/M) ratio, which measures the book value of a company relative to its market value. As a formula, it is given by

$$\text{B/M} = \frac{\text{Common Shareholders' Equity}}{\text{Market Capitalization}} \tag{2.1.6}$$

Note that Common Shareholders' Equity is the difference between a company's total assets and total liabilities, and thus equivalent to book value; Market Capitalization is simply a company's total shares outstanding multiplied by its share price. Prior research conducted by various economists, such as Fama and French (1992), show that a portfolio of high B/M companies generally outperforms a portfolio of low B/M companies. Fama and French (1992) explain B/M can be perceived as a variable capturing financial distress, and thus the successive returns represent fair compensation for risk. Despite the impressive returns documented by various research (Fama and French, 1992; Rosenberg et al., 1985; Lakonishok et al., 1994), Piotroski (2000) states "the success of that strategy relies on the strong performance of a few firms, while tolerating the poor performance of many deteriorating companies". Therefore, Piotroski's F-Score discriminates between strong performing companies and those that are 'financially distressed' through the analysis of historical financial statements.

Piotroski (2000) selects nine financial variables (measures) that are based on three fundamental areas of a company's financial condition; these are discussed below.

Profitability

A company's current profitability and cash flow provide useful information on its ability to generate funds internally (Piotroski, 2000). Given that 'value' companies have high B/M values (i.e. the market undervalues their stock prices due to poor historical earnings), any sign of improving or positive profits or cash flow demonstrates the ability to raise funds internally. Piotroski's F-Score uses four financial variables to measure a company's profitability: ROA, CFO, ΔROA, and ACCRUAL. ROA is defined to be a company's net income before any extraordinary items (i.e. gains or losses that have been recorded from rare circumstances) have been included. CFO is defined to be a company's net income before the cash flow from operations has been included. ΔROA is the change in ROA between the current year and previous year. ACCRUAL is defined to be a company's net income for the current year before any extraordinary items less cash flow from operations; note that ACCRUAL is scaled by the company's total assets at the beginning of the year.

Leverage, Liquidity, and Source of Funds

Information regarding leverage and source of funds provide valuable insight into a company's capital structure. Miller and Rock (1985) report the action of raising external capital signals an inability to raise funds internally, especially if a company is financially distressed. Liquidity is also related to a company's external financing habits. Piotroski (2000) states "liquidity is designed to measure a company's ability to meet future debt service obligations". Companies with greater levels of liquidity are more likely to service existing debt obligations or invest at the first-best level without resorting to external finance (Denis, 2011). Piotroski's F-Score uses three financial variables to measure a company's leverage, liquidity and source of funds; ΔLEVER, ΔLIQUID, and EQ_OFFER. ΔLEVER is defined as the change in a company's long-term debt structure; if a company frequently raises funds through external debt services (e.g. issuing corporate bonds) than this can be a sign that it does not have enough internal sources of funds. ΔCLIQUID is defined to be the difference between the current assets and liabilities of the current year and the previous year. EQ_OFFER is defined as the company's total amount of equity (stock) issuance.

Operating Efficiency

Operating efficiency provides detail on a company's ability to generate revenue and control costs. Piotroski (2000) explains the benefits of analyzing the changes in the efficiency of a company's operations, namely obtaining information on the "decomposition of its return on assets". Piotroski's F-Score uses two financial variables to measure a company's operating efficiency; ΔMARGIN and ΔTURN. ΔMARGIN measures the change in a company's gross margin ratio in the current year against the previous year; a high ΔMARGIN signifies a decrease in the cost of sale or an increase in a company's revenue (or a combination of both). ΔTURN measures the change in a company's asset turnover ratio in the current year against the previous year; a high ΔTURN signifies an increase in operating productivity.

From the nine fundamental variables, Piotroski (2000) defines the following scoring variables:

- F_ROA is defined to be one if ROA $>$ 0, and zero otherwise.
- F_CFO is defined to be one if CFO $>$ 0, and zero otherwise.
- F_ΔROA is defined to be one if ΔROA $>$ 1, and zero otherwise.
- F_ACCRUAL is defined to be one if CFO $>$ ROA, and zero otherwise.
- F_ΔLEVER is defined to be one if ΔLEVER $<$ 0, and zero otherwise.
- F_ΔLIQUID is defined to be one if ΔLIQUID $>$ 0, and zero otherwise.
- F_EQ_OFFER is defined to be one if EQ_OFFER $\leq$ 0, and zero otherwise.
- F_ΔMARGIN is defined to be one if ΔMARGIN $>$ 0, and zero otherwise.
- F_ΔTURN is defined to be one if ΔTURN $>$ 0, and zero otherwise.

The F-Score is defined as the sum of the scores given to each of the variables described above:

$$\begin{aligned} \text{F-Score} = \; & \text{F\_ROA} + \text{F\_CFO} + \text{F\_}\Delta\text{ROAF\_ACCRUAL} + \text{F\_}\Delta\text{LEVER} + \text{F\_}\Delta\text{LIQUID} \\ & + \text{F\_EQ\_OFFER} + \text{F\_}\Delta\text{MARGIN} + \text{F\_}\Delta\text{TURN} \end{aligned} \tag{2.1.7}$$

Given there are nine variables, the F-Score can range from 0 to 9. Any company with a low (high) score is expected to have weak (strong) financial performance. Companies that achieve a high score of 8 or 9 are classified as "high F-Score companies" (known as expected winners) and companies that achieve a low score of 0 or 1 are classified as "low F-Score companies" (known as expected losers).

Piotroski (2000) conducts various tests to identify the most optimal portfolio of high B/M companies. The main tests compare the returns earned by high F-Score companies against those of low F-Score companies and the complete portfolio of high B/M companies. The results conclude that high B/M investors can increase their annual mean return by 7.5% through constructing a portfolio of high

F-Score companies, and achieve a 23% annual return by purchasing expected winners and shorting expected losers.

2.2 Sentiment Analysis

2.2.1 The Meaning of Sentiment Analysis

Sentiment analysis is considered to be an important field of Natural Language Processing (NLP), which is the study of how computers can interrupt natural language text or speech (Chowdhury, 2003). Lui (2015) defines sentiment analysis as "the field of study that analyzes people's opinion, sentiments, appraisals, attitudes, and emotions toward entities (e.g., products, services, etc.) and their attributes expressed in written text." The sentiments expressed toward these entities and their attributes can be classified; for example, linguistic polarity categories (e.g., very positive, positive, etc.). In this regard, sentiment analysis can be viewed as a classification exercise where each category represents a sentiment (Prabowo and Thelwall, 2009).

Sentiment analysis can be applied to one of three levels; document-level, sentence-level, or attribute-level (Medhat and Korashy, 2014). Document-level sentiment analysis aims to obtain the sentiment of a complete document or paragraph. Sentence-level sentiment analysis aims to obtain the sentiment of a sentence only. Medhat and Korashy (2014) state that the identification of subjectivity at sentence-level is necessary to perform sentiment classification. Lui (2015) supports this claim by explaining the difference between sentence and document characteristics; a sentence in an 'opinion document' can either be objective or subjective, while the whole document can contain a mixture of objective and subjective sentences. Attribute-level sentiment analysis aims to obtain the sentiment of an entity's attributes. To conduct attribute-level sentiment analysis, it is necessary to identify the attributes of an entity or entities of interest; Schouten and Frasincar (2016) note that attributes of an entity can be identified through a predetermined list or discovered freely. Pang and Lee (2004) show that a sentiment analysis approach with a higher granularity-level (i.e., attribute-level) produces superior classification results than an approach with a lower granularity-level (i.e., document-level).

The existing methods used to implement sentiment classification systems can be classified into thee three main categories shown in figure 2 (Cambria, 2016).

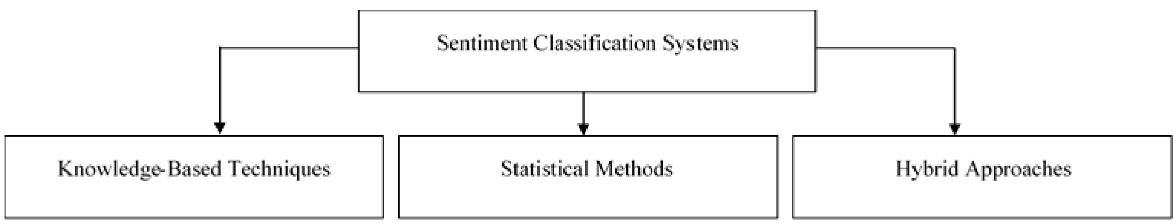

Figure 2: Sentiment analysis methods

2.2.2 Knowledge-Based Techniques

Knowledge-based sentiment analysis techniques aim to classify text using classical NLP methods (e.g., stemming, lemmatization, etc.) and lexicons defining the polarity of words (Melville et al., 2009). Popular lexicons that are frequently used in knowledge-based classification systems include the Affective Lexicon (Ortony et al., 1987), WordNet (Miller et al., 1990), WordNet-Affect (Strapparava and Valitutti, 2004), and SentiWordNet (Esuli and Sebastiani, 2006). Lexicons such as those previously listed are known as 'semantic' dictionaries; they attempt to identify the semantic relationships between words through constructing networks, known as semantic networks, for different parts of speech (e.g., nouns, verbs, etc.) contained in natural language or corpora. Semantic networks aim to "represent concepts expressed by natural-language words and phrases as nodes which are connected to other such concepts by a particular set of arcs called semantic arcs" (Simmons, 1973). Lexicons that simply categorise words or phrases into polarity groups are also frequently used in knowledge-based sentiment analysis in addition to lexicons constructed of semantic networks. An example of a simple knowledge-based lexicon implementation would involve defining a list of positive words and a list of negative words (or using a predefined lexicon), counting the number of positive and negative words that appear in some given text, and classifying the sentiment of the text as positive if the number of positive words is greater than the number of negative words, or negative if vice versa; if the number of positive words is equal to the number of negative words, the sentiment of the text is classified as neutral.

Examples of state-of-the-art knowledge-based sentiment analysis systems include VADER (Hutto and Gilbert, 2014), SO-CAL (Taboada et al., 2014), and LIWC (Tausczik and Pennebaker, 2010). VADER (Valence Aware Dictionary and sEntiment Reasoner) is a large scale rule-based lexicon model designed for sentiment analysis of social media text, although it can be generalized to other domain-specific text; using a semantic orientated lexicon, the system also produces sentiment intensity feedback (i.e. a score for each piece of text). LIWC (Linguistic Inquiry and Word Count), on the other hand, is a large scale lexicon designed to group words into categories and calculate the degree to which each category appears in a corpus. LIWC was originally built for searching and counting words in psychology-relevant categories across multiple documents (Tausczik and Pennebaker, 2010), however, its uses have also been generalized to other domain-specific text. The SO-CAL (Semantic Orientation CALculator) sentiment analysis system is another semantic oriented lexicon similar to VADER, which incorporates more intense negation measures to identify contradictory opinions. Knowledge-based sentiment analysis systems such as the ones previously discussed have proved to be popular due to their accessibility and economy (Cambria, 2016).

Knowledge-based systems do not generally adapt well to different domains (Melville et al., 2009). This is because knowledge-based systems use lexicons that contain words or phrases belonging to word senses that might be deemed different in other domains; for example, the word 'leverage' may correspond to long-term debt levels in a financial lexicon and power of influence in a psychological lexicon. Another limitation of knowledge-based sentiment analysis techniques lies around the validity of such systems. Cambria (2016) states "without a comprehensive knowledge base that encompasses human knowledge it is not easy for a sentiment-mining system to grasp the semantics associated with natural language or human behaviour". Therefore, it is important to have a depth of resources before constructing a knowledge-based sentiment analysis system.

2.2.3 Statistical Methods

Statistical-based sentiment analysis methods classify text through the use of machine learning algorithms. Supervised machine learning classifiers, such as Naïve Bayes, are the most popular for statistical-based sentiment analysis in comparison to unsupervised classifiers (Boiy and Moens, 2009). Kabir et al. (2019) identify the workflow shown in figure 3 for supervised sentiment analysis.

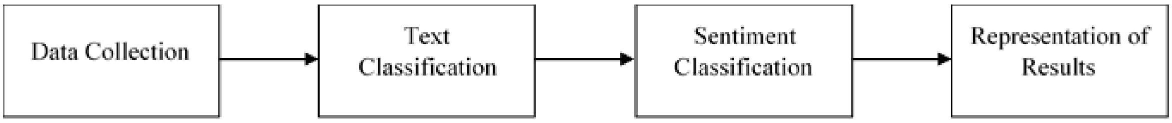

Figure 3: Supervised sentiment analysis workflow

The first step of the workflow is data collection; textual data (e.g., a collection of corpora) is required to conduct sentiment classification. The second step of the workflow is text preprocessing. Similar to knowledge-based sentimental analysis methods, text preprocessing involves applying classical NLP methods for feature extraction; these include techniques such as:

- Tokenization - The process of segmenting a piece of text into individual words, which are known as 'tokens' (Webster and Kit, 1992).
- Stemming - The procedure of removing and replacing suffixes of all words with the same root to a common form, which is known as the 'stem' (Lovins, 1968). For example, the word 'cars' would be reduced to its stem 'car'.
- Lemmatization - The process of removing and replacing suffixes of a word to recover its 'lemma', or more commonly known as dictionary form Lu (2014). For example, the lemma of the word 'walked' would be 'walk'.
- Removing Stop Words - The procedure of removing words that reveal nothing about the content of the piece of text they belong to (Wilbur and Sirotkin, 1992). Examples of stop words include 'the', 'if', 'is', etc.

Text preprocessing also involves organizing extracted features into suitable data representations ready for modelling. A frequent, although naive, data representation approach used for statistical-based sentiment classification is the Bag-of-Words model (BOW). The BOW represents a text document as if it were a bag-of-words, that is, an unordered set of words with their position ignored, keeping only their frequency in the document (Jurafsky and Martin, 2008). Note that the frequencies of each word are stored in a vectorized form and used as input features to train a machine learning classifier. A critical limitation of BOW is that semantic information contained within the document text is essentially lost (Wu et al., 2010). Word embedding models attempt to retain the semantics of text by representing individual words as numerical vectors. A popular word embedding model used in the field of NLP is the Word2Vec model (Mikolov et al., 2013). The Word2Vec model is a two-layer neural network that requires a corpus as input and provides a set of word (feature) vectors as output. The words contained in the 'input' corpus are projected into a high dimensional vector space. The distance between each word vector represents a level semantic relationship. In this regard, the Word2Vec model seems to be an ideal data representation structure for capturing the semantics of text.

However, machine learning classifiers typically require fixed-length inputs meaning that the multiple outputs (i.e., word vectors) from the Word2Vec model would have to be converted into a singular representation; therefore, the use of the Word2Vec model would be computationally expensive for tasks such as organizing features into suitable data representations. The Doc2Vec model proposed by Le and Mikolov (2014) alleviates the Word2Vec model's computational inefficiency and produces a single vector representation for sentences, paragraphs and documents of text. The Doc2Vec model works similarly to the Word2Vec model, although it takes an additional input known as the 'paragraph ID' to group words by document.

The third step of the supervised sentiment analysis workflow is sentiment classification. The preprocessed data representations obtained in step two are divided into a training set and a test set; a chosen machine learning classifier is trained on the training set data and evaluated on the test set data using some loss function. Training techniques, such as k-fold cross-validation and bootstrapping (Kohavi, 1995), are frequently used to improve the classifier's ability to generalize to unseen data. In recent years, deep learning neural networks have become more popular than traditional machine learning algorithms for tasks such as sentiment classification (LeCun et al., 2015). This is mainly due to deep learning algorithms being able to learn detailed feature relationships from raw input data, thus requiring no (or minimal) feature engineering. A significant proportion of previous research on text classification tasks have employed convolutional neural networks (CNN), recurrent neural networks (RNN), or a combination of the two (Santos and Bayser, 2014; Socher et al., 2011; Lai et al., 2015). The last step of the supervised sentiment analysis workflow is the representation of results. This includes extracting meaningful evaluation metrics from the sentiment classifier used in step three and visualizing results. Accuracy, precision, recall and f1 scores are generally used as evaluation metrics for statistical-based sentiment classification systems (Sokolova and Lapalme, 2009).

Through applying the supervised sentiment analysis workflow discussed above, statistical methods can adequately learn to classify the polarity of natural text or speech (Cambria, 2016). However, statistical-based sentiment classification systems have some limitations. Obtaining input features and class labels for statistical-based sentiment analysis methods generally require a large amount of time on any level (Prabowo and Thelwall, 2009). Another limitation of statistical-based sentiment analysis is that they commonly suffer from weak semantic generality (Dang et al., 2010) even with the use of techniques such as Word2Vec and Doc2Vec.

2.2.4 Hybrid Approaches

Hybrid approaches combine knowledge-based techniques and statistical-based methods to perform classification tasks such as emotion recognition and polarity detection in text (Cambria, 2016). An early sentiment classification system that implements a hybrid architecture is presented by König and Bill (2006); they use a two-stage classifier to label the polarity of consumer feedback with a positive or negative sentiment. The first stage consists of checking the consumer feedback documents for any 'discriminating patterns', and if such patterns exist, classifying the documents with the use of a knowledge-based classifier. If a 'discriminating pattern' does not exist, the documents are passed to a machine learning classifier. Prabowo and Thelwall (2009) present multiple hybrid architectures that employ knowledge-based classifiers and statistical-based classifiers on movie reviews, product reviews and social media text; they conclude hybrid approaches result in "better effectiveness" in comparison to individual classifiers. Hybrid approaches to sentiment analysis have the potential of achieving a balance between significant accuracy and a solid understanding of text semantics, thus making them the architecture of choice.

2.3 Theory Surrounding Relevant Neural Networks

2.3.1 Multi-layer Perceptron

A Multi-layer Perceptron (MLP) is a class of supervised feed-forward artificial neural network that consists of an input layer of nodes, one or more hidden layers of nodes, and an output layer of nodes (Haykin, 1999). The nodes of any two layers are linked through connections known as synaptic weights. An MLP with two hidden layers is illustrated in figure 4; note the I denotes input features and O denotes outputs.

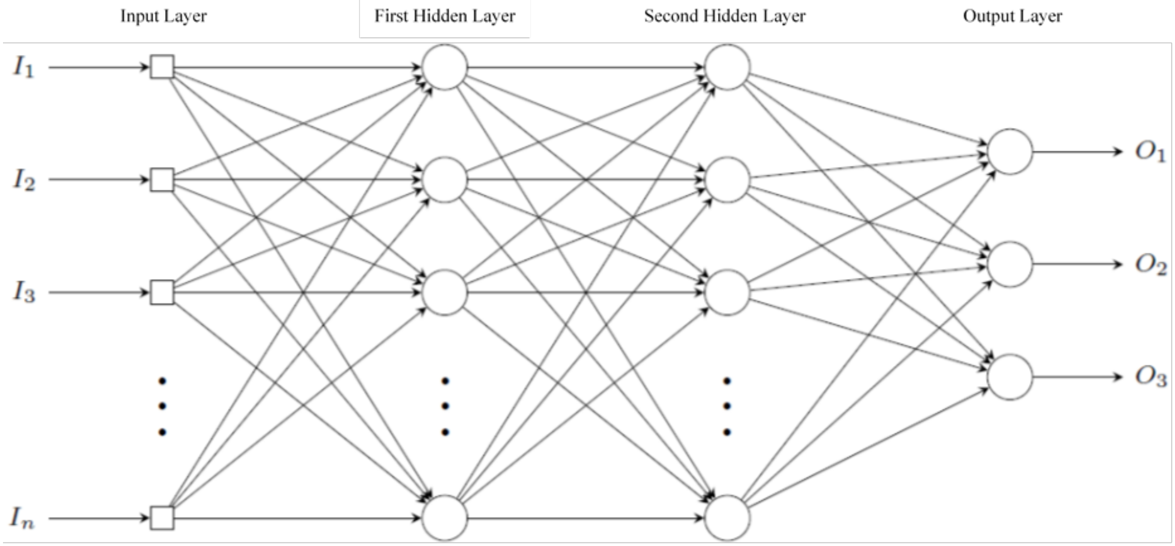

Figure 4: Multi-layer perceptron network with two hidden layers

Nodes that are situated in a hidden layer or output layer of an MLP network are known as artificial neurons or simply neurons. The model of each neuron in the network is constructed of a nonlinear activation function that is differentiable (Haykin, 2009). A neuron can be described by the equation

$$y = \phi(\sum_{j=1}^{m} w_j x_j + b), \tag{2.3.1}$$

where $x_1, x_2, \ldots, x_m$ are the input values to the neuron; $w_1, w_2, \ldots, w_m$ are the values of the synaptic weights connected to the neuron; b is a constant, known as the bias of the neuron; ϕ is the activation function of the neuron; y is the output of the neuron. Figure 5 illustrates the operations of a neuron.

The sigmoid function is frequently used as the activation function for neurons in an MLP network due to its ability to model linear and nonlinear behaviour successfully. Other common activation functions used in MLP networks include the hyperbolic tangent function and the rectified linear units function; these are defined in table 1.

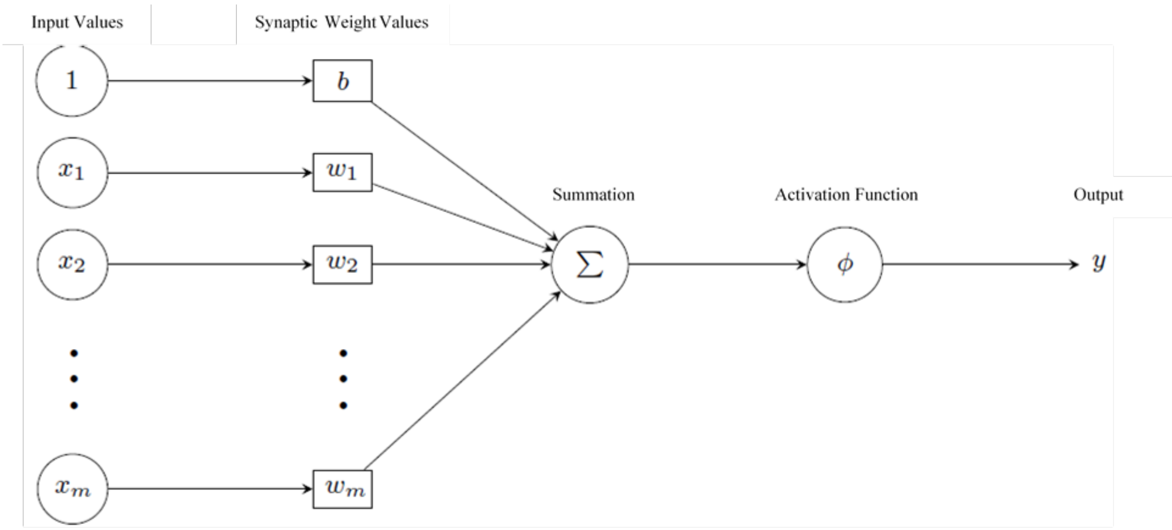

Figure 5: Model of a neuron

Activation Function	Formula
Sigmoid (Logistic)	$\phi(x) = \frac{1}{1 + \exp(-x)}$
Hyperbolic Tangent (Tanh)	$\phi(x) = \frac{\exp(2x) - 1}{\exp(2x) + 1}$
Rectified Linear Units (ReLu)	$\phi(x) = \max(x, 0)$

Table 1: Common Activation Function

Through the use of 'many' hidden neurons constructed of non-linear activation functions, a multilayer feedforward network can approximate any continuous function when given appropriate parameters (Hornik et al., 1989). In this sense, multilayer feedforward networks (including the MLP) can be described as universal approximators and used for predictive modelling tasks such as classification and regression.

MLP networks are trained using a method known as backpropagation Rumelhart et al. (1986). This process attempts to minimize the value of some error function with respect to the networks synaptic weight values with a method known as gradient descent (Andrychowicz, 2016). The choice of the loss function is dependent on whether the purpose of the network is for a classification problem or a regression problem. MLP networks for binary classification problems frequently employ the binary cross-entropy loss as a loss function. MLP networks for multi-class classification problems use an adaption of the binary cross-entropy loss, known as categorical cross-entropy loss. An MLP network designed for regression problems usually uses a continuous loss function, such as the root mean squared error (RMSE), to calculate losses during training.

2.3.2 Long Short-term Memory Networks

A popular artificial neural network used for more difficult tasks such as sentiment analysis is the long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997). Note that the LSTM is a type of recurrent neural network that consists of LSTM units in an LSTM layer. An LSTM unit consists of our vital components; cell state, input gate, forget gate, and output gate.

The cell state is a vector containing a continuous flow of information that originates from n time steps. The input gate decides which new information passing through the unit is going to be added to the cell state. Note that the input gate has a sigmoid activation function to keep its value differentiable. The forget gate also has a sigmoid activation function and is used to decide how much information from the previous cell state should be remembered for future time steps, and how much information should be discarded. The output gate also has a sigmoid activation for the same reason as the input gate and forget gate; this gate decides on how much information should be passed on to the next cell state.

A vector that modifies the cell state value, has a tanh activation function to solve the vanishing/exploding gradient problem (Sundermeyer et al., 2012) that exists in gradient-based training algorithms such as back-propagation. Due to the memory operations in the units, LSTMs are ideal neural networks for modelling sequence data, such as speech generation, sentiment analysis, time-series prediction, etc.

3 Methods

3.1 Financial Statement Data Collection and Manipulation

3.1.1 NYSE Listings Data

The data collected for this project is based on non-financial companies[1] listed on the New York Stock Exchange (NYSE). A comma-separated values (CSV) file of companies listed on the NYSE was sourced from the National Association of Securities Dealers Automated Quotations (NASDAQ) website (Nasdaq Inc., 2019). The CSV file consisted of 9 features (refer to table 2) and $3,123$ rows and was 422 kilobytes in size.

Feature Name	Feature Description
Ticker_Symbol	Ticker symbol used to identify a particular stock.
Comapny_Name	Name of particular company.
Last_Sale	The most recent sale price quoted for a particular stock. Quoted in US dollars ($).
Market_Capitalization	The total market value of a company's outstanding shares of stock. Quoted in US dollars ($)
ADR_TSO	Notation used to identify whether a particular stock is classed as an American Depository Receipt.
IPO_Year	The year a particular company's stock is floated or listed on the NYSE.
Sector	The economic area in which companies share identical or similar product or services.
Industry	A group of companies that are related based on their similar operations.
Summary_Quote	URL to particular stock quote summary page on the NASDAQ website.

Table 2: Description of features in NYSE stock listing data set

The data contained in the NYSE listing CSV file was imported into a data frame using the *read_csv* module of the Python package Pandas (McKinney, 2010). Information about the features was retrieved using the Pandas data frame function *info()*. The information revealed **Ticker_Symbol**, **Company_Name**, **Sector**, **Industry**, and **Summary_Quote** were object data types; **Last_Sale**, **Market_Capitalization**, **ADR_TSO**, and **IPO_Year** were float64 data types. Note that Pandas uses the object data type to store strings (McKinney and PyData, 2019). **Last_Sale**, **Market_Capitalization**, **ADR_TSO**, **IPO_Year**, and **Summary_Quote** were identified as redundant features, and thus removed from the data frame.

The total number of missing values in the data frame was identified through the use of the Pan-

[1] Financial companies are excluded because the high leverage that is common for these companies probably does not have the same meaning as for non-financial companies, where high leverage more likely indicates distress (Fama and French, 1992).

das functions *isnull()* and *sum()*; 1006 missing values were located and removed using the Pandas *dropna()* function. The indexes of rows corresponding to financial companies were located via the **Sector** column using the Pandas *loc* functionality and removed from the data frame. As a result of removing missing values and rows corresponding to financial companies, the data frame had $1,765$ remaining rows.

3.1.2 Financial Statement Data

To calculate Piotroski's F-Score (Piotroski, 2000), specific financial statement data was sourced from the Standard and Poor's Compustat Daily Updates - Fundamentals Annual data table (Wharton Research Data Services, 2019). The Compustat Daily Updates - Fundamentals Annual data table is accessible via Wharton Research Data Services' (WRDS) website or the Python package WRDS-Py. For this project, WRDS-Py was used to gain direct access to the data table in JupyterLab. Note that a WRDS username and password are required when importing the WRDS-Py package in Python.

All WRDS data is stored in a PostgreSQL database, and thus can be obtained through SQL statements via WRDS-Py. The following SQL statement was used to retrieve a Pandas data frame containing 15 features (refer to table 3) from the Compustat Daily Updates - Fundamentals Annual library:

```
SELECT DISTINCT cik, tic, conm, datadate, mkvalt, at, lt,
ibadj, oancf, dltt, act, lct, csho, sale, cogs,
FROM compa.funda WHERE tic IN %(ticker_symbol)s
AND datadate >= '01/01/2009'
AND datadate <= '08/01/2019'
AND cik IS NOT NULL
AND mkvalt IS NOT NULL
AND at IS NOT NULL
AND lt IS NOT NULL
AND ibadj IS NOT NULL
AND oancf IS NOT NULL
AND dltt IS NOT NULL
AND act IS NOT NULL
AND lct IS NOT NULL
AND csho IS NOT NULL
AND sale IS NOT NULL
AND cogs IS NOT NULL
```

where compa.funda was the identifier for the Compustat Daily Updates - Fundamentals Annual data table, and ticker_symbol was the key of a Python dictionary which stored the observations of the **Ticker_Symbol** feature in the NYSE listing data frame. The data was obtained for the period between January 1, 2009, and August 1, 2019 (i.e., 10 years and 7 months). This data frame was referred to as the fundamentals data frame and contained $10,109$ rows.

The **Sector** feature from the NYSE listing data frame was added to the fundamentals data frame to conduct sector-based analysis later. This increased the number of features in the fundamentals data frame from 15 to 16. As with the NYSE listing data frame, the Pandas *info()* function was used to retrieve information about the features contained in the fundamentals data frame. The information revealed **CIK**, **Ticker_Symbol**, **Company_Name**, **Data_Date**, and **Sector** were object data types, and the remaining features were float64 data types. Fortunately, the fundamentals data frame did not contain any missing values.

Compustat Feature Name	New Feature Name	Feature Description
cik	**CIK**	A unique identification number given to an individual company by the SEC.
tic	**Ticker_Symbol**	Refer to table 2.
conm	**Company_Name**	Refer to table 2.
datadate	**Data_Date**	Date corresponding to a particular company's fiscal year-end.
mkvalt	**Market_Capitalization**	The total market value of a company's outstanding shares of stock at its fiscal year-end.[2]
at	**Total_Assets**	The total assets of a particular company at its fiscal year-end.[2]
lt	**Total_Liabilities**	The total liabilities of a particular company at its fiscal year-end.[2]
ibadj	**Net_Income_Before_Extra_Items**	Net income excluding extraordinary items of a particular company at its fiscal year-end.[2]
oancf	**Cash_Flow_From_Operations**	The amount of income that a particular company generates from regular operating activities at its fiscal year-end.[2]
dltt	**Total_Long_Term_Debt**	The total amount of long term debt of a particular company at its fiscal year-end.[2]
act	**Current_Assets**	The total amount of current assets of a particular company at its fiscal year-end.[2]
lct	**Current_Liabilities**	The total amount of current liabilities of a particular company at its fiscal year-end.[2]
csho	**Common_Shares_Outstanding**	The total amount of a company's common shares outstanding at its fiscal year-end.[2]
sale	**Total_Sales**	The total amount of sales made by a particular company at its fiscal year-end.[2]
cogs	**Cost_Of_Goods_Sold**	The cost of goods sold by a particular company at its fiscal year-end.[2]

Table 3: Compustat feature names, new feature names, and description of features in fundamentals data set

[2] Quoted in US dollars ($); units in millions.

3.2 Implementing Piotroski's Methodology

3.2.1 Computing Book-to-Market Ratios and Identifying Book-to-Market Quintile Cutoffs

The first step in Piotroski's methodology is to identify data observations with high book-to-market (B/M) ratios. Therefore, it was necessary to calculate the B/M ratio for data observations contained in the fundamentals data frame. The following formula was used to compute the B/M ratio for each data observation in the fundamentals data frame:

$$\text{B/M Ratio} = \frac{\textbf{Total\_Assets} - \textbf{Total\_Liabilities}}{\textbf{Market\_Capitalization}} \tag{3.2.1}$$

These values were stored in the fundamentals data frame under a new feature/column called **BM_Ratio**. Just as Piotroski (2000), B/M ratio quintile cutoffs for the data observations in the fundamentals data frame were identified with the use of the *qcut*() function in Pandas. The quintile cutoffs had the following labels: very low, low, medium, high, and very high. The quintile values for each data observation were stored in the fundamentals data frame under a new column/feature called **BM_Quintile**. The mean, minimum, and maximum B/M ratio for each quintile is shown in table 4.

BM_Quintile	Mean BM_Ratio	Minimum BM_Ratio	Maximum BM_Ratio
Very Low	−0.74	−721.17	0.22
Low	0.30	0.22	0.37
Medium	0.45	0.37	0.53
High	0.65	0.53	0.80
Very High	1.75	0.80	199.68

Table 4: Mean, minimum, and maximum B/M ratio by B/M quintile (to 2.d.p.)

From table 4, it is evident that the very low **BM_Quintile** contains a significant amount of negative B/M ratios. A count on the data observations with negative B/M ratios was completed using the *count()* function of Pandas. This revealed there was a total of 395 data observations with a negative B/M ratio in the fundamentals data frame. These values could have potentially skewed/distorted future results of this project, and thus were removed using the Pandas *drop()* function. The mean, minimum, and maximum B/M ratio for each quintile after removing data observations with a negative B/M ratio is shown in table 5.

BM_Quintile	Mean BM_Ratio	Minimum BM_Ratio	Maximum BM_Ratio
Very Low	0.15	0.00	0.25
Low	0.32	0.25	0.39
Medium	0.46	0.39	0.55
High	0.67	0.55	0.82
Very High	1.79	0.82	199.68

Table 5: Mean, minimum, and maximum B/M ratio by B/M quintile after removing data observations with a negative B/M ratio (to 2.d.p.)

3.2.2 Identifying Company Size Tercile Cutoffs

The second step in Piotroski's methodology is to identify company size tercile cutoffs with the use of market capitalization. Therefore, company size tercile cutoffs were identified using the *qcut*() function in Pandas with the quantiles parameter equaling 3. The tercile cutoffs had the following labels: small, medium, and large. These values were stored in the fundamentals data frame under a new column/feature called **Size_Tercile**. The mean, minimum, and maximum market capitalization for each tercile is shown in table 6; note that market capitalization is abbreviated 'MC'.

Size_Tercile	Mean MC	Minimum MC	Maximum MC
Small	648.58	3.19	1409.53
Medium	2981.26	1411.25	5542.94
Large	29840.45	5543.16	438702.00

Table 6: Mean, minimum, and maximum market capitalization by company size tercile (to 2.d.p.)

3.2.3 Removing Companies with Two or Fewer Years of Data

To calculate Piotroski's F-Score, a company must have more than 2 years of data; this is because various fundamental measures that are crucial to the F-Score calculation require data from 2 years ago. An example of such a fundamental signal is the change in leverage feature, which is calculated in section 3.2.5. The removal of such companies was completed using the Pandas *groupby()* function on **Ticker_Symbol**, and filtering groups that contained more than 2 observations through applying the *lambda* function with *len()* > 2 as the expression. Once completed, $9,513$ rows remained.

3.2.4 Creating Additional Features for Ease of Access

As mentioned in section 3.2.3, various fundamental measures require data from 2 years ago, as well as 1 year ago. For ease of access when calculating Piotroski's 9 fundamental measures, the columns/features in table 7 were created in the fundamentals data frame. These features were created through grouping by the **Ticker_Symbol** feature and applying the *lambda* function with *shift(x)* as the expression. Note that the variable *x* represents the number of years a feature was to be shifted.

Feature Name	Description
Total_Assets_PY1	**Total_Assets** value from 1 year ago.
Total_Assets_PY2	**Total_Assets** value from 2 years ago.
Total_Long_Term_Debt_PY1	**Total_Long_Term_Debt** value from 1 year ago.
Current_Assets_PY1	**Current_Assets** value from 1 year ago.
Current_Liabilities_PY1	**Current_Liabilities** value from 1 year ago.
Common_Shares_Outstanding_PY1	**Common_Shares_Outstanding** value from 1 year ago.
Total_Sales_PY1	**Total_Sales** value from 1 year ago.
Cost_Of_Goods_Sold_PY1	**Cost_Of_Goods_Sold** value from 1 year ago.

Table 7: Additional feature names and description of features in fundamentals data set

3.2.5 Calculation of Nine Fundamental Measures

The third step in Piotroski's methodology is to calculate 9 fundamental measures of a company's financial strength. These are shown below with their corresponding formula:

$$\text{Return on Assets (ROA}_t) = \frac{\text{Net Income before Extraordinary Items}_t}{\text{Total Assets}_{t-1}} \tag{3.2.2}$$

$$\text{Cash Flow from Operations (CFO}_t) = \frac{\text{Cash Flow from Operations}_t}{\text{Total Assets}_{t-1}} \tag{3.2.3}$$

$$\text{Change in Return on Assets } (\Delta\text{ROA}_t) = \text{ROA}_t - \text{ROA}_{t-1} \tag{3.2.4}$$

$$\text{Accruals (ACCRUAL}_t) = \text{ROA}_t - \text{CFO}_t \tag{3.2.5}$$

$$\begin{aligned}\text{Change in Leverage } (\Delta\text{LEVER}_t) = {} & \frac{\text{Long-Term Debt}_t}{\frac{1}{2}(\text{Total Assets}_t + \text{Total Assets}_{t-1})} - \\ & \frac{\text{Long-Term Debt}_{t-1}}{\frac{1}{2}(\text{Total Assets}_{t-1} + \text{Total Assets}_{t-2})}\end{aligned} \tag{3.2.6}$$

$$\text{Change in Liquidity } (\Delta\text{LIQUID}_t) = \frac{\text{Current Assets}_t}{\text{Current Liabilities}_t} - \frac{\text{Current Assets}_{t-1}}{\text{Current Liabilities}_{t-1}} \tag{3.2.7}$$

$$\begin{aligned}\text{Equity Offering (EQ\_OFFER}_t) = {} & \text{Common Shares Outstanding}_t - \\ & \text{Common Shares Outstanding}_{t-1}\end{aligned} \tag{3.2.8}$$

$$\begin{aligned}\text{Change in Margin } (\Delta\text{Margin}_t) = {} & \frac{\text{Total Sales}_t - \text{Cost of Goods Sold}_t}{\text{Total Sales}_t} - \\ & \frac{\text{Total Sales}_{t-1} - \text{Cost of Goods Sold}_{t-1}}{\text{Total Sales}_{t-1}}\end{aligned} \tag{3.2.9}$$

$$\text{Change in Turnover } (\Delta\text{TURN}_t) = \frac{\text{Total Sales}_t}{\text{Total Assets}_{t-1}} - \frac{\text{Total Sales}_{t-1}}{\text{Total Assets}_{t-2}} \tag{3.2.10}$$

Note that t represents a period (e.g. years).

The formulas above were computed by grouping by the **Ticker_Symbol** feature and applying the *lambda* function to the relevant columns/features in the fundamentals data frame. The following column/features were created in the fundamentals data frame to store the values resulting from the equations (3.2.3) to (3.2.10): **ROA**, **CFO**, **CROA**, **ACCRUAL**, **CLEVER**, **CLIQUID**, **EQ_OFFER**, **CMARGIN**, and **CTURN**. The Pandas *dropna()* function was used to drop any rows containing any null values.

3.2.6 Implementation of Nine Scoring Functions and Calculation of F-Score

The final step in Piotroski's methodology is to create scoring functions for each of the 9 fundamental measures described in section 3.2.5 and calculate the F-Score by summing the results of the scoring functions. The 9 scoring functions and the F-Score formula are shown below:

$$\text{F\_ROA}_t = \begin{cases} 1, & \text{if } \text{ROA}_t > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.11}$$

$$\text{F\_CFO}_t = \begin{cases} 1, & \text{if } \text{CFO}_t > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.12}$$

$$\text{F\_}\Delta\text{ROA}_t = \begin{cases} 1, & \text{if } \Delta\text{ROA}_t > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.13}$$

$$\text{F\_ACCRUAL}_t = \begin{cases} 1, & \text{if } \text{ACCRUAL}_t < 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.14}$$

$$\text{F\_}\Delta\text{LEVER}_t = \begin{cases} 1, & \text{if } \Delta\text{LEVER}_t < 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.15}$$

$$\text{F\_}\Delta\text{LIQUID}_t = \begin{cases} 1, & \text{if } \Delta\text{LIQUID}_t > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.16}$$

$$\text{F\_EQ\_OFFER}_t = \begin{cases} 1, & \text{if } \text{EQ\_OFFER}_t \leq 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.17}$$

$$\text{F\_}\Delta\text{MARGIN}_t = \begin{cases} 1, & \text{if } \Delta\text{MARGIN}_t > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.18}$$

$$\text{F\_}\Delta\text{TURN}_t = \begin{cases} 1, & \text{if } \Delta\text{TURN}_t > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2.19}$$

$$\begin{aligned} \text{F-Score} = \text{F\_ROA}_t + \text{F\_CFO}_t + \text{F\_}\Delta\text{ROA}_t + \text{F\_ACCRUAL}_t + \text{F\_}\Delta\text{LEVER}_t + \\ \text{F\_}\Delta\text{LIQUID}_t + \text{F\_EQ\_OFFER}_t + \text{F\_}\Delta\text{MARGIN}_t + \text{F\_}\Delta\text{TURN}_t \end{aligned} \tag{3.2.20}$$

As with section 3.2.5, the 9 scoring functions in equations (3.2.11) to (3.2.19) were computed by grouping by the **Ticker_Symbol** feature and applying the *lambda* function to relevant columns/features in the fundamentals data frame. The expressions used in the *lambda* functions were predominately summations and subtractions. The following columns/features were created in the fundamentals data frame to store the values resulting from the 9 scoring functions: **F_ROA**, **F_CFO**, **F_CROA**, **F_ACCRUAL**, **F_CLEVER**, **F_CLIQUID**, **F_EQ_OFFER**, **F_CMARGIN**, and **F_CTURN**. The F-Score for each data observation was then computed for each data observation in the fundamentals data frame, and stored in a new column/feature called **F_SCORE**.

3.3 Security Price Data Collection and Manipulation

3.3.1 Stock Price Data

To evaluate Piotroski's methodology on the sample of very high B/M data observations contained in the fundamentals data frame, stock price data was sourced from the Standard and Poor's Compustat Daily Updates - Security Daily data table (Wharton Research Data Services, 2019). As with the financial statement data in section 3.1.2, access to the Security Daily data table was obtained through the Python package WRDS-Py.

The first step in obtaining the stock price data for the data observations in the fundamentals data frame was to retrieve the purchase date. A new data frame was created to store the stock price information of each data observation of the fundamentals data frame; this new data frame was referred to as the stock price data frame. The **Ticker_Symbol**, **Company_Name**, **Data_Date**, **BM_Quintile**, and **F_SCORE** columns/features from the fundamentals data frame were copied to the stock price data frame. Furthermore, only the very high B/M data observations were kept through filtering via the **BM_Quintile** column/feature.

As financial statement data for an NYSE listed company is not available immediately at its fiscal year-end, the date that its data is finalized (and available to the public) was retrieved from the Compustat Daily Updates - Fundamentals Annual data table using the following SQL statement:

```
SELECT DISTINCT fdate
FROM compa.funda WHERE tic=%s
AND datadate=%s
```

where **Ticker_Symbol** was passed as the parameter for tic, and **Data_Date** was passed for datadate. Note that the **Data_Date** column/feature of the stock price data frame was converted to a timestamp object with the Pandas *to_datetime()* function before using as a parameter to the SQL statement above. These values were stored in a new column/feature in the stock price data frame called **Purchase_Date**. If the result of the SQL statement was null for a particular data observation, its **Purchase_Date** value would be set to 'N/A'. 16 data observations had a **Purchase_Date** value of 'N/A', and 2 data observations that had a **Purchase_Date** value that was null. Of the 18 data observations that did not have a **Purchase_Date** value, 16 were successfully sourced and manually input into the stock price data frame. A check was conducted on the **Purchase_Date** values for each data observation to see whether they were a US working day; this was done with the use of the *is_working_day()* function from the Python package Workalender (Bord, 2019). Any data observations with **Purchase_Date** values falling on US holidays were amended to the next closest working day with the Workalender function *add_workings_day()*.

The next step was to compute the sell price for each data observation. As the observations in the stock price data frame correspond to annual financial statement data, the sell date was set to the **Purchase_Price** feature plus one year. The *datetime* and *timedelta* modules were used from the Python package Datetime (Ansell, 2016) to carry out operations on dates. If a data observation had a **Purchase_Date** greater than or equal to September 1, 2018, then its sell date would be set to September 2, 2019, as this was the date when the returns data was extracted. The sell date values for the data observation were stored in a new column/feature called **Sell_Date** in the stock price data frame. As with the **Purchase_Date** column/feature, the **Sell_Date** values for each data observation were checked to see whether they were a US working day. Any data observations with **Sell_Date** falling on US holdings were amended to the next closest working day. The **Sell_Date** was set to 'N/A' for any data observations that had a **Purchase_Date** value of 'N/A'; these were then removed from the stock price data frame.

The final step in obtaining the stock price data for the observations in the stock price data frame was to retrieve the purchase and sell price. As mentioned at the beginning of this section, stock price data was retrieved from the Compustat Daily Updates - Security Daily data table; the following SQL statement was used to obtain the purchase and sell price for the data observations in the stock price data frame:

```
SELECT DISTINCT prccd
FROM compa.secd WHERE tic=%s
AND datadate=%s
```

where compa.secd was the identifier for the Compustat Daily Updates - Security Daily data table, **Ticker_Symbol** was passed as the parameter for tic, and **Purchase_Date/Sell_Date** was passed for datadate depending on whether the stock purchase price or stock sell price was being retrieved. If the result from the SQL statement was null for a data observation, its purchase/sell price would be retrieved from Yahoo Finance via the Python package Pandas-Datareader (PyData, 2019). Any stock purchase price and/or stock sell price values that were not able to be obtained were set to 'N/A'. The stock purchase price and stock sell price values were stored in new columns/features called **Stock_Purchase_Price** and **Stock_Sell_Price** respectively in the stock price data frame. There were 48 data observations with **Stock_Purchase_Price** and/or **Stock_Sell_Price** values of 'N/A'; of the 48, 46 were successfully sourced and input manually into the stock price data frame.

After the above had been completed, the stock price data frame contained 1,455 rows. Of the whole data frame, the stock purchase price and/or stock sell price for 2 data observations were 'N/A'.

3.3.2 SPDR S&P 500 ETF Data

To compare the returns of the data observations in the stock price data frame, pricing data for the SPDR S&P 500 ETF was extracted from the Compustat Daily Updates - Security Daily data table. A similar SQL statement was used to extract the purchase price and sell price of the SPDR S&P 500 ETF to that of the individual data observations in section 3.3.1. Purchase prices and sell prices were obtained for the SPDR S&P 500 ETF corresponding to each data observation's **Purchase_Date** and **Sell_Date**; these values were stored in new columns/features called **SPDR_Purchase_Price** and

SPDR_Sell_Price respectively in the stock price data frame. By storing SPDR S&P 500 ETF pricing data for each row of the stock price data frame, each data observation could be compared at a highly granular level.

3.4 Back-Testing Piotroski's Methodology

3.4.1 Calculation of Returns

Various returns were calculated to assess the performance of the data observations in the stock price data frame; these are described below.

The holding period return (HPR) on a long position of each data observation in the stock price data frame was computed using the following formula:

$$\text{Long Position HPR} = \frac{\textbf{Stock\_Sell\_Price}}{\textbf{Stock\_Purchase\_Price}} - 1 \tag{3.4.1}$$

The long position HPR for each data observation was stored in a new column/feature called **Long_-Holding_Period_Return_(%)** in the stock price data frame.

The HPR on a short position of each data observation in the stock price data frame was computed using the following formula:

$$\text{Short Position HPR} = \frac{\textbf{Stock\_Purchase\_Price}}{\textbf{Stock\_Sell\_Price}} - 1 \tag{3.4.2}$$

The short position HPR for each data observation was stored in a new column/feature called **Short_-Holding_Period_Return_(%)** in the stock price data frame.

The annualized HPR on a long position of each data observation in the stock price data frame was computed using the following formula:

$$\text{Long Position Annualized HPR} = (1 + \text{Long Position HPR})^{(\frac{365}{h} - 1)} \tag{3.4.3}$$

where h represents the total number of days a data observation was held (i.e. **Sell_Date** − **Purchase_Date**). The long position HPR for each data observation was stored in a new column/feature called **Long_Annualized_Holding_Period_Return_(%)** in the stock price data frame.

The annualized HPR on a short position of each data observation in the stock price data frame was computed using the following formula:

$$\text{Short Position Annualized HPR} = (1 + \text{Short Position HPR})^{(\frac{365}{h} - 1)} \tag{3.4.4}$$

where h represents the total number of days a data observation was held (i.e. **Sell_Date** − **Purchase_Date**). The short position HPR for each data observation was stored in a new column/feature called **Short_Annualized_Holding_Period_Return_(%)** in the stock price data frame.

Formulas (3.4.1) to (3.4.4) were also applied to the SPDR S&P 500 columns/features in the stock price data frame, and stored in new columns/features called **SPDR_Long_Holding_Period_Return_(%)**, **SPDR_Short_Holding_Period_Return_(%)**, **SPDR_Long_Annualized_Holding_Period_Return_(%)**, and **SPDR_Short_Annualized_Holding_Period_Return_(%)**.

The annualized excess return on a long position of each data observation in the stock price data frame was computed using the following formula:

$$\text{Long Position Annualized Excess Return} = x - y \tag{3.4.5}$$

where x represents the long position annualized HPR on a data observation, and y represents the long position annualized HPR on the SPDR S&P 500 ETF. The long position annualized excess return for each data observation was stored in a new column/feature called **Long_Excess_Returns_(%)**.

The annualized excess return on a short position of each data observation in the stock price data frame was computed using the following formula:

$$\text{Short Position Annualized Excess Return} = x - y \tag{3.4.6}$$

where x represents the short position annualized HPR on a data observation, and y represents the short position annualized HPR on the SPDR S&P 500 ETF. The short position annualized excess return for each data observation was stored in a new column/feature called **Short_Excess_Returns_(%)**.

3.4.2 Analysis of Returns by Portfolio

The data observations in the stock price data frame were divided into the following portfolios:

- Portfolio of high F-Score data observations (i.e., observations with an F-Score of 8 or 9).
- Portfolio of low F-Score data observations (i.e., observations with a F-Score of 0 or 1).
- Portfolio of data observations with F-Score of x (where $x = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$).
- Portfolio of long positions on high F-Score data observations and short positions on low F-Score data observations.

Just as Piotroski (2000), the mean, median, 10th percentile, 25th percentile, 75th percentile, and 90th percentile were calculated for each portfolio's returns were possible. The *mean()* and *median()* functions of Pandas were used to calculate the mean and median of each portfolio's returns respectively. The *quantile()* function of Pandas was used to generate the 10th percentile, 25th percentile, 75th percentile, and 90th percentile of each portfolio's returns.

3.5 Automation of Piotroski's Methodology with Multi-Layer Perceptron

3.5.1 Dealing with Imbalanced Data Set

To automate Piotroski's methodology using a multi-layer perceptron (MLP) architecture, the **F_SCORE** column/feature in the fundamentals data frame was used as the target variable. The very high B/M data observations in the fundamentals data frame were filtered and over-written on the fundamentals data frame (i.e., the fundamentals data frame was saved with very high B/M data observations only). After filtering these observations, the fundamentals data frame had $1,457$ rows.

An imbalance of **F_SCORE** classes was realized during an analysis of results that had been produced when implementing Piotroski's methodology. Table 8 shows the frequency and proportion of data observations belonging to each **F_SCORE** class. Notable classes with significant imbalances were **F_SCORE** 0, 1, and 9.

F_SCORE	**Frequency**	**Proportion**
0	1	0.00069
1	7	0.00480
2	70	0.04804
3	185	0.12697
4	300	0.20590
5	355	0.24365
6	285	0.19561
7	175	0.12011
8	70	0.04804
9	9	0.00618

Table 8: **F_SCORE** class distribution

A class imbalance in such a complex data set could lead to adverse results (Japkowiz and Stephen, 2002). Therefore, various synthetic data generation techniques were explored such as SMOTE (Chawla et al., 2002) and ADASYN (He et al., 2008). However, due to the **F_SCORE** class of 1 having just 1 data observation, no over-sampling technique could have been used. This is because over-sampling with just 1 data observation 100-or-so times would have made overfitting very likely (Weiss et al., 2007). To address this problem, the Python module *random* was used to generate synthetic values for each of the features in the fundamentals data set. A neighbourhood was defined in which a synthetic value was contained; for example, a synthetic value for the column/feature **Total_Assets** was generated using the formula:

$$\mathbf{Total\_Assets\_Synthetic} = \text{random}(\min(\mathbf{Total\_Assets_{F_0}}) - 5,\ \max(\mathbf{Total\_Assets_{F_0}}) + 5) \quad (3.5.1)$$

where $\mathbf{F}_0$ denotes a data observation from the **F_SCORE** class of 0. A radius value of 5 was used to find similar column/feature values that have characteristics of the **F_SCORE** class equal to 0. Similar formulas to that of (3.5.1) were used to find synthetic values for each numerical non-engineered column/feature[3] for the **F_SCORE** class of 0. A loop was used to generate 354 synthetic data observations for the **F_SCORE** class of 0, thus giving the class 355 observations in total.

A similar method was used to generate synthetic data observations for the other minority **F_SCORE** classes, although the following formula was used to find synthetic values for each numerical non-engineered column/feature in the fundamentals data frame:

$$\textbf{Feature\_Synthetic} = \text{random}(\min(\textbf{Feature}_{\mathbf{F}_i}),\ \max(\textbf{Feature}_{\mathbf{F}_i})) \tag{3.5.2}$$

where **Feature** is the name of the column/feature that the synthetic data is being generated for, and $\mathbf{F}_i$ denoted a from the **F_SCORE** class of i. Once completed for every minority **F_SCORE** class, every **F_SCORE** had a total of 355 data observations each. Therefore, the classes of the data set were balanced, and the model data frame had $3,550$ rows.

3.5.2 Train-Test Set Split and One Hot Encoding of F_SCORE

The 9 fundamental measure columns/features (**ROA**, **CFO**, etc.) and corresponding scoring function columns/features (**F_ROA**, **F_CFO**, etc.) were stored in a new data frame called model data frame for ease of access. Furthermore, the **F_SCORE** column/feature was also stored in the model data frame. The *train_test_split()* function of the Python package Scikit-Learn (Scikit-Learn, 2019) was used to split the data observations in the model data frame; 75% of the data observations were used in the training set and 25% used in the test set.

The **F_SCORE** feature/column in the model data frame was one hot encoded using the *to_categorical()* function of the Python package Keras (Keras, 2019).

3.5.3 Multi-Layer Perceptron Architecture

Multi-layer feedforward networks are universal approximators, meaning that they can approximate any continuous function for inputs in a specific range (Hornik et al., 1989). Due to the discontinuous nature of Piotroski's scoring functions (**F_ROA**, **F_CFO**, etc.), which model the Heaviside step function, an architecture based on a single multi-layer perceptron (MLP) would not have been feasible. Therefore, multiple MLPs with single hidden layers were designed to model the discontinuity of each scoring function with the use of their corresponding fundamental measure columns/features (**ROA**, **CFO**, etc.). As each of the 9 scoring functions assigns a 0 or 1 to their corresponding fundamental measure, each MLP was designed to model a binary classification problem; thus, one output neuron with a sigmoid activation function was used, and the binary cross-entropy loss function (3.5.3) was used to assess each model's loss.

$$\text{Binary Cross-Entropy Loss} = \sum_{i}^{N=2} t_i \log(s_i) \tag{3.5.3}$$

Once predictions had been generated from each scoring function MLP, these were then used as input into a final (**F_SCORE**) MLP to calculate the **F_SCORE** of any given data observation. As

[3] A numerical non-engineered column/feature is a column/feature in the fundamentals data frame that is not a fundamental measure, scoring function or F-Score from Piotroski's methodology.

the **F_SCORE** feature/column of the model data frame contained values ranging from 0 to 9, the **F_SCORE** MLP was designed to model a multi-class classification problem; thus, ten output neurons with softmax activation functions were used, and the categorical cross-entropy loss function (3.5.4) was used to assess the model's loss.

$$\text{Categorical Cross-Entropy Loss} = \sum_{i}^{N=10} t_i \log(s_i) \tag{3.5.4}$$

The overall model architecture is illustrated in figure 6.

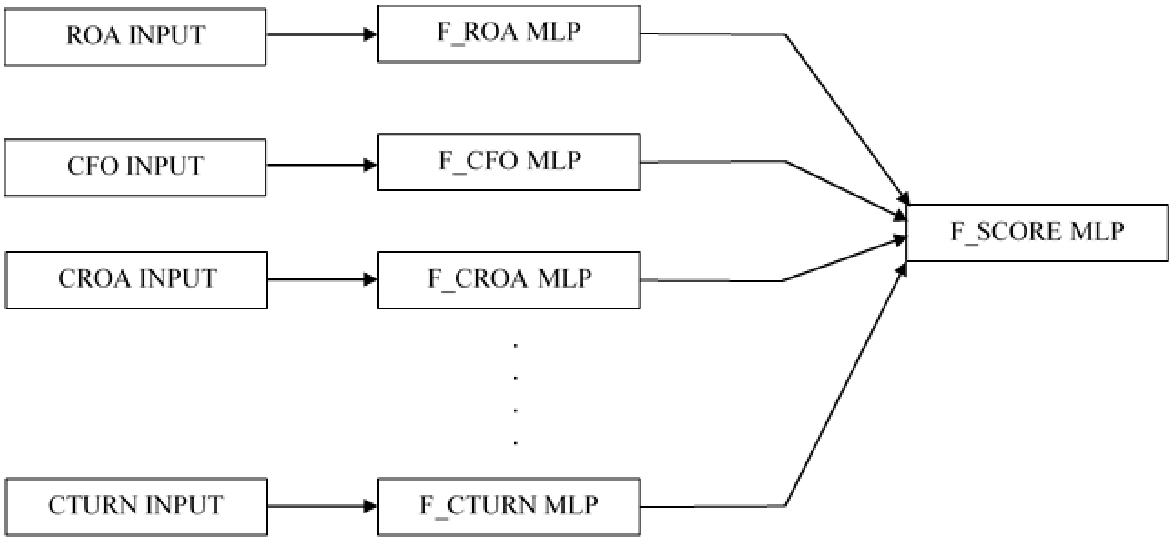

Figure 6: Multi-layer perceptron architecture for Piotroski's methodology

3.5.4 Hyper-parameter Grid Search

A grid search was conducted on each of the MLP models to source an optimal combination of model hyperparameters. For the scoring function MLPs (**F_ROA MLP**, **F_CFO MLP**, etc.), the hyperparameters and corresponding values in table 9 were tested. For the **F_SCORE** MLP, identical hyperparameters were tested as for the scoring function MLPs, although the number of hidden neurons implemented were [9, 18, 27, 36] instead of the values that are shown in table 9.

Hyper-parameter Name	Values
Number of Hidden Neurons	[1, 2, 3, 4]
Optimizer	[sgd, adam, adamax, nadam]
Activation Function of Hidden Neurons	[relu, sigmoid]

Table 9: Hyper-parameters tested for MLP models

Note that during the training of the MLP models, a validation set of 25% was held out for evaluation. Furthermore, early-stopping was employed during training with the validation loss being the metric that was monitored. The model that had the smallest validation loss over 400 epochs (or below) was selected as the most optimal MLP and thus used in the final MLP architecture described in section (3.5.3).

3.6 Annual Filing Management's Discussion and Analysis Data Collection and Manipulation

3.6.1 Retrieving Annual Filing Page URLs

The annual filings for any company listed on a US exchange are available via the SEC EDGAR database (Securities and Exchange Commission, 2019). Therefore, this database was used to scrape annual filings data for the data observations contained in the fundamentals data frame. The management's discussion and analysis (MD&A) narrative of each annual filing were to be extracted, as this is where a particular company's financial results are discussed (Schroeder and Gibson, 1990).

The first step in scraping the MD&A narrative from the required annual fillings was to obtain their corresponding filing page URLs. For ease of access, the data observations were stored in a new data frame referred to as filling page data frame. The **CIK**, **Ticker_Symbol**, **Company_Name**, **Data_Date**, columns/features were copied from the fundamentals data frame, and the new columns/features listed in table 10 were also created.

Feature Name	Feature Description
Filling_Date	The date on which a particular company filed its annual filing with the SEC.
Filing_Type	The type of annual filing; note that companies domiciled in the US are required to file 10-K reports, companies that are domiciled in Canada are required to file 40-F reports, and other companies are required to file 20-F reports.
Filing_Url	The Url of a particular filing page corresponding to a particular company.
MDA_Placement	The placement of the MD&A for a particular filing; note that some companies file their MD&A narrative in separate exhibits.

Table 10: Description of new features created in filing page data frame

The Python packages Requests (Reitz, 2019) and Beautiful Soup (Richardson, 2017) were used to get access to the SEC EDGAR web page. The filing for each data observation in the filing page data frame was searched for on EDGAR via their **CIK** values and a filing type (e.g. 10-K). The results of the search returned a table, such as the one shown in figure 7, of filings for the **CIK** value and filing type value that was requested. A function was built to match the year of a data observation's **Data_Date** value to a row in the returned table with an equal year in the 'Filing Date' column. The value in the 'Filing Date' column of the returned table was stored in the corresponding data observation's **Filing_Date** cell in the filing page data frame. The data observation's filing URL was also obtained from the same row, although from the 'Documents' button in the 'Format' column of the returned table. To search the returned table, the first <table> tag was sourced using Beautiful Soup's *find()* function and the content in each of the table's rows (which are represented by the HTML tag <tr>) were looped over until the 'Filing Date' year matched the year of the corresponding data observation's **Data_Date** value. The value 'N/A' was assigned to data observations' **Filing_Date** cell and/or **Filing_Url** cell for which filing dates and/or filing URLs could not be retrieved.

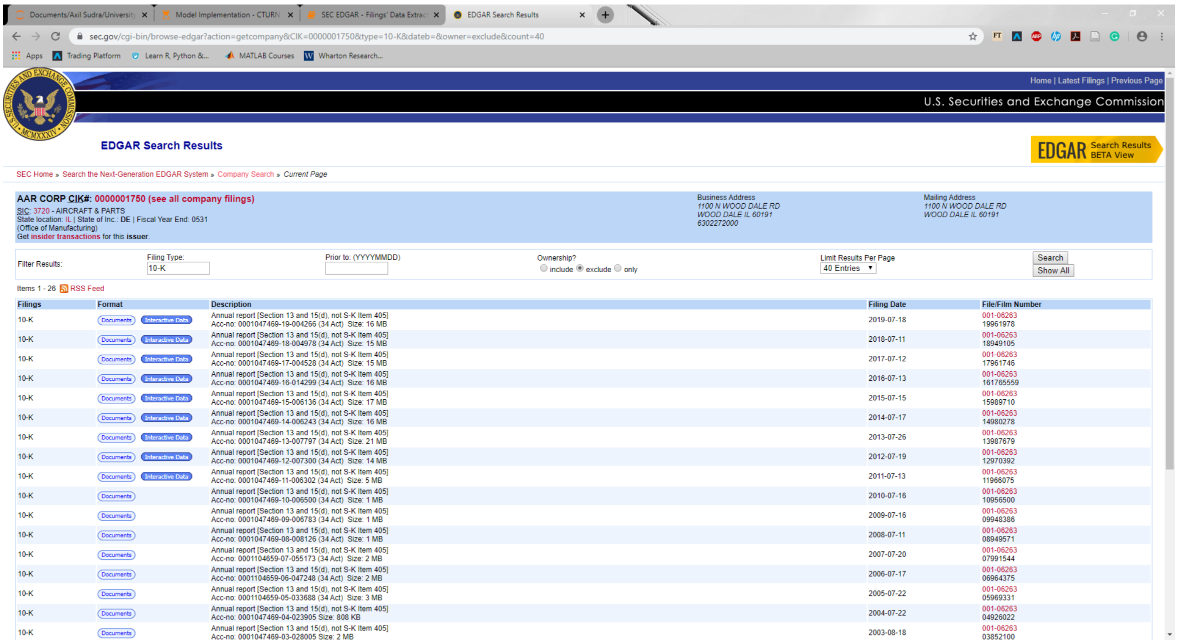

Figure 7: Example of SEC EDGAR filing table for AAR Corp

392 data observations had ‘N/A’ values in their **Filing_Url** cells in the filing page data frame; these were investigated, and manually retrieved where possible.

3.6.2 Checking Annual Filing Page URL Correspond to Correct Filings

Once the annual filing page URL for each data observation had been obtained (apart from those were unable to be sourced), the second step involved a check to see whether the **Data_Date** value matched the ‘period of report’ date on each filing page for a particular data observation. The ‘period of report’ date is the fiscal year-end date for a particular company that the filing corresponds to.

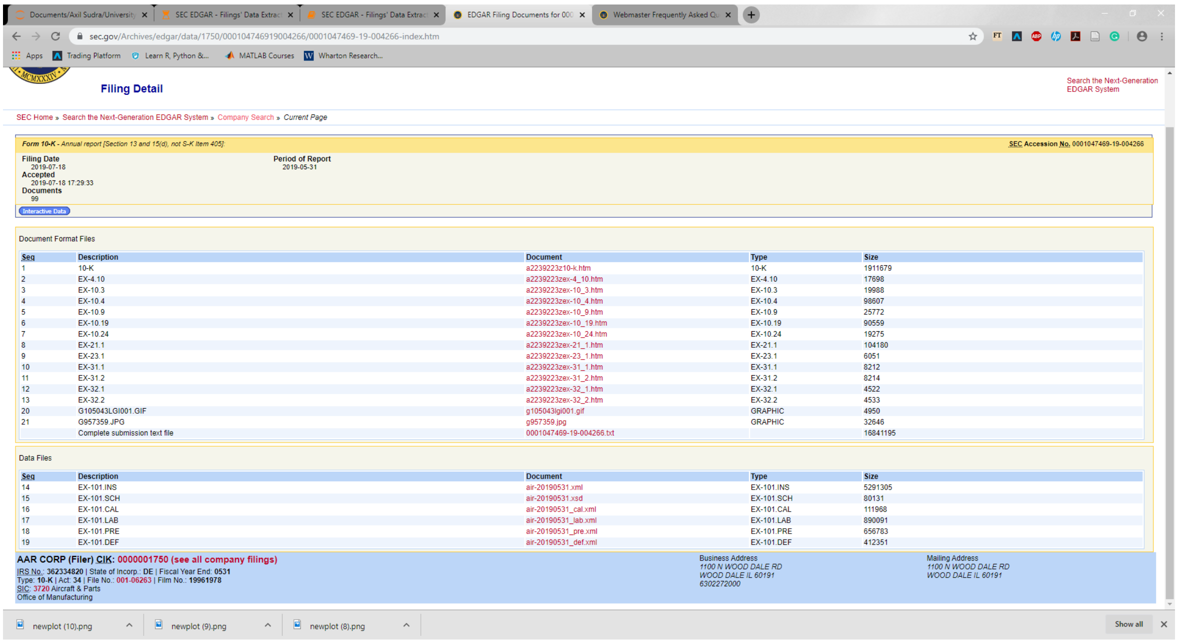

Figure 8: Example of SEC EDGAR filing page for AAR Corp

Figure 8 shows an example of a filing page on the SEC EDGAR website. The 'period of report' date is the value positioned on the top yellow banner just off to the right from the centre of the page. Through analyzing the HTML code of a sample of filing pages corresponding to data observations in the filing page data frame, it was discovered the 'period of report' item was always a child node to the class 'formGrouping'. Therefore, the *find_all()* function of Beautiful Soup was used to retrieve this specific HTML tag, and extract the text. A new column/feature was created in the filing page data frame called **Filing_Check**. If the **Data_Date** of a data observation matched its corresponding filing page 'period of report' date, then 'True' would be stored in the **Filing_Check** column; if the values did not match, then 'False' would be stored in the **Filing_Check** column.

The number of data observations that had 'False' values in the **Filing_Check** column was equal to 104. These data observations were manually investigated to see if the date on the annual filing matched with the **Data_Date** values, and set to 'True' if they did indeed match. All of 104 data observations' **Filing_Check** column values were set to 'True' as they matched the date on their corresponding annual filing reports.

3.6.3 Retrieving Annual Filing Report URLs

The next step in collecting the data that was needed was to obtain the actual annual filing report URL for each data observation in the filing page data frame. Details about any annual filing report are stored as the first item in the main table of any corresponding filing page; the first large table shown in figure 8 is an example of this. Therefore, the *find_all()* function of Beautiful Soup was used to find all the tables on any filing page (which were identified by the HTML `<table>` tag), and extract the URL from the 'Document' column of the first row of the first table. The extracted annual filing URL for each data observation was stored in a new column/feature called **Filing_Report_Url** in the filing page data frame.

3.6.4 Locate Management's Discussion and Analysis Narrative Position in Annual Filing Report

The penultimate step in collecting the MD&A narrative for each data observation in the filing page data frame was to locate the position of the MD&A narrative in a company's annual filing report. This was done by searching for the title of the MD&A section via regular expressions and obtaining the start and end page of that section from the table of contents. Note that the MD&A narrative in 20-F filing reports is denoted as 'Operating and Financial Review of Prospects' (OFP). Also, note that the Python package re was used to compile regular expressions when required.

Using the *find_all()* function of Beautiful Soup once again, all the tables were located in a particular company's annual filing report for a given date. The regular expression below was used to find the row in the table of contents corresponding to the MD&A narrative:

```
(?i)management(.|)s\s*discussion\s*(and|\&)\s*analysis.*
```

Once this item was located, the page number was extracted from the row in the table of contents. The page number for the next item below the MD&A narrative was also located in the table of contents and was also extracted. Note that in a 10-K filing the next item to the MD&A narrative is 'Quantitative and Qualitative Disclosures of Market Risk' (QQDMR). The following regular expression was used to search for the latter:

```
(?i)quantitative\s*(and|\&)\s*qualitative\s*disclosures.*
```

Note that data observations with **Filing_Type** values of 40-F have the same naming convention as that of 10-K filings, and thus used the same regular expressions like the ones above.

For data observations that had **Filing_Type** values of 20-F, the regular expression below was used to source the starting page number of the MD&A (OFP) narrative.

```
(?i)operating\s*(and|\&)\s*financial\s*review.*
```

As with the data observations with **Filing_Type** values of 10-K, the page number for the next item below the OFP narrative in the 20-F reports was also located in the table of contents and was also extracted. The next item to the OFP narrative in a 20-F filing is 'Directors, Senior Management and Employees' (DSME). The following regular expression was used to search for the latter:

```
(?i)directors(\,|.|)\s*senior\s*management\s*(and|\&)\s*emp.*
```

The start and end page numbers were stored in the filing page data frame under the new columns **MDA_Start_Position** and **MDA_End_Position** or **OFP_Start_Position** and **OFP_End_Position** depending on the **Filing_Type** value of the data observation.

It was observed that the majority of annual filing reports contained thematic break tags between pages, and thus the closest tag above these thematic break tags were searched for the start and end page numbers stored in the filing page data frame. The tag containing the start page number of the MD&A narrative was stored in a new column/feature called **MDA_Page_Number_Start_Tag** in the filing page data frame; similarly, the tag containing the start page number of the QQDMR section was stored in a new column/feature called **MDA_Page_Number_End_Tag**. For data observations with **Filing_Type** values of 20-F, the new columns/features **OFP_Page_Number_Start_Tag** and **OFP_Page_Number_End_Tag** were created for the same purpose. If a tag could not be found for the start or end page number, the column/feature value would be 'N/A'.

3.6.5 Extracting the Management's Discussion and Analysis Narrative

The last step was to extract the MD&A narrative (or OFP narrative) for each data observation and store in a text file format. This was completed by finding the position of the tags that had been stored in the columns/features **MDA_Page_Number_Start_Tag** (or **OFP_Page_Number_Start_Tag**) and **MDA_Page_Number_End_Tag** (or **OFP_Page_Number_End_Tag**) in a string of each annual filing's HTML code and extracting through indexing. Once the MD&A (or OFP) HTML code had been extracted, the *get_text()* function of Beautiful Soup was used to remove all the HTML tags. Furthermore, the Python package unicodedata (Kuchling, 2019) was used to get the normal form of Unicode strings in the annual filings' text. The text for the MD&A (or OFP) narrative for each data observation in the filing page data frame was then saved in a text file, and the corresponding file name was stored in a column/feature called **File_Name**.

3.7 Labeling Sentiment of Management's Discussion and Analysis Narrative

3.7.1 Storing Meaningful Columns/Features

Meaningful columns/features were extracted from the filing page data frame stored in a new Pandas data frame structure. This new data frame was referred to as the master filing data frame, and consisted of the following columns/features: **CIK**, **Ticker_Symbol**, **Company_Name**, **Data_Date**, **Filing_Date**, **Filing_Type**, **MDA_Placement**, and File_Name.

3.7.2 Labeling Sentiment with Loughran and Mcdonald Financial Dictionary

The Loughran and Mcdonald (2011) financial dictionary was used to label each of the data observations' MD&A (or OFP) narratives with a polarity sentiment. This dictionary consisted of the following 7 lists of words: positive, negative, uncertainty, litigious, strong-modal, weak-modal, and constraining. To label the text documents corresponding to each data observation's MD&A (or OFP) narrative, the positive and negative word lists were used. Note that positive list contained 354 words, and the negative list contained $2,355$ words.

Two dictionaries were defined to keep a count on the number of positive and negative words in each data observation's MD&A (or OFP) narrative. Two new columns/features were created in the master filing data frame to store the number of positive and negative words the data observations' text files contained. These columns/features were called **No_Of_Positive_Words** and **No_Of_Negative_Words** respectively. After obtaining the count of positive and negative words, a sentiment score was calculated for each of the data observations' text files using the following formula:

$$\text{Sentiment Score} = \frac{\textbf{No\_Of\_Positive\_Words} - \textbf{No\_Of\_Negative\_Words}}{\textbf{No\_Of\_Positive\_Words} + \textbf{No\_Of\_Negative\_Words}} \tag{3.7.1}$$

If the sentiment score was greater than 0, the sentiment of a data observation's MD&A (or OFP) narrative was classed as positive. On the other, if the sentiment score was less than or equal to 0, the sentiment of a data observation's MD&A (or OFP) was classed as negative.

Lastly, a feasibility analysis was conducted to see whether labelling with the use of Loughran and Mcdonald's financial dictionary could differentiate good and bad stocks that had high (8 or 9) and low (0 or 1) **F_SCORE** values. The same return metrics used to evaluate Piotroski's methodology in section 3.4.1 were used to gauge success; the final decision to use the Loughran and Mcdonald dictionary was to be based on excess returns.

3.7.3 Labeling Sentiment with Long Position Annualized Holding Period Returns

In case the Loughran and Mcdonald labelling method described in section 3.7.2 was not feasible (i.e., the returns of selected portfolios were poor), the sentiment of each data observation's MD&A (or OFP) narrative text was labeled using its corresponding long position annualized HPR contained in the **Long_Annualized_Holding_Period_Return_(%)** column of the stock price data frame. If a data observation's **Long_Annualized_Holding_Period_Return_(%)** value was greater than 0, its sentiment would be classed as positive; otherwise, it would be classed as negative.

3.8 Automation of MD&A Narrative Sentiment Analysis

3.8.1 Textual Preprocessing

For an LSTM recurrent neural network (RNN) to be trained on any textual data, there are various steps that have to be taken to convert the data into a 'readable' format for the model. These steps include tokenization of text, removal of stop-words and punctuation, converting tokens to lowercase, and removing blank space tokens.

The natural language tool kit (nltk) package (Bird, 2017) in Python was used to tokenize each data observation's MD&A (or OFP) narrative text, and its function *stopwords* from the corpus module was used to remove any stop-words in any text. The string module was used to remove common punctuation characters, and a regular expression statement was used to remove any non-word patterns. Note that any words containing digits were removed with the of a regular expression statement also. Blank space tokens were filtered via Python's *filter* functionality. A fundamental step to covert the token lists of each data observation to sequences of numerical vectors was completed via the Keras *text_to_sequences()* function. Lastly, the numerical vector of each data observation was padded to equal lengths using the *pad_sequences()* function in Keras. Note that due to running the LSTM RNN architecture on a standard CPU, and given that the average length of each MD&A narrative text file was between $64,000$ and $74,000$ characters, only the first 500 elements were taken from each vector.

3.8.2 Train-Test Split and Encoding Sentiment Labels as Binary Outputs

The **Sentiment_Class** column/feature was converted from text to binary output values; 'positive' was encoded with a 1 and 'negative' was encoded with a 0. As with train and test division in section 3.5.2, 75% of the data observations were used for training and the remaining 25% for testing.

3.8.3 LSTM Recurrent Neural Network Architecture

An RNN was implemented with long short-term memory (LSTM) units in hidden layers using the Keras package once again. The network consisted of a word embedding layer to process the numerical vectors representing the MD&A text for each data observation, two LSTM hidden layers, and an output layer consisting of a single neuron with a sigmoid activation function to predict the probabilities of the binary classes. The first LSTM hidden layer was designed to return sequences for any given input and pass to the second LSTM hidden layer. This is because the network is dealing with sequence data (i.e., text). The binary cross-entropy loss function was used to assess the model's loss; this is given in equation (3.5.3)

3.8.4 Hyper-parameter Grid Search

Just as in section 3.5.4 for the MLP architecture for predicting Piotroski's F-Score, a hyper-parameter grid search was used to source the optimal set of hyperparameters for the RNN model. The hyper-parameters that were tested and their corresponding are shown in table 11.

Hyper-parameter Name	Values
Number of Hidden Neurons	[18, 27]
Optimizer	[sgd, adam, adamax, nadam]

Table 11: Hyper-parameters tested for LSTM RNN model

4 Results

4.1 Implementation and Back-testing of Piotroski's Methodology

4.1.1 Descriptive Statistics and Visualizations

The product of implementing Piotroski's methodology on the sample of very high B/M companies was a vast range of descriptive statistics and visualization plots. The most significant results are discussed in this section. Note that the results presented in this section are based on a sample size of $1,457$ data observations between the dates January 1, 2011, and September 2, 2019.

Table 12 shows the mean, median, and standard deviation of the columns/features in the fundamentals data frame the represent Piotroski's 9 fundamental measures (as described in section 3.2.5). The table also shows the proportion of each fundamental measure that has been scored with a positive signal (i.e. 1) from its corresponding scoring function (as described in section 3.2.6). Consistent with Piotroski (2000) and Fama and French (1995), the sample of very high B/M companies displayed poor performance. The average and median **ROA** for these companies were -0.0069 and 0.0081 respectively. Companies in the sample had seen declines in **ROA** over the specified time period, with the mean and median **CROA** being -0.0167 and -0.0071 respectively. From the mean of **CLEVER** and **CLIQUID**, it was evident that the average high B/M company has high long-term debt levels and consistent problems with liquidity. The average high B/M company also struggled to raise capital internally, as the mean **EQ_OFFER** was greater than 0.

The majority of Piotroski's nine fundamental measures contained data observations with less than half qualifying for a positive signal. The **CFO** and **ACCRUAL** columns/features were considered as significant outliers with respect to the proportion of data observations that had been scored with a positive signal. The **EQ_OFFER** feature/column had the smallest proportion of data observations that had been scored with a positive signal, which was equal to 0.3240.

Feature Name	Mean	Median	Standard Deviation	Proportion with Positive Signal
ROA	−0.0069	0.0081	0.0910	0.5772
CFO	0.0673	0.0649	0.0577	0.9128
CROA	−0.0167	−0.0071	0.1041	0.4008
ACCRUAL	−0.0742	−0.0599	0.0873	0.9108
CLEVER	0.0051	0.0000	0.0994	0.4798
CLIQUID	−0.0601	−0.0163	1.6726	0.4791
EQ_OFFER	6.8627	0.3000	77.8609	0.3240
CMARGIN	−0.0153	−0.0046	0.4103	0.4221
CTURN	−0.0203	−0.0043	0.3368	0.4715

Table 12: Mean, Median, Standard Deviation, and Proportion with Positive Signal of Piotroski's Nine Fundamental Measures (where $n = 1,457$) (to 2.d.p.)

The histogram in figure 9 illustrates the distribution of the **F_SCORE** column/feature of very high

B/M companies. The majority of data observations had an **F_SCORE** of 5; only 1 data observation had an **F_SCORE** of 0. The methodology used to address the imbalance in the **F_SCORE** column/feature is discussed in section (3.5.1).

The box-plots in figure 10 illustrate the normalized **ROA** for very high B/M companies divided by **Size_Tercile**. Note that **ROA** was normalized using the following formula:

$$\text{Normalized ROA}_t = \frac{\mathbf{ROA}_t - \min(\mathbf{ROA})}{\max(\mathbf{ROA}) - \min(\mathbf{ROA})} \tag{4.1.1}$$

where t represents the a specific the data observation. From the box-plots, it was observed that the **Size_Tercile** feature had a non-perfect positive linear relationship with normalized **ROA** (i.e., the larger the company, the larger the value of normalized **ROA**). Very High B/M companies in each of the **Size_Tercile** categories roughly had the same amount of normalized **ROA** outliers; however, normalized **ROA** outliers in the large **Size_Tercile** category had a wider spread.

The box-plots in figure 11 depicts normalized **CFO** for very high B/M companies divided by **Size_Tercile**. A similar formula to (4.1.1) was used to normalize **CFO** values, although **ROA** was replaced by **CFO**. Very High B/M companies in the large **Size_Tercile** category generated the highest amount of cash flow from operating activities, which was expected as large companies have larger operation resources and thus can produce more goods and/or services. Significantly, the mean normalized **CFO** of very high B/M companies in the small **Size_Tercile** category was higher than of the very high B/M companies in the medium **Size_Tercile**.

The box-plots in figure 12 show normalized **CLEVER** for very high B/M companies divided by **Size_Tercile**. As with normalized **CFO**, a similar formula to (4.1.1) was used to normalize **CLEVER** values, although **ROA** was replaced by **CLEVER**. Very High B/M companies in the small **Size_Tercile** category had the highest average normalized **CLEVER** value. This was be expected as small companies that do not have sufficient internal funds cannot raise capital frequently through equity, and thus have to borrow from financial institutions or issue corporate bonds. Very High B/M companies in the medium **Size_Tercile** category had the lowest average normalized **CLEVER** value. The observations in the large **Size_Tercile** category had the largest spread.

Scatter-plots were created to illustrate the normalized **EQ_OFFER** of very high B/M companies by **Market_Capitalization**; these are shown in figure 13. As with normalized **CFO** and normalized **CLEVER**, a similar formula (4.1.1) was used to normalize **EQ_OFFER** values, although **ROA** was replaced by **EQ_OFFER**. Significantly, very high B/M companies in the small and medium **Size_Tercile** categories issued the most equity; this evidence goes against the explanation that small companies opt for long-term debt over equity due to not being able to raise capital frequently through equity financing. The normalized **EQ_OFFER** values for very high B/M companies in the large **Size_Tercile** category were also unexpected; these companies raised the least funds through equity financing.

A Pearson correlation heatmap in figure 14 depicts the correlation between each of the nine fundamental measures. There were no significant relationships between any two fundamental measures, apart from **ROA** and **ACCRUAL**.

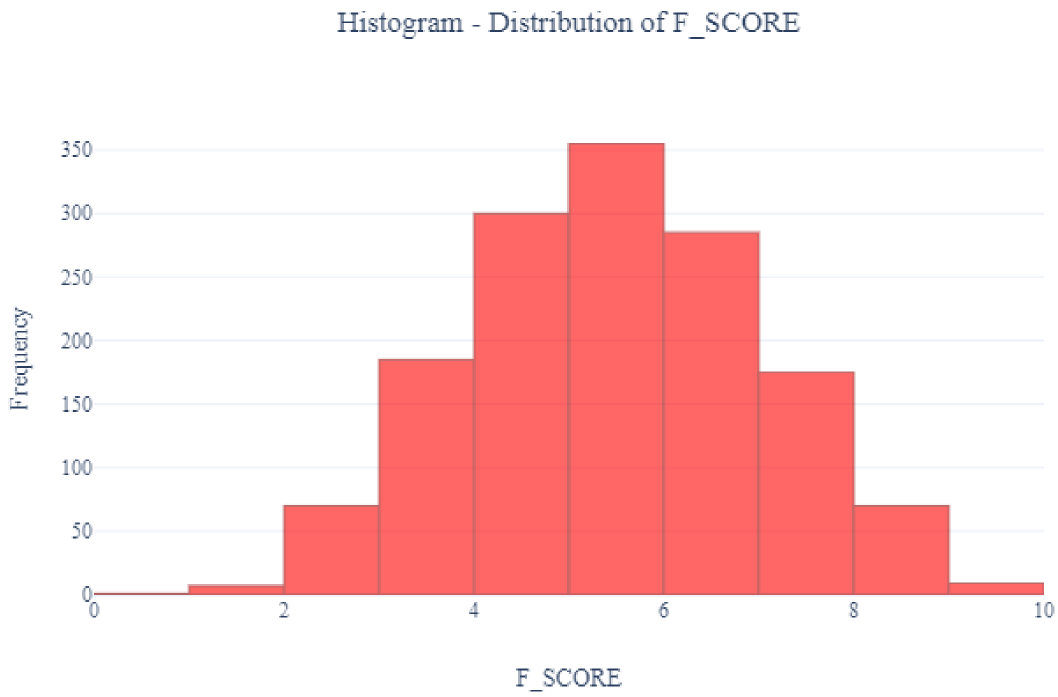

Figure 9: Histogram - distribution of **F_SCORE**

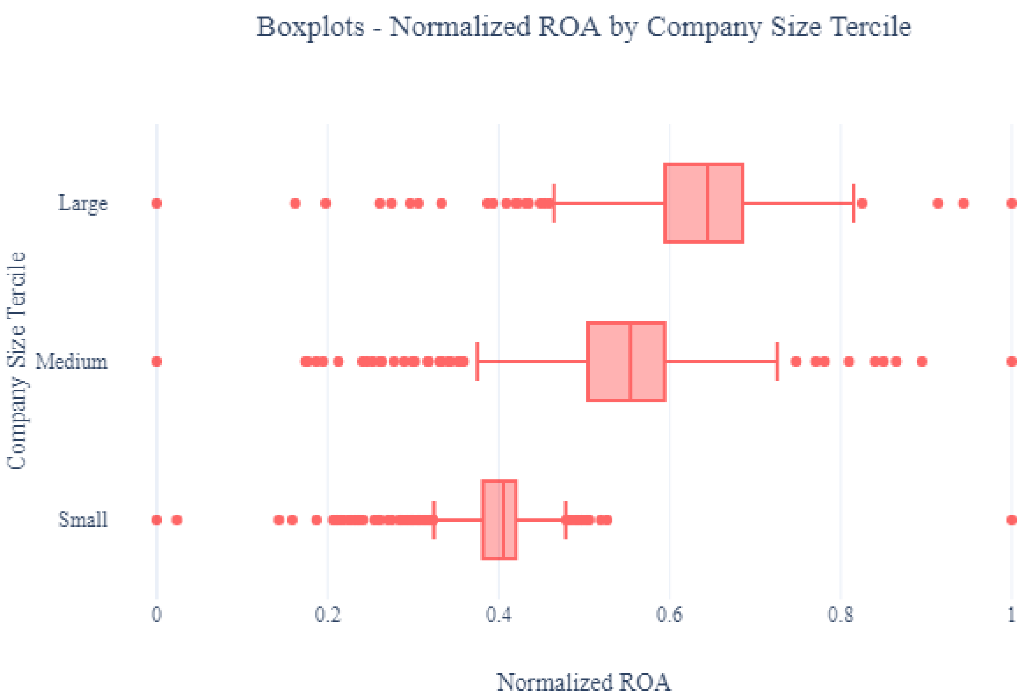

Figure 10: Box-plots - normalized **ROA** by company size tercile

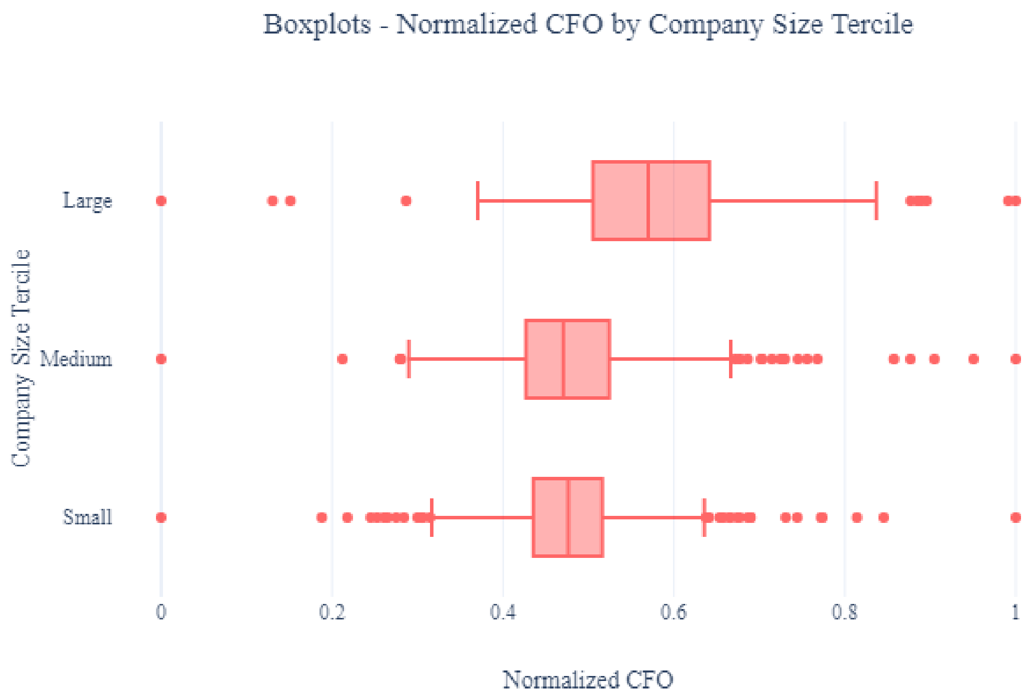

Figure 11: Box-plots - normalized **CFO** by company size tercile

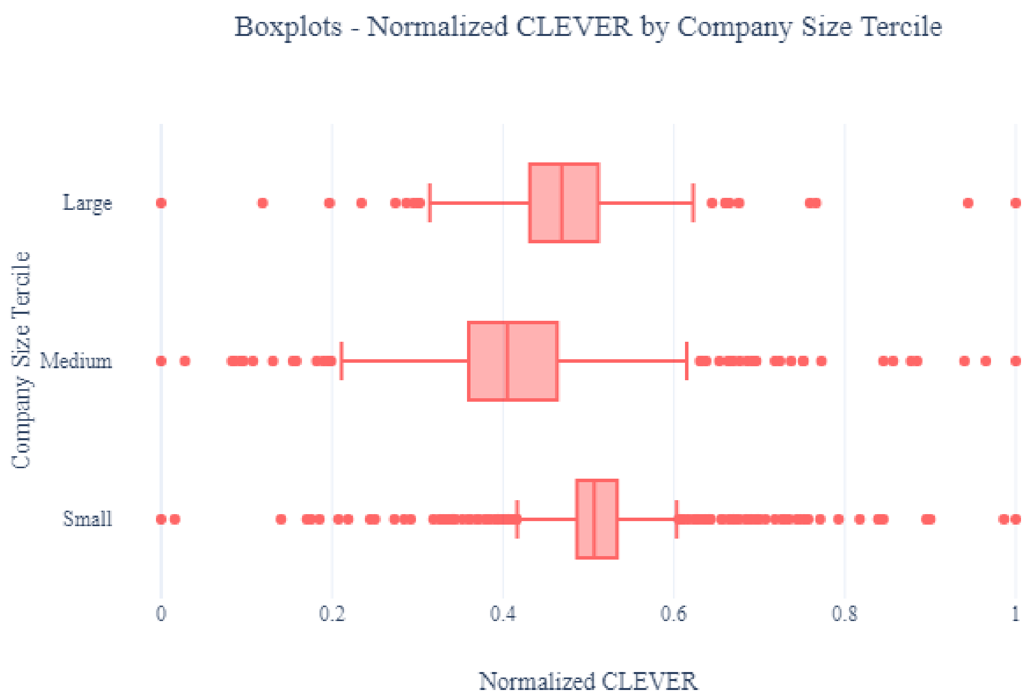

Figure 12: Box-plots - normalized **CLEVER** by company size tercile

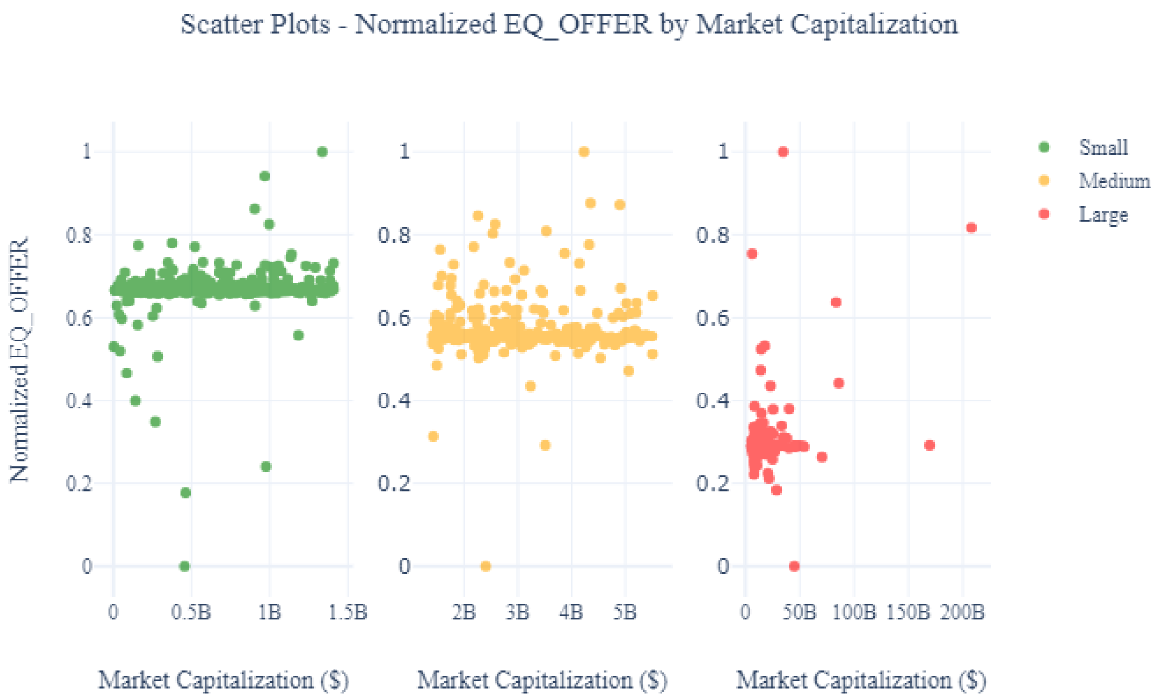

Figure 13: Scatter-plots - normalized **EQ_OFFER** by market capitalization

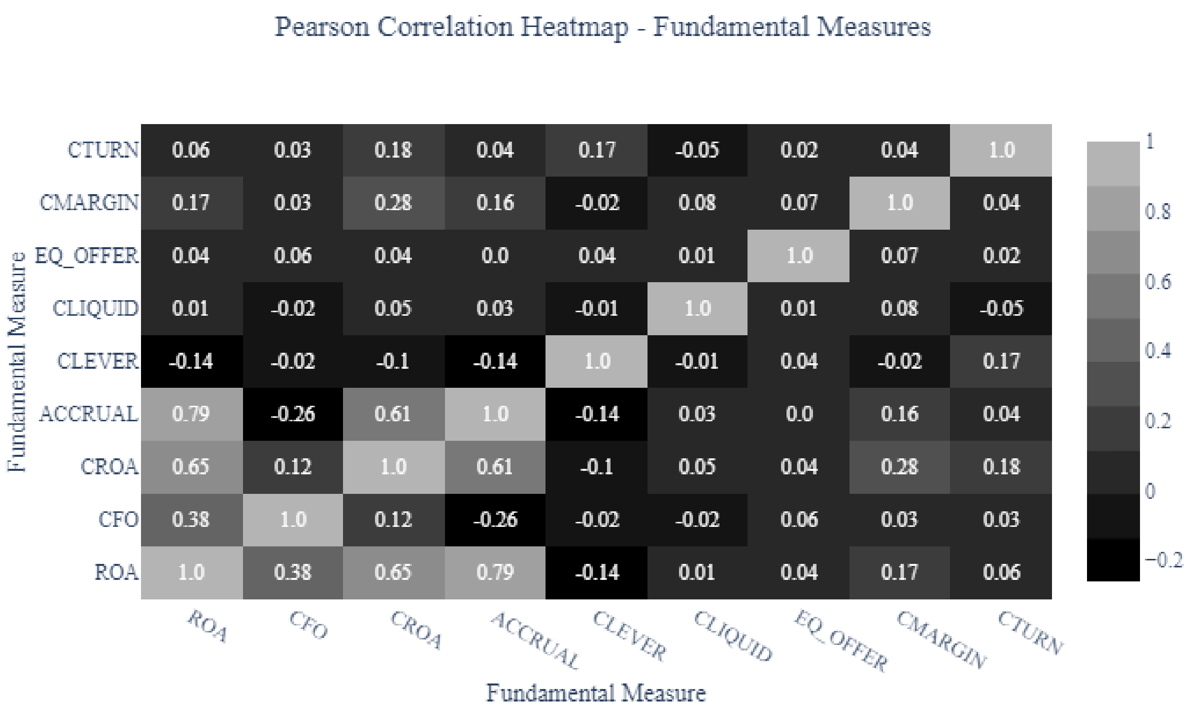

Figure 14: Heatmap - Pearson correlation coefficient between fundamental measures

4.1.2 Analysis of Back-testing Results

The raw mean return of a long-only portfolio of all high B/M companies (which amounted to $1,453$ observations) for the period between January 1, 2011, and September 2, 2019, were 23.4% (to 2.d.p). The annualized mean return for the same portfolio was 101395.70%; the stock LCI (Lannett Co Inc) was found to have an abnormal long-position annualized return, and hence why the annualized mean return for the portfolio was so significant. This outlier was removed from the sample of high B/M companies, and the portfolio's annualized return was evaluated once again. As a result, the annualized mean return for the portfolio was 83.32%. The main return metric of interest was the annualized mean excess return. The long-only portfolio of all high B/M companies achieved an annualized mean excess return of 101385.30% with the outlier in the sample, and 72.80% without the outlier in the sample.

The next two portfolios that were analyzed were Piotroski's long-only high and low **F_SCORE** portfolios. The high **F_SCORE** portfolio achieved a raw mean return of -1.50%, an annualized mean return of -3.70%, and an annualized mean excess return of -13.40%. These returns were unexpected for the long-only high **F_SCORE** portfolio as this portfolio contains the more fundamentally strong high B/M observations. In comparison, the long-only low **F_SCORE** portfolio achieved a raw mean return of -6.40%, an annualized mean return of -15.1%, and an annualized mean excess return of -23.7%. These results were expected of the long-only low **F_SCORE** portfolio.

A short-only portfolio constructed of the low **F_SCORE** observations was also back-tested. This portfolio attained a mean raw return of 65.2%, an annualized mean return of 95.10%, and an annualized mean excess return of 102.80%. Given these substantial returns, this portfolio was one of the highest performing portfolios constructed.

The final portfolio that was constructed by taking long positions on the high B/M companies with high **F_SCORE** values (8 or 9), and taking short positions on the high B/M companies with low **F_SCORE** values. This portfolio achieved a raw mean return of 4.60%, an annualized mean return of 5.40%, and an annualized mean excess return of -2.70%.

The annualized mean excess returns of portfolios constructed from individual **F_SCORE** values are given in Appendix B.

4.2 Automation of Piotroski's Methodology with Multi-Layer Perceptron

4.2.1 MLP Model Training Times and Hyper-parameter Grid Search Results

The training times with grid search for each MLP model are shown in table 13. The slow training times can be explained by the processor of the hardware that the models were executed on; an i5-6400 CPU was used with 4 cores. Unexpectedly, the **F_SCORE** MLP model had taken less time to train than a majority of the scoring function MLP models (e.g. **F_ROA**). The **F_ACCRUAL** MLP was the quickest MLP model to train, which was also unexpected.

The grid search results for each MLP model are available in Appendix C. The most optimal **F_ROA** model had a hidden layer containing 2 neurons with relu activation functions; the best training optimizer for this MLP model was the adam optimizer (Kingma and Ba, 2015). Furthermore, this MLP model achieved a test accuracy of 0.99887 and test loss of 0.01471. The most optimal **F_CFO** model

MLP Name	Training Times
F_ROA	22 minutes and 41 seconds
F_CFO	19 minutes and 40 seconds
F_CROA	19 minutes and 2 seconds
F_ACCRUAL	9 minutes and 14 seconds
F_CLEVER	13 minutes and 56 seconds
F_CLIQUID	22 minutes and 29 seconds
F_EQ_OFFER	17 minutes and 33 seconds
F_CMARGIN	19 minutes and 17 seconds
F_CTURN	22 minutes and 20 seconds
F_SCORE	16 minutes and 1 second

Table 13: Training times for MLP models

had a hidden layer consisting of 4 hidden neurons with relu activation functions; the best training optimizer for this MLP model was the nadam optimizer Dozat (2016). The **F_CFO** MLP achieved a test accuracy of 1.0 and a test loss of 0.00653. The most optimal **F_CROA** MLP model had a hidden layer with 4 neurons with relu activation functions also. The best training optimizer for this MLP model was the adamax optimizer (Kingma and Ba, 2015); it achieved a test accuracy of 0.98986 and a test of 0.0476. The **F_ACCRUAL** MLP model had an architecture consisting of 4 neurons in its hidden layer which all had sigmoid activation functions. The most optimal training optimizer for this MLP model was the adam optimizer; it achieved a test accuracy and loss of 0.99662 and 0.0337 respectively.

The **F_CLEVER** MLP model had a hidden layer consisting of 2 neurons which had sigmoid activation functions. The most optimal training optimizer for this MLP was nadam; it achieved a test accuracy of 0.98761 and a test loss of 0.07555. Similarly, the **F_CLIQUID** MLP model also had 2 neurons in its hidden layer, although they had relu activation functions. The adam optimizer was the best training optimizer for the **F_CLIQUID** MLP model, which achieved a test accuracy and loss of 1.0 and 0.00671 respectively. The best training optimizer for the **F_EQ_OFFER** MLP model was adam also; this model was constructed of 3 hidden neurons which had sigmoid activation functions. The model's test accuracy and loss were 0.9955 and 0.02045 respectively.

Both **F_CMARGIN** and **F_CTURN** MLP models had 4 hidden neurons in their hidden layers; these had relu activation functions. The test accuracy and loss for the **F_CMARGIN** MLP were 0.99662 and 0.02912 respectively. The **F_CTURN** MLP had a test accuracy and loss of 0.99662 and 0.02025 respectively.

The best **F_SCORE** MLP model was constructed of 27 hidden neurons in its one hidden layer; these had sigmoid activation functions and were trained using the adam optimizer. This model had a best test accuracy of 1.0 and best test loss of 0.01178.

4.2.2 Evaluation of MLP Architecture

An independent test set of 888 data observations was randomly selected from the model data frame (which was mentioned in section 3.5.2) to evaluate the performance of the architecture described in section 3.5.3. Note that the separate scoring function MLP models (**F_ROA** MLP, **F_CFO** MLP, etc.) and the **F_SCORE** MLP were saved as h5 files and loaded into one Python environment. The classification result of the MLP architecture is shown in the confusion matrix in figure 9. The overall MLP architecture classified 850 of the data observations correctly, thus giving the model a test accuracy of 0.95721 (to 5.d.p.). Table 14 shows the precision, recall, and f1-score for the individual **F_SCORE** classes. The majority of precision and recall scores are above 0.90, which demonstrates that the MLP model predicts the correct **F_SCORE** classes frequently.

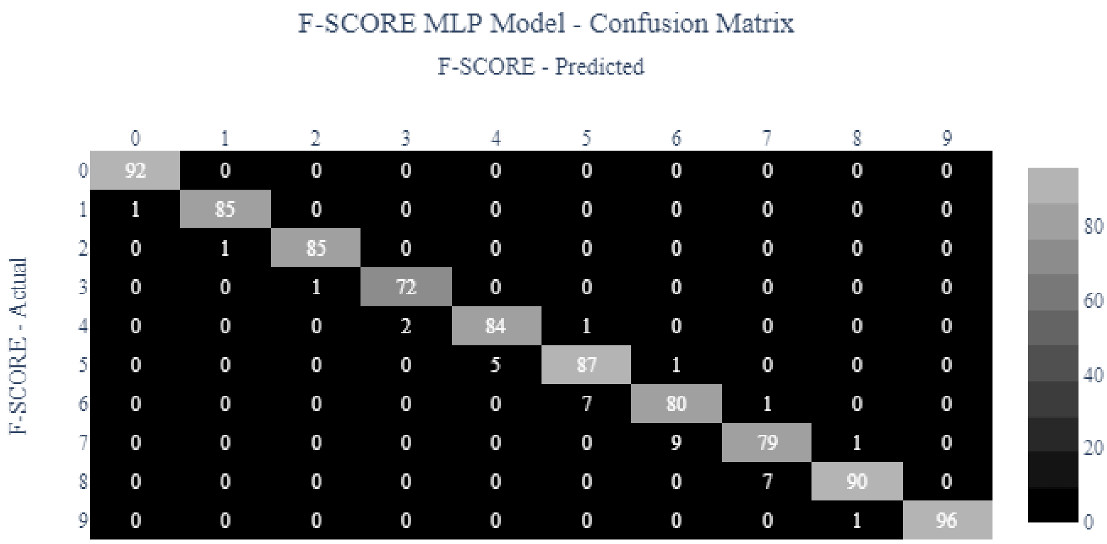

Figure 15: Confusion matrix of MLP architecture

F_SCORE	Precision	Recall	F1-Score
0	0.98925	1.00000	0.99459
1	0.98837	0.98837	0.98837
2	0.98837	0.98837	0.98837
3	0.97297	0.98630	0.97959
4	0.94382	0.96552	0.95455
5	0.91579	0.93548	0.92553
6	0.88889	0.90909	0.89888
7	0.90805	0.88764	0.89773
8	0.97826	0.92784	0.95238
9	1.00000	0.98969	0.99482

Table 14: Precision, recall, and f1-Score of **F_SCORE** classes

4.3 Feasibility Study Results of Sentiment Labeling Methods

4.3.1 Loughran and Mcdonald Financial Dictionary - Back-testing Results

The annualized excess returns of long-only and short-only portfolios based on data observations with high (8 or 9) and low (0 or 1) **F_SCORE** values respectively are given in Appendix D. The purpose of conducting a feasibility study on the labelling of MD&A (or OFP) narrative text was to assess whether the polarity associated with the positive and negative word lists in the Loughran and Mcdonald financial dictionary could differentiate good and bad performance stocks. Furthermore, the dictionary was used to assess whether good and bad data observations with high and low **F_SCORE** values could be separated.

A long-only portfolio consisting of data observations with a positive sentiment achieved an annualized mean excess return of -14.0%, which was a poor result. Unexpectedly, a long-only portfolio consisting of data observations with a negative sentiment achieved an annualized mean excess return of 180261.90%, although there was an outlier present in the sample. Removing the outlier from the sample of negative sentiment data observations gave an annualized mean excess return of 82.0%, which was still significant.

A long-only portfolio created from the high **F_SCORE** valued data observations with a positive sentiment yielded an annualized mean excess return of 10.5%, which was a satisfying result. However, this portfolio only consisted of 2 data observations, and thus the result could not have been taken as evidence that the labelling technique was feasible. Therefore, the data observations with an **F_SCORE** of 7 were also factored into the portfolio. This increased the data sample to 6; the portfolio with the high **F_SCORE** valued data observations and the data observations with **F_SCORE** values of 7 achieved an annualized mean return of -13.5%. From these results so far, it was clear that sentiment labelling using Loughran and Mcdonald's financial dictionary was not feasible. On the other hand, short-only portfolios constructed with data observations with negative sentiments achieved an annualized mean excess return of 100.4%. However, due to the back-testing results on the long-only portfolios, this labelling technique was considered infeasible.

4.3.2 Long Position Annualized Holding Period Returns-Based Sentiment Labeling - Backtesting Results

As the Loughran and Mcdonald method of sentiment labelling was considered infeasible, the sentiment of each data observation was labelled based on its long position annualized HPR (as described in section 3.7.3). The back-testing results for the positive and negative sentiment portfolios are given in Appendix E. A long-only portfolio consisting of high **F_SCORE** data observations with positive MD&A (or OFP) narratives achieved an annualized mean excess return of 17.9%. Similarly, a short-only portfolio consisting of data observations with low **F_SCORE** values with negative sentiments achieve an annualized mean excess return of 87.8%. Furthermore, a portfolio based on a combination of the two strategies achieves a significant annualized mean excess return of 206.3%. This was expected as the labelling technique is based on differentiating strong and weak stocks using their long-position annualized mean returns.

4.4 Automation of MD&A Narrative Sentiment Analysis

4.4.1 Descriptive Statistics and Visualizations

During the textual preprocessing stage of implementing the LSTM model for sentiment analysis, various descriptive statistics and visualizations were produced. The most significant and important results in understanding the data are discussed in this section. Note that 855 MD&A (or OFP) textual narratives were successfully retrieved from a sample size of $1,479$ data observations (as shown in the results for section 4.3.1). Of these 855 data observations, 427 were labelled with a positive sentiment, and 428 were labelled with a negative sentiment; this is illustrated in figure 16. As the distribution of the sentiment classes was near-almost even, no sampling techniques had to used to address the issue of class imbalances.

Of the 855 data observations, 68.2% (to 2.d.p.) were the MD&A narratives from 10-K filings, 18.0% were the OFP narratives of 20-F filings, and the remaining 13.8% were MD&A narratives from 40-F filings. The average length of each text file was between $64,000$ and $74,000$ characters; note this was the average length of the text before any preprocessing was completed. Figure 17 depicts the distribution of the length of the text files. It can be observed that the distribution of the length of the text files was skewed to the left. The text file with the greatest length consisted of $374,000+$ characters.

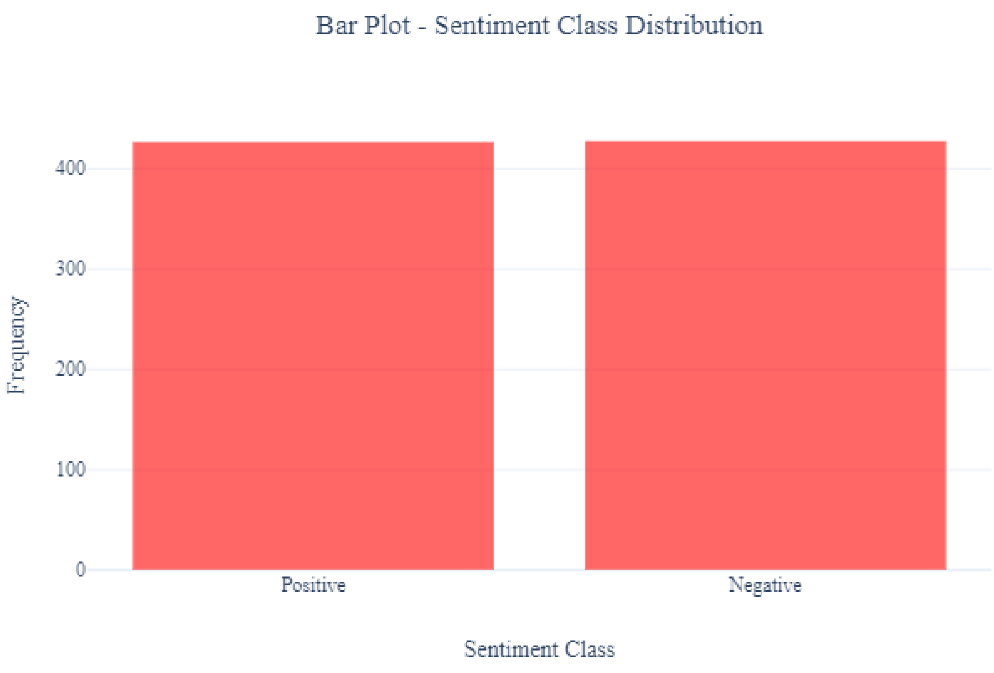

Figure 16: Bar plot - distribution of **Sentiment_Class**

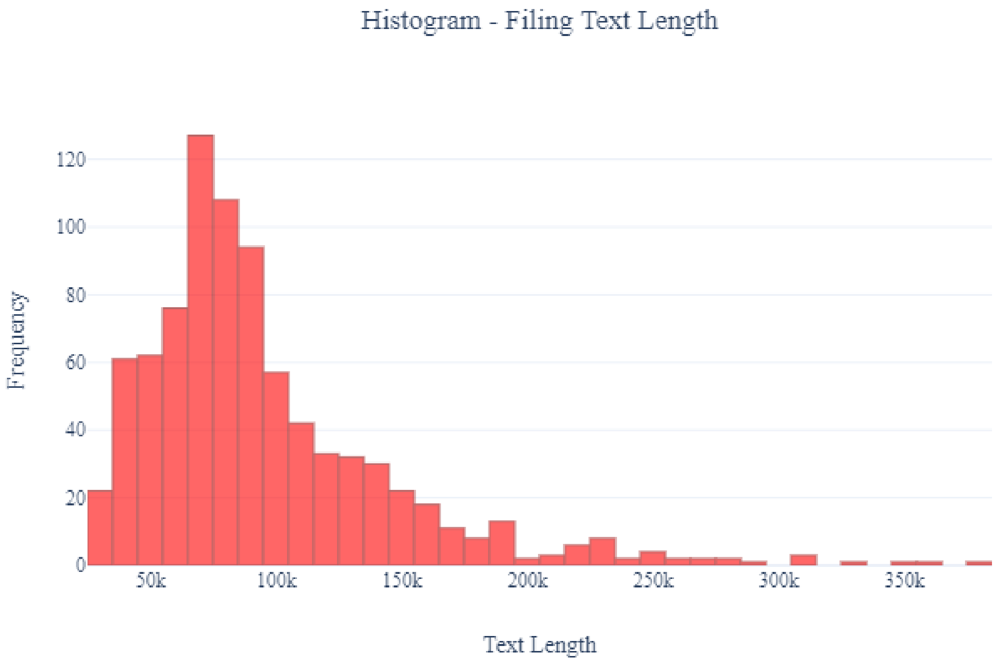

Figure 17: Histogram - distribution of text file length

4.4.2 LSTM Recurrent Neural Network Model Training Time and Hyper-parameter Grid Search Results

The training time with grid search for the LSTM model was 11 minutes and 2 seconds; as each preprocessed MD&A (or OFP) narrative vector was truncated to a length of 500, the training time for the model was quite quick, although at the expense of discarding information that could have improved the network's performance. However, as mentioned in section 3.8.1, training was conducted on a standard CPU with limited processing power so training on the full-length MD&A (or OFP) narrative vectors would have been infeasible.

The grid search results for the LSTM network are available in Appendix F. The best LSTM model was formed on 27 hidden neurons in each of the 2 hidden layers, and was trained using the stochastic gradient descent optimizer in Keras. The training, validation, and testing accuracy (and loss) for this optimal model very disappointing; the model achieved a training, validation, and testing accuracy of 0.52083, 0.49068, and 0.46262 respectively. The model's training, validation, and testing loss was 0.69259, 0.69312, and 0.69506 respectively.

4.4.3 Evaluation of LSTM Architecture

The test set used created with the training set was used to evaluate the performance of the LSTM model architecture described in section 3.8.3. Just as the MLP models, the optimal LSTM was saved as an h5 file and loaded back into a Python environment when the evaluation was completed. The classification result of the LSTM architecture is shown in the confusion matrix in figure 18. Of the 214 data observations in the test set, 96 were classified correctly; thus, the model achieved a test accuracy of 0.44680 (to 5.d.p.). Table 15 shows the precision, recall, and f1-scores for both of the sentiment classes.

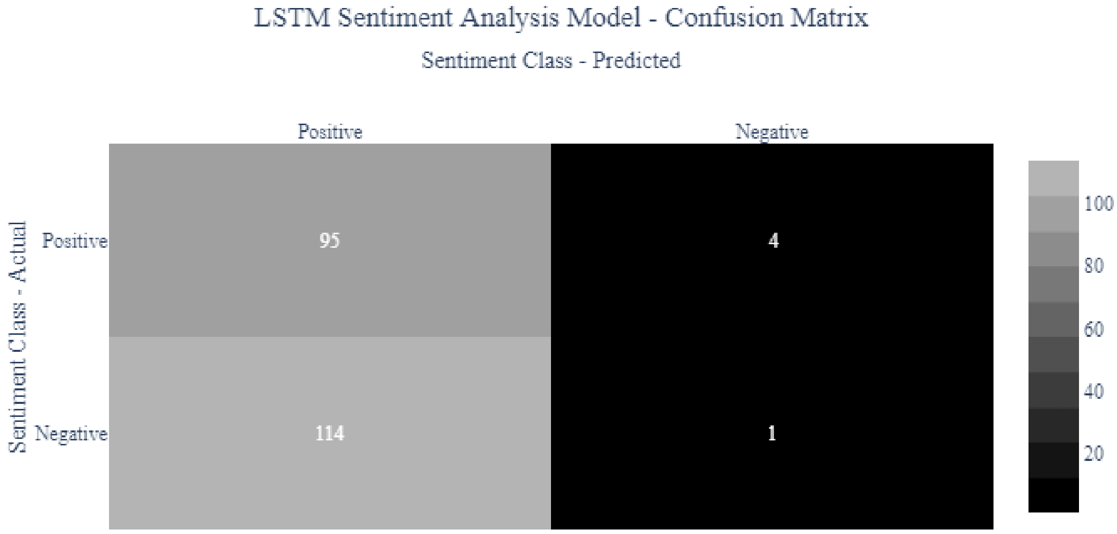

Figure 18: Confusion matrix of LSTM architecture

Sentiment Class	Precision	Recall	F1-Score
Positive	0.45455	0.95960	0.61688
Negative	0.20000	0.00870	0.01667

Table 15: Precision, recall, and f1-Score of **Sentiment_Class** classes

5 Discussion

5.1 Evaluation of Objectives

5.1.1 Implement Piotroski's Methodology and Compute Corresponding Returns

A good understanding of the process and rationale behind the methodology of Piotroski's (2000) F-Score investment strategy was obtained during the completion of the literature survey in this project. The relevant financial statement data variables required to compute Piotroski's F-Score were successfully identified and retrieved from the Compustat database (Wharton Research Data Services, 2019). Once the data was obtained, Piotroski's methodology was successfully implemented in a systematic manner via Python. The annualized mean returns and annualized mean excess returns for a collection of portfolios were calculated successfully also. By dividing the high B/M companies into the portfolios that Piotroski had defined in his research, it was found that the short-only portfolio of high B/M companies with low F-Scores provided the most annualized mean excess return in the time period between January 1, 2011, and September 2, 2019. The annualized mean excess return for this was a significant 102.80%.

The portfolio composition of long-positioned high B/M companies with high F-Scores and short-positioned high B/M companies with low F-Scores did not provide significant returns as reported by Piotroski. One reason contributing to this result was the poor performance of long-positioned high B/M companies with high F-Scores; the negative annualized mean excess returns of this individual portfolio was negating the effects of the short-positioned high B/M companies with low F-Scores.

If the tasks in this objective were to be repeated, the emphasis would be put on obtaining more data and for more companies (i.e., on multiple US exchanges) to conduct significance tests on returns. Overall, given that Piotroski's methodology was implemented and the corresponding returns were computed, this project objective was completed with high success.

5.1.2 Automate Piotroski's Methodology with the use of a Suitable Neural Network Architecture

Given that the task of predicting Piotroski's F-Score was a multi-class classification problem, the multi-layer perceptron (MLP) was chosen to automate the investment strategy. Initially, one large MLP network architecture was implemented and every feature involved in computing the F-Score was used as input to the network. It was quickly realized that the nine fundamental measures (F_ROA, F_CFO, etc.) in Piotroski's methodology were discontinuous functions. Through the paper written by Hornik et al. (1989), it was identified that MLP architectures cannot train to learn functions with discontinuities. Therefore, the nine fundamental measures (ROA, CFO, etc.) were divided into nine different MLP architectures which were used to predict the value of their corresponding scoring functions. The output of these nine MLP networks was used as input to a final MLP architecture to produce the F-Score.

Once the final MLP architecture had been trained and hyperparameters sourced, it was tested on an independent test set; the network achieved a remarkable accuracy of 0.95721. Given the significant performance of the MLP architecture, it could undoubtedly save investment management professionals time when calculating the F-Score of newly released financial statement data for various companies. The only expense of using an MLP (or any other neural network) is the time required to source the most optimal hyperparameters. However, the use of big data frameworks could reduce the time taken to source the set of optimal hyperparameters for neural networks, and thus training times as a whole (Chung et al., 2017). This would definitely be considered when conducting further work on the automation of investment strategies, such as Piotroski's F-Score.

5.1.3 Label the Sentiment of MD&A narrative data

The extraction of MD&A narratives for various companies via the SEC EDGAR database (Securities and Exchange Commission, 2019) was required in order to complete this objective. This step was the most taxing in completing the overall objective. The process of sourcing the filing page URLs, and conducting checks on whether the filing dates were correct was the most inefficient stage of the MD&A narrative extraction task. This is due to a considerable amount of data processing that had to be completed manually. The Python packages Requests and Beautiful Soup were great in providing direct access to the HTML/XML code of any web page. It was observed at the commencement of this project that preprocessed SEC annual filings were available from various financial data vendors (e.g., Wharton Research Data Services), however, the subscription fee to obtain any of these services was very expensive. In a real-life investment management environment, the time to obtain any data of this kind using web-scraping methods would be highly infeasible.

Once the MD&A narrative data had been obtained, the Loughran and Mcdonald (2011) financial dictionary was used to label the filings' corresponding sentiment using a word count of positive and negative phrases. Results from the feasibility analysis showed that this labelling method was not viable as the annualized mean excess return corresponding to the sample of high B/M companies with a high F-Score (including data observations with an F-Score of 7 or above) and a positive sentiment was negative. It should be noted that the higher frequency of words in the negative list made the labelling quite bias towards negative sentiments. Therefore, the use of a company's long-positioned annualized mean returns were used to label its corresponding MD&A narrative. Due to the vast amount of contextual differences in financial-oriented phrases, labeling the sentiment of the MD&A narratives has proved to be difficult. Manually labeling the data would require a significant amount of time; therefore, unsupervised machine learning techniques might be feasible if completed again.

All in all, this objective was met although with a method that did not involve gaining an understanding of the semantics of each MD&A narrative.

5.1.4 Automate the Sentiment Analysis of MD&A textual data

As this objective was associated with the main innovative part of this project's research, it was hugely disappointing to see that the results of the LSTM network were inferior. The test set accuracy of the LSTM network was a low 0.44680. This could be explained by the truncation of the MD&A narrative vectors to a length of 500; by disregarding the rest of the data, the neural network could have not seen any patterns that might have been present. However, making the numerical vector representations of the MD&A any longer than a length of 500 proved to be infeasible; this would have increased the training time of the LSTM network drastically and would have not been executable on a standard

CPU. Therefore, future considerations of this part of the project would be to implement the network using a big data framework with the full length of each MD&A narrative. Furthermore, a more ideal word embedding layer could be used in the model's architecture, such as Word2Vec (Mikolov et al., 2013) or Doc2Vec (Le and Mikolov, 2014) to locate deeper semantic relationships between the words in MD&A narrative data.

5.2 Answering The Research Question

Can the sentiment of a company's management's discussion and analysis narrative via a neural network improve the returns of Piotroski's methodology?

The aim of this paper was to identify whether a company's MD&A narrative could be used to improve the returns of Piotroski's F-Score investment strategy (2000) via a neural network. From the results obtained in this paper the evidence suggests it is not possible to improve the returns of such a strategy with the use of LSTM network. However, given the average computational resources used in this project, future work could be based around implementing a similar network architecture on more powerful resources.

Although no supporting evidence was obtained for the research question, there were definite products of this project that could be taken in this paper and used in the investment management domain; for example, the MLP architecture that was defined to automate Piotroski's methodology could provide an efficient solution in computing the F-Score for a company's newly issued financial statement data.

6 Evaluation, Reflections, and Conclusions

6.1 Overall Evaluation and Conclusion

The results presented in this project suggest that the returns of Piotroski's F-Score investment strategy cannot be improved by the sentiment of companies' management's discussion and analysis narratives through the use of an LSTM neural network. However, the implementation of the LSTM network was limited to a standard CPU which was deemed to be unsuitable for handling full-length MD&A narratives. Therefore, fragments of such narratives were used to train the LSTM network. The network's performance metrics suggested that any patterns present in the truncated MD&A narratives were difficult to identify.

The overall findings from the project objectives have been significantly informative, regardless of the final result. For example, through the use of particular regular expressions, it has been revealed that the majority of company MD&A narratives can be scraped from the SEC EDGAR database. On the whole, the choice of objectives has been vital in ensuring that the research question was answered. Most of the objectives laid out in this project have been near-identical to those in the original project proposal (refer to Appendix A), although they have been presented in a streamlined format. The most significant change that was made in this project in comparison to the project proposal was that the central sentiment analysis system is based on a polarity sentiment orientation (i.e., positive and negative) instead of discrete scores (e.g., 0, 1, etc.). Furthermore, an MLP architecture was used to implement Piotroski's methodology instead of an recurrent neural network architecture; this was changed after discovering the discontinuities in the nine fundamental scoring functions of the investment strategy. The data employed in Piotroski's methodology is not 'sequence data', and therefore a recurrent neural network was decided to be infeasible for automation purposes.

Whilst the project plan outlined in the original proposal was mildly adequate, the collection and extraction of MD&A narratives had taken a considerable amount of time than originally allocated. In retrospect, appropriate identification of more accurate estimation times would have resulted in more available time for modelling. As mentioned in section 1.5, work on this project was halted midway through the time allocated due to unforeseen circumstances. No improvements in the project plan could have accounted for the unforeseen circumstance, and thus the month and a half lost was inevitable.

In conclusion, this project implemented Piotroski's methodology for companies listed on the NYSE between the periods January 1, 2011, to September 2, 2019, and calculated corresponding returns. Piotroski's methodology was automated using a unique MLP architecture defined to deal with discontinuities, and evaluated with well-known machine learning metrics (e.g., recall). The MD&A narratives data for particular companies were obtained via web-scraping techniques on the SEC EDGAR database; these were labelled with a polarity sentiment based on annualized mean returns. Lastly, an LSTM network was defined to classify the sentiment of MD&A narratives.

6.2 Proposed Future Work

The main area of proposed future work would be on implementing the LSTM architecture on a big data framework (e.g., Apache Spark). This would allow the model to be trained on full-length MD&A narratives and thus gain a better understanding of any patterns or relationships that are present in the textual data.

Different types of recurrent neural network architectures could be implemented (e.g., gated recurrent unit networks (Bengio et al., 2014)) to assess which one performs the best. Once again, training multiple recurrent neural network architectures on a standard CPU would be highly infeasible, and therefore would require an implementation solution on a big data framework.

The use of a pre-trained word embedding, such as Word2Vec (Mikolov et al., 2013), within the LSTM neural network, would have also been an interesting area of further work. Custom word embedding models would most likely provide better representations of the context of words within a given vector than the standard Keras embedding layer.

6.3 Personal Reflections

From a personal perspective, this was a challenging project due to the techniques involved. Given no previous experience in sentiment analysis, or natural language processing as a whole, the opportunity to study methods in this area has provided a solid foundation on which direction future research is headed. The most rewarding aspect of the project has been learning about different implementation techniques (e.g., LSTM networks in Keras, web scraping, etc.).

The deadlines set out in the planning of the original project proposal were slightly too ambitious as data collection for the MD&A narratives were not fully looked into. Furthermore, the use of a standard CPU to training an LSTM neural network on textual data of very long lengths was also too ambitious and could be considered as an unwise decision that led to a negative outcome for the research question. As previously mentioned, the use of big data frameworks would be most definitely be utilized if this project was completed again.

Overall, the work conducted in this project was a satisfying undertaking. The amount of knowledge gained from reviewing a range of literature in section 2 has been invaluable. Furthermore, the practical experience gained from implementing solutions with the use of different Python packages has been invaluable also.

Glossary

10-K Filing A report that is filed by a particular US public company to the SEC. The report describes financial performance for the past fiscal year.

20-F Filing A report that is filed by a particular non-US or non-Canadian company to the SEC. Note that 20-F filings are the equivalent to 10-K filings for foreign companies that are listed on public exchanges in the US.

40-F Filing A report that is filed by a particular company domiciled in Canada to the SEC. Note that the content of 40-F filings are very similar to 10-K filings.

Accruals Revenue and/or expense adjustments that have not been accounted for by a company's fiscal year-end.

American Depository Receipt (ADR) A certificate issued by a US depository bank that represents a specified number of shares in a foreign company.

Asset Something that is considered valuable to a particular entity (e.g. company). A company can have physical assets (e.g. real estate), also known as tangible assets, and intangible assets (e.g. patents).

Book Value A company's total assets minus its total liabilities.

Book-to-Market (B/M) Ratio A ratio used to determine the value of company by comparing its book value to its market value.

Capital Financial assets (usually cash) that are acquired their various financing sources (e.g. bank).

Cash Flow from Operations The amount of cash a particular generates (or consumes) from regular operating activities.

Central Index Key (CIK) A unique identification number given to an individual company by the SEC.

Common Shares Outstanding The total amount of shares of a particular company's common stock that are owned by investors.

Common Stock A stock that gives investors voting rights on electing or removing members from a company's board of directors, and corporate policy.

Compustat A extensive database of fundamental financial and market data on active and inactive global companies, indices and industries provided by Standard and Poor's Market Intelligence.

Corporate Bond A debt security that is issued by a company to public investors.

Cost of Goods Sold The direct cost associated with producing a good or providing a service. The direct cost can include labour costs, production costs, and other costs.

Current Assets Assets that can be expected to be converted to money within a year.

Current Liabilities Liabilities that are due to be paid within a year.

Debt A amount of money that is owed to an entity (e.g. bank).

Excess Return The proportion of a security's return that is above or below the return of a investing benchmark. Note that indices (e.g. S&P 500) are usually used as benchmarks for stocks.

Exchange-Traded Fund (ETF) A type of security that can be traded on a exchange, much like stocks, although they combine the movements of multiple securities into one price..

Extraordinary Item Income or losses that are the result of events that are deemed to be unusual or infrequent in nature (e.g. losses that have been the result of an earthquake.

Fiscal Year-End The completion of a one-year accounting period for a particular company.

Holding Period Return (HPR) The total return received from a security over a specific time period. Note that the time period is known as the holding period.

Initial Public Offering (IPO) A process in which shares of a company are sold on a public exchange (e.g. NYSE).

Leverage The use of debt to undertake an investment or project and amplify returns.

Liability A company's legal debts or obligations that arise during the course of business operations.

Liquidity The degree to which an asset or security can be purchased or sold in a particular market place.

Long Position The action of purchasing a security with the expectation its market price will rise in the future.

Long-Term Debt Debt that has a maturity of more than one-year, or is to be paid in more than one year.

Margin The difference between the selling price and production cost of a good or service.

Market Value The market value of a company is defined as its market capitalization.

Net Income The total amount of sales made by a particular company minus its total expenses. Expenses can include cost of goods sold, general and administrative expenses, operating expenses, depreciation, interest, taxes, and other expenses.

Quintile A statistical value that represents 20% of any population. The first quintile represents 0% to 20%, the second quintile represents 20% to 40%, etc.

Return on Assets (ROA) A ratio of how profitable a particular company is relative to its total assets.

Securities and Exchange Commission (SEC) A independent US government agency responsible for protecting investors, maintaining fair practices in securities markets, and facilitating capital formation.

Security A financial instrument (e.g. stock) that has a monetary value and can be traded on a public exchange or over-the-counter.

Short Position The action of borrowing and selling a security with the expectation its market price will fall in the future. Note that the security will be purchased back by the investor owning the short position at a later date..

Stock A type of security that signifies a share of ownership in a particular company. Note that stock is also refer to as equity.

Tercile A statistical value that represents $3\dot{3}\%$ of any population. The first tercile represents 0% to $3\dot{3}\%$, the second tercile represents $3\dot{3}\%$ to $6\dot{6}\%$, etc.

Ticker Symbol A abbreviation used to uniquely identify a security listed on a particular exchange (e.g. NYSE).

Total Sales The total amount of units sold multiplied by the units price. Note that units can represent goods and/or services.

Turnover The total amount of sales generated by a particular company in a particular time period.

References

Andrychowicz, M. (2016), Learning to Learn by Gradient Descent by Gradient Descent, *in* '30th Conference on Neural Information Processing Systems'.

Ansell, T. M. (2016), *Python Datetime TZ Documentation*, Python Software Foundation. Release 0.2.

Asness, C., Frazzani, A., Israel, R. and Moskowitz, T. (2015), 'Fact, Fiction, and Value Investing', *The Journal of Portfolio of Management* **42**(1), pp. 34–52.

Basu, S. (1977), 'Investment Performance of Common Stocks in Relation to Their Price-Earnings-Ratios: A Test of the Efficient Market Hypothesis', *The Journal of Finance* **32**(3), pp. 663–682.

Bengio, Y., Chung, J., Gulcehre, C. and Cho, K. (2014), Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, *in* 'NIPS Deep Learning and Representation Learning Workshop'.

Bird, S. (2017), *NLTK Documentation*, NLTK Project. Release 3.2.5. Available at: https://www.nltk.org. Date Accessed: September 2019.

Boiy, E. and Moens, M. (2009), 'A Machine Learning Approach to Sentiment Analysis in Multilingual Web Texts', *Information Retrieval* **12**(5), pp. 526–558.

Bord, B. (2019), *Workalender*. Release 6.0.0. Available at: https://peopledoc.github.io/workalendar. Date Accessed: August 2019.

Cambria, E. (2016), 'Affective Computing and Sentiment Analysis', *IEEE Intelligent Systems* **31**(2), pp. 102 – 107.

Chan, L. K. C. and Lakonishok, J. (2004), 'Value and Growth Investing: Review and Update', *Financial Analysts Journal* **60**(1), pp. 71–86.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002), 'SMOTE: Synthetic Minority Over-sampling Technique', *Journal of Artificial Intelligence Research* **16**(1), pp. 321–357.

Chowdhury, G. G. (2003), 'Natural Language Processing', *Annual Review of Information Science and Technology* **37**(1), pp. 51–89.

Chung, I., Sainath, T. N., Ramabhadran, B., Picheny, M., Gunnels, J., Austel, V., Chauhari, U. and Kingsbury, B. (2017), 'Parallel Deep Neural Network Training for Big Data on Blue Gene/q', *IEEE Transactions on Parallel and Distributed Systems* **28**(6), 1703–1714.

Dang, Y., Zhang, Y. and Chen, H. (2010), 'A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews', *IEEE Intelligent Systems* **25**(4), pp. 46–53.

Denis, D. J. (2011), 'Financial Flexibility and Corporate Liquidity', *Journal of Corporate Finance* **17**(3), pp. 667–674.

Dozat, T. (2016), Incorporating Nesterov Momentum into Adam, *in* "International Conference for Learning Representations'.

Esuli, A. and Sebastiani, F. (2006), 'SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining', *In Proceedings of the 5th Conference on Language Resources and Evaluation,* pp. 417–422.

Fama, E. F. and French, K. F. (1992), 'The Cross-Section of Expected Stock Returns', *The Journal of Finance* **47**(2), pp. 427–465.

Fama, E. F. and French, K. R. (1995), 'Size and Book-to-Market Factors in Earnings and Returns', *The Journal of Finance* **50**(1), pp. 131–155.

Graham, B. and Dodd, D. D. (2009), *Security Analysis: Principles and Technique*, 6th edn, McGraw-Hill, New York.

Graham, B. and Zweig, J. (2003), *The Intelligent Investor: A Book of Practical Counsel*, 4th rev. edn, HarperBusiness Essentials, New York.

Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, 2nd edn, Pearson Prentice Hall, Upper Saddle River, New Jersey.

Haykin, S. (2009), *Neural Networks and Learning Machines*, 3rd edn, Pearson Prentice Hall, Upper Saddle River, New Jersey.

He, H., Bai, Y., Garcia, E. A. and Li, S. (2008), ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, *in* '2008 IEEE International Joint Conference on Neural Networks', pp. 1322–1328.

Hochreiter, S. and Schmidhuber, J. (1997), 'Long Short-term Memory', *Neural Computation* **9**(8), pp. 1735–80.

Hornik, K., Stinchcombe, M. and White, H. (1989), 'Multilayer Feedforward Networks are Universal Approximators', *Neural Networks* **2**(5), pp. 359–366.

Hutto, C. J. and Gilbert, E. (2014), 'VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text', *In Proceedings of the 8th International AAAI Conference on Weblogs and Social Media,* pp. 216–225.

Japkowiz, N. and Stephen, S. (2002), 'The Class Imbalance Problem: A Systematic Study', *Intelligent Data Analysis* **6**(5), pp. 429–449.

Jurafsky, D. and Martin, J. H. (2008), 'Naïve Bayes and Sentiment Classification', *in* '*Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*', 2nd edn, Pearson Prentice Hall, Upper Saddle River, New Jersey, chapter 4, pp. 63–81.

Kabir, M., Kabir, M. M. J., Xu, S. and Badhon, B. (2019), 'An Empirical Research on Sentiment Analysis Using Machine Learning Approaches', *International Journal of Computers and Applications,* pp. 1–9.

Keras (2019), *Keras Documentation.* Release 2.0.0. Available at: https://keras.io/documentation. Date Accessed: August 2019.

Kingma, D. P. and Ba, J. (2015), Adam: A Method for Stochastic Optimization, *in* "International Conference for Learning Representations'.

Kohavi, R. (1995), 'A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection', *'95 Proceedings of the 14th International Joint Conference on Artificial intelligence* **2**(1), pp. 1137–1143.

König, A. C. and Bill, E. (2006), 'Reducing the Human Overhead in Text Categorization', *'06 Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 598–603.

Kuchling, A. (2019), *Unicode Howto Documentation*, The Python Software Foundation. Release 1.12.0. Available at: https://docs.python.org/3/howto/unicode.html. Date Accessed: September 2019.

Lai, S., Xu, L., Liu, K. and Zhao, J. (2015), 'Recurrent Convolutional Neural Networks for Text Classification', *'15 Proceedings of the 29th AAAI Conference on Artificial Intelligence,* pp. 2267–2273.

Lakonishok, J., Shleifer, A. and Vishny, R. W. (1994), 'Contrarian Investment, Extrapolation, and Risk', *The Journal of Finance* **49**(5), pp. 1541–1578.

Le, Q. and Mikolov, T. (2014), 'Distributed Representations of Sentences and Documents', *Proceedings of the 31st International Conference on Machine Learning* **32**(2), pp. 1188–1196.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), 'Deep Learning', *Nature* **521**(7553), pp. 436–444.

Lee, C. M. C., Myers, J. and Swaminathan, B. (1999), 'What is the Intrinsic Value of the Dow?', *The Journal of Finance* **54**(4), pp. 1693–1741.

Lev, B. and Thiagarajan, S. R. (1993), 'Fundamental Information Analysis', *Journal of Accounting Research* **31**(2), pp. 190–215.

Loughran, T. and Mcdonald, B. (2011), 'When is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks', *The Journal of Finance* **66**(1), pp. 35–65.

Lovins, J. B. (1968), 'Development of a Stemming Algorithm', *Mechanical Translation and Computational Linguistics* **11**(1), pp. 22–31.

Lu, X. (2014), *Computational Methods for Corpus Annotation and Analysis*, Springer, Dordrecht.

Lui, B. (2015), *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, Cambridge University Press, Cambridge.

McKinney, W. (2010), 'Data Structures for Statistical Computing in Python', *'10 Proceedings of the 9th Python In Science Conference,* pp. 51–56.

McKinney, W. and PyData (2019), *Pandas: Powerful Python Data Analysis Toolkit*, PyData. Release 0.25.1.

Medhat, W., H. A. and Korashy, H. (2014), 'Sentiment Analysis Algorithms and Applications: A Survey', *Ain Shams Engineering Journal* **5**(4), pp. 1093–1113.

Melville, P., Gryc, W. and Lawrence, R. D. (2009), 'Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification', *KDD '09 Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 1275–1284.

Mikolov, T., Corrado, G. S., Chen, K. and Dean, J. (2013), 'Efficient Estimation of Word Representation in Vector Space', *Proceedings of the International Conference on Learning Representations* .

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K. J. (1990), 'Introduction to WordNet: An On-line Lexical Database', *Cognitive Science* **3**(4), pp. 235–244.

Miller, M. H. and Rock, K. (1985), 'Dividend Policy under Asymmetric Information', *The Journal of Finance* **40**(4), pp. 1031–1051.

Nasdaq Inc. (2019), 'Stock Listings'. Available at: https://www.nasdaq.com. Date Accessed: June 2019.

Ortony, A., Clore, G. L. and Foss, M. A. (1987), 'The Referential Structure of the Affective Lexicon', *Cognitive Science* **11**(3), pp. 341–364.

Pang, B. and Lee, L. (2004), 'A Sentiment Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts', *ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics,* pp. 271–278.

Piotroski, J. D. (2000), 'Value Investing: The Use of Historical Financial Statement Information to Separate Winners from Losers', *Journal of Accounting Research* **38**(1), pp. 1–41.

Prabowo, R. and Thelwall, M. (2009), 'Sentiment Analysis: A Combined Approach', *Journal of Informetrics* **3**(2), pp. 143–157.

PyData (2019), *Pandas-Datareader Documentation*, PyData. Release 0.8.0+4.gec799a0.

Reitz, K. (2019), *Requests Documentation.* Release 2.22.0. Available at: https://buildmedia.readthedocs.org/media/pdf/requests/latest/requests.pdf. Date Accessed: August 2019.

Richardson, L. (2017), *Beautiful Soup Documentation.* Release 4.4.0. Available at: https://buildmedia.readthedocs.org/media/pdf/beautiful-soup-4/latest/beautiful-soup-4.pdf. Date Accessed: August 2019.

Rosenberg, B., Reid, K. and Lanstein, R. (1985), 'Persuasive Evidence of Market Inefficiency', *The Journal of Portfolio Management* **11**(3), pp. 9–16.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986), 'Learning Representations by Back-Propagating Errors', *Nature* **323**(1), pp. 533–336.

Santos, C. N. and Bayser, M. G. (2014), 'Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts', *'14 Proceedings of the 25th International Conference on Computational Linguistics,* pp. 69–78.

Schouten, K. and Frasincar, F. (2016), 'Survey on Aspect-Level Sentiment Analysis', *IEEE Transactions on Knowledge and Data Engineering* **28**(3), pp. 813–830.

Schroeder, N. and Gibson, C. (1990), Readability of Management's Discussion and Analysis, *in* 'Accounting Horizons', pp. 78–87.

Scikit-Learn (2019), *Scikit-Learn User Guide*. Release 0.21.3. Available at: https://scikit-learn.org. Date Accessed: August 2019.

Securities and Exchange Commission (2019), 'EDGAR - Company Filings'. Available at: https://www.sec.gov/edgar. Date Accessed: July 2019.

Simmons, R. F. (1973), 'Semantic Networks: Their Computation and Use for Understanding English Sentences', *in* R. C. Schank and C. K. M., eds, '*Computer Models of Thought and Language*', W. H. Freeman, San Francisco, chapter 2, pp. 63–113.

Socher, R., Huang, E. H., Pennington, J., Ng, A. Y. and Manning, C. D. (2011), 'Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection', *'11 Proceedings of the 24th International Conference on Neural Information Processing Systems,* pp. 801–809.

Sokolova, M. and Lapalme, G. (2009), 'A Systematic Analysis of Performance Measures for Classification Tasks', *Information Processing and Management* **45**(4), pp. 427–437.

Strapparava, C. and Valitutti, A. (2004), 'WordNet-Affect: An Affective Extension of WordNet', *In Proceedings of the 4th International Conference on Language Resources and Evaluation,* pp. 1083–1086.

Sundermeyer, M., Schlüter, R. and Ney, H. (2012), LSTM Neural Network for Language Modeling, *in* '13th Annual Conference of the International Speech Communication Association'.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. and Stede, M. (2014), 'Lexicon-Based Methods for Sentiment Analysis', *Computational Linguistics* **37**(2), pp. 267–307.

Tausczik, Y. R. and Pennebaker, J. W. (2010), 'The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods', *Journal of Language and Social Psychology* **29**(1), pp. 24–54.

Webster, J. J. and Kit, C. (1992), 'Tokenization as the Initial Phase in NLP', *COLING '92 Proceedings of the 14th Conference on Computational linguistics* **4**(1), pp. 1106–1110.

Weiss, G. M., McCarthy, K. and Zabar, B. (2007), Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?, *in* 'DMIN'.

Wharton Research Data Services (2019), 'Compustat Daily Updates Database'. Available at: https://wrds-www.wharton.upenn.edu. Date Accessed: July 2019.

Wilbur, W. J. and Sirotkin, K. (1992), 'The Automatic Identification of Stop Words', *Journal of Information Science* **18**(1), pp. 45–55.

Williamson, J. P. (1971), *Investments: New Analytic Techniques*, Praeger, Santa Barbara.

Wu, L., Hoi, S. C. H. and Yu, N. (2010), 'Semantics-Preserving Bag-of-Words Models and Applications', *IEEE Transactions on Image Processing* **19**(7), pp. 1908–1920.

Appendix A - Project Proposal

INM363 Research Methods and Professional Issues Coursework – Research Project Proposal

1 | Introduction

Over the past 80 years, value investing has become a popular long-term investment paradigm for value investors. The origins of value investing date back to the late 1920s when Columbia Business School professors Benjamin Graham and David Dodd (2009) advocated the important idea of purchasing profitable, although undervalued, securities through identifying a combination of opportunities on a company's financial statements. These included the following measures:

- A lower than average price-to-book (P/B) ratio.
- A lower than price-to-earnings ratio (P/E) ratio.
- A higher than average dividend yield.

In recent years, several academic researchers and investors have proposed strategies that build upon the Graham and Dodd framework. A successful example of this is Piotroski's F-Score investment strategy developed by Stanford professor Joseph D. Piotroski (2000). Piotroski's F-Score framework measures the strength of nine 'fundamental signals' using a discrete scoring system to ultimately determine the financial position of a company in a given time period. In his famous research paper 'Value Investing: The Use of Historical Financial Statement Information to Separate Winners from Losers' Piotroski reports that the strategy generated an annual return of 23% between 1976 and 1996. Whilst Piotroski's F-Score investment strategy employs intensive fundamental analysis techniques to identify value equities, it disregards any valuable information that may be contained within the narrative of a company's annual report. Therefore, the purpose of my proposed research project will be to address the latter. More specifically, I propose a natural language-based extension of Piotroski's F-Score framework to include a suitable sentiment score for the narrative in a company's annual report as well as the original 'fundamental signals' scores. As human text is presented in a sequence, it would make sense to use a natural language processing (NLP) architecture driven by deep recurrent neural networks (RNNs) to summarise relevant annual report narratives and produce the final sentiment scores. Once the sentiment scores and Piotroski's F-Score have been computed, another RNN architecture will be employed to retrieve stock selection recommendations. These recommendations will be back-tested to identify the potential return/loss of the strategy.

1.1 Title of Proposed Project

An NLP-Based Extension of Piotroski's F-Score Investment Strategy for Stock Selection: Integrating a Sentiment Scoring System for Company Annual Reports Using an NLP Architecture Driven by RNNs

1.2 Aims

The proposed research project should:

- Extend Piotroski's F-Score investment strategy by incorporating an NLP sentiment score for the narrative of a company's annual reports using RNNs.
- Employ a further RNN to retrieve stock recommendations based on Piotroski's F-Score methodology and annual report sentiment scores.

1.3 Research Questions and Objectives

The research questions posed for my proposed research project are:

- How can we extend Piotroski's F-Score investment strategy to consider qualitative aspects of a company's annual reports?

- Are there any significant anomalies in the stocks that have been recommended by the extended scoring system?
- How does the proposed investment strategy perform? Are the returns of the strategy below, identical or above the results reported by Piotroski?

The research objectives for my proposed research project are therefore:

1. To identify the underlying processes involved in Piotroski's F-Score investment strategy and attempt to reproduce his results in a rule-based systematic manner.
2. To explore any research attempted in similar areas by searching and reviewing relevant literatures.
3. To identify any limitations of Piotroski's research and methods found whilst completing objective 2.
4. To implement an NLP sentiment scoring system for the narrative of a company's annual reports using deep RNNs.
5. To implement a further deep RNN to produce stock recommendations using the scores produced by Piotroski's approach and the NLP sentiment scoring system?
6. To back-test the proposed investment strategy on a suitable time period and compare returns with Piotroski's original strategy.
7. To evaluate the performance of the proposed investment strategy and the extent to which it addresses the problems identified in objective 3.

1.4 Products of the Work and Beneficiaries

The products of my proposed research project will include: a thorough understanding of Piotroski's F-Score Investment strategy and its limitations; an NLP-architecture driven by RNNs to summarise narratives within the annual reports of US companies and provide a sentiment score following Piotroski's framework; a RNN model that can provide time recommendations based on our scoring results in a sequential manner (i.e., recommendations every year, so investors can re-balance their equity portfolios); back-tested results of the proposed strategy and a comparison to Piotroski's original strategy; and lastly, suggestions on how the proposed strategy can be improved.

The beneficiaries of the work comprise of investment management industry professionals (hedge-fund managers, quantitative analysts, etc.), data scientists and academic researchers from non-specific fields focused on NLP developments.

2 | Critical Context

2.1 An Introduction to Value Investing

Value investing has undoubtedly had the most influence on the equity investment landscape for over nearly a century (Asness et al., 2015). The formation of 'value investing' was introduced by Benjamin Graham and David Dodd (2009) through a series of predefined fundamental analysis metrics and the important concept of 'margin of safety'. Note that three fundamental metrics were highlighted in section 1. Graham defined the margin of safety to be the difference between a company's equity price and 'intrinsic value'; a measure that attempts to reflect the fundamental value of a company at a given period (Graham, 1976). In his famous book, the intelligent investor, Graham (1973) states "the function of the margin of safety is, in essence, that of rendering unnecessary an accurate estimate of the future. If the margin is a large one, then it is enough to assume that future earnings will not fall far below those of the past in order for an investor to feel sufficiently protected against the vicissitudes of time". Here, Graham implies that an individual investor should consistently act as an investor and not as a speculator given, they have a considerable margin of safety on the selected equities in their portfolio. To discourage speculative behaviour and considerable risk, Graham (1973) outlined three important principles that value investors should follow:

- A long-term investor should thoroughly analyse the development of a company's accounting fundamentals and management structure/philosophy.
- A long-term investor should always protect him or herself from losses by using the concept of a margin of safety and diversification.
- A long-term investor should never search for short-term profits; he or she should focus on steady returns.

Although value investing is an ambiguous topic in investment management, Graham's teachings proved to be well-defined as many value strategies have been documented in the past; relevant work include investigations by Chan and Lakonishok (2004) and, by Pätäri et al. (2018).

2.2 Piotroski's F-Score Investment Strategy

Joseph D. Piotroski's F-Score investment strategy is a simple accounting-based fundamental analysis approach that focuses on companies with a high book-to-market (B/M) ratio. Fama and French (1992) observe that "firms with high B/M ratios and those that the market judges to have poor prospects, have higher expected stock returns than firms with strong prospects". However, Piotroski (2000) argues that an investment strategy solely based on a high B/M ratio and poor prospects suffers from tolerating the performance of genuinely deteriorating firms. Therefore, Piotroski (2000) investigates whether a financial statement analysis approach can distinguish between firms that have strong prospects from those that have poor prospects. Piotroski's F-Score strategy consists of measuring nine "fundamental signals" in three areas of a firm's financial statements to assess its financial condition; these are as follows (Piotroski, 2000):

- PROFITABILITY – The profitability position of a firm provides considerable information on its ability to generate funds internally. The fundamental signals of interest in the area of profitability are; return on assets (ROA), operating cash flow (OCF), change in ROA (ΔROA) and ACCRUAL. ROA is defined to be a firm's net income before any extraordinary items (i.e. gains or losses that have been recorded from rare circumstances) have been included. OCF is defined to be a firm's net income before the cash flow from operations have been included. ΔROA is the change in ROA between the current year and previous year. ACCRUAL is defined to be a firm's net income for the current year before any extraordinary items less cash flow from operations; note that ACCRUAL is scaled by the firm's total assets at the beginning of the year.
- LEVERAGE, LIQUITY, AND SOURCE OF FUNDS – The leverage, liquidity and source of funds position of a firm provides an insight into its capital structure and ability to repay (current and future) debt. In this area, the fundamental signals that are used are; the change in leverage (ΔLEVER), change in liquidity (ΔLIQUID) and the issue of common stock (EQ_OFFER). The ΔLEVER is defined as the change in a firm's long-term debt structure; if a firm frequently raises funds through external debt services (e.g. issuing corporate bonds) than this can be a sign that it does not have enough internal sources of funds. The ΔLIQUID is defined to be the difference between current assets and liabilities of the current year and the previous year. EQ_OFFER is defined to the firm's total amount of equity (stock) issuance.
- OPERATING EFFICIENCY – The operating efficiency of a firm provides a considerable amount of information on whether operations are becoming more efficient in terms of costs, productivity, etc. The financial signals used to monitor operating efficiency are; changes in gross margin ratio (ΔMARGIN) and changes in asset turnover ratio (ΔTURN). ΔMARGIN measures the change in a firm's gross margin ratio in the current year against the previous year; a high ΔMARGIN signifies a decrease in the cost of sale or an increase in a firm's revenue (or a combination of both). ΔTURN measures the change in a firm's asset turnover

ratio in the current year against the previous year; a high ΔTURN signifies a increase in operating productivity.

Given the nine fundamental signals listed above, Piotroski (2000) defined a scoring system based on the rules shown in Figure 1. A firm with strong prospects should have a significantly high F-Score (8 or 9); a firm with poor prospects should have a significantly low F-Score (0, 1 or 2). Investors are advised to invest in the companies with the highest F-Scores and B/M ratios.

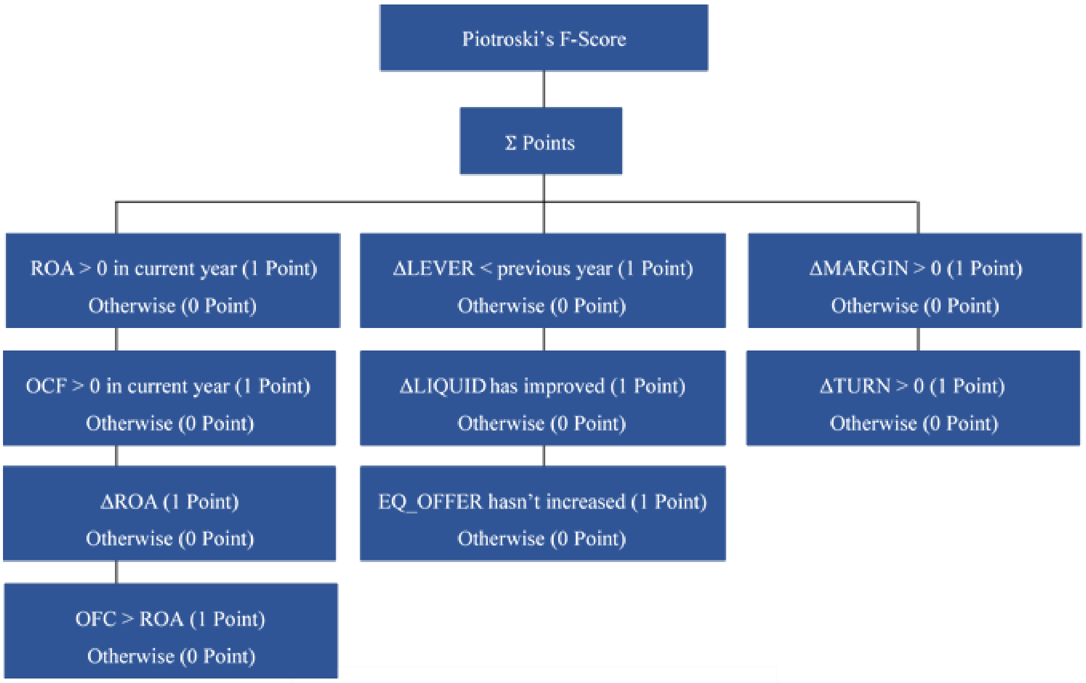

Figure 1: Piotroski's F-Score Calculation

Pätäri et al. (2018) investigated the value-added by using Piotroski's F-Score investment strategy in conjunction with the book-to-price ratio (note that the B/P ratio is an inverse of B/M ratio). The findings suggested that using the Piotroski's strategy boosted a portfolio's returns significantly. However, it is argued that the process of the F-Score methodology is very time consuming; this could be easily automated through a systematic way.

2.3 NLP Architectures for Sentiment Analysis

Natural Language Processing (NLP) uses computational and statistical techniques to extract and comprehend human language content. Hirschberg and Manning (2015) highlight that most of the research in the field of NLP has been focused on sentiment analysis; i.e., the identification of positive, negative and neutral sentiments in textual language. Detection of sentiment within textual language requires one of two approaches; a lexicon (dictionary) approach or a granular sentence-based information approach. The availability of online lexicons makes it convenient to employ a lexicon approach to sentiment analysis. In relations to the financial domain, Loughran and McDonald (2011) identified that word lists (or dictionaries) misclassify common words in financial text most of the time; therefore, Loughran and McDonald (2011) developed a sentiment specific lexicon for 10-K reports (fillings). Note that 10-K reports are the US equivalent to annual reports. This is a useful resource since it is specific to text contained in annual reports of US companies. NLP tasks are usually investigated through 'syntactic' analysis or 'semantic' analysis. Syntactic analysis involves assessing the extent to which the structure of natural text follows grammatical rules. Semantic analysis involves understanding the actual meaning of natural text. Collobert *et al*. (2011) proposes a neural network architecture and learning algorithm that can be applied to syntactic and semantic analysis tasks; this comprises of transforming various corpus' into feature vectors, training a multi-layer neural network on a variety of labelled and unlabelled data (and extracting high level features in the process), testing the architecture on unseen data through a mixture of syntactic and semantic tasks (e.g. part of speech tagging - POS, name entity recognition - NER, etc.) and evaluating results. The option of using a multi-layer neural network to conduct sentiment analysis on financial data has limitations. For example, the sequential nature of financial statements (i.e. released on a annual basis) means that an ordinary deep layered neural network would fail to process the data in an orderly fashion and provide 'memory-driven' sentiment scores (i.e. the previous year's annual statements could be given a high score, whilst the current year could be given a low score; therefore, it would not factor in previous fundamental results). Mikolov *et al*. (2011) presents a recurrent neural network architecture using a backpropagation through time algorithm (Rumelhart, Hinton and Williams, 1986) for natural language. Mikolov *et al*. (2011) proposed architecture is 'memory-driven', especially with the use of a backpropagation through time algorithm (BPTT); this would be ideal for sentiment-scoring-based applications in the financial context due to most data being of temporal nature.

2.4 The Case for Deep Learning Models

Albert and Lipton (2017) emphasize the power of using deep learning neural network models in forecasting company fundamentals (e.g. B/M) to predict equity price. The layered architecture of a neural network has the quality of learning the importance of features within a dataset with minimal pre-processing (Haykin, 2009). However, given the fact that a company's financial statements contain a large amount of financial metrics, a shallow neural network could potentially find it difficult to find any significant importance amongst features. Therefore, using a deep neural network could alleviate this problem. Figure 2 shows a shallow and deep neural network architecture.

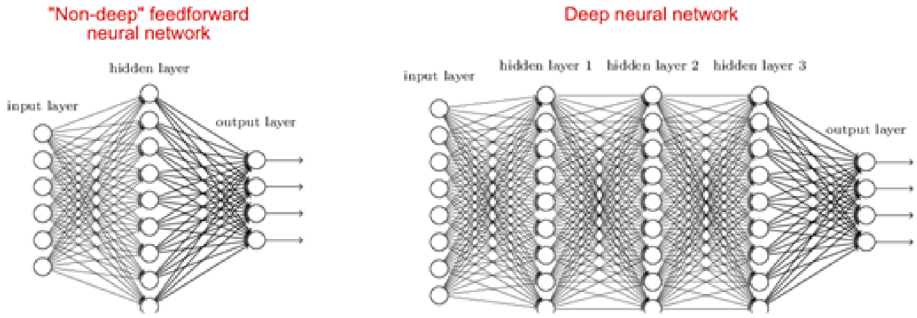

Figure 2: Neural Network Architecture - Shallow vs Deep

3 | Approaches: Methods and Tools for Design, Analysis and Evaluation

The following subsections outline the methodology and necessary steps needed to provide credible answers for the research questions proposed in the introduction of this proposal. Note that these steps are subject to alteration.

3.1 Literature Search and Review

As a primary step, a thorough search and review of relevant literature will be completed further to the critical context explored in this research project proposal. To conduct the search and review I will use various sources that are readily available (e.g. IEEE Xplore Digital Library). To effectively manage time, I will employ the selective procedures outlined in Dawson (2009). More specifically, I will review the abstract of research papers (introduction/preface for books) that appear relevant, identify keywords specific to our research project objectives and assess credibility via references and citations. The main purposes of searching and reviewing relevant literature are to understand what has been attempted with NLP techniques (e.g. lexicon based solutions) in the area of value investing, what are the limitations involved and have they been explored and most importantly, to help me get to speed with methods I am unfamiliar with.

3.2 Data Sourcing and Pre-processing

The data required for the proposed research project will be company annual reports, annual financial statements and daily equity prices. Based on Piotroski's analysis (2000), I will focus my NLP automated investment strategy on US mid-cap and large-cap companies. To retrieve company annual reports (which are known as 'Form 10-K' reports for US companies), I will parse relevant XML-formatted data from the SEC (Securities and Exchange Commission) EDGAR (Electronic Data Gathering, Analysis, and Retrieval System) database into csv files using Python. Note that the SEC EDGAR database uses a CIK (Central Index Key) number to identify different public companies; the CIK for companies of interest will have to be retrieved too. The annual financial statements for the relevant US companies can also be retrieved through the SEC EDGAR database, although the quality of older data might be considerably low. If this is the case, I will source the data from a financial data provider such as Quandl or Standard and Poor's, although this will be at cost. Lastly, the daily equities data will be sourced from Quandl using a database API provided by QuoteMedia. Once the data has been obtained, exploratory data analysis (EDA) will be conduction in Python; this will be an opportunity to identify any missing or invalid data, identify any feature significances/relationships and transform the data if needed. My experience with Python's data analysis and visualisation libraries (e.g. Pandas, Matplotlib, etc.) should be useful during the EDA phase in the proposed project.

3.3 Design, Implementation and Testing of NLP Architecture Driven by RNNs

Based on my literature search and review, I will identify and implement a suitable NLP architecture driven through RNN models (i.e., a modification of the standard word2vec model) to compute sentiment scores summarised company annual report narratives. Note that the summarisation of company annual reports will be conducted through a separate deep RNN transformer using word embeddings. Nasukawa and Yi (2003) present an NLP architecture to produce a sentiment for web pages and news articles; their system applies semantic analysis with a syntactic parser following a sentiment lexicon. Nasukawa and Yi (2003) sentiment lexicon approach could potentially be modified and applied to my proposed sentiment scoring system. As I will be using Python for most of my proposed research project, I will have to become familiar with a few deep learning and NLP libraries (e.g. Keras, NLTK, etc.). Furthermore, the implementation and training of any models using vast volumes of annual report and financial statement data might not be viable (i.e. computation could be very slow). Therefore, big data frameworks such as Apache Sparke or cloud services with integrated big data technologies might be beneficial. If this is the case, I will have to also become familiar with these technologies. The output of the proposed NLP RNN should be a score that follows the same structure as Piotroski's F-Score measurement (i.e. give a summarised annual report narrative a sentiment of 1-9 with 1 being negative and 9 being positive). The success of the RNNs could be quantified by an existing metric; for example, the Bilingual Evaluation Understudy (BLEU) measure proposed by Andrew Ng.

3.4 Design and Implementation of Stock Recommendation RNN Model

Once sentiment scores have been produced and Petroski's F-Score measures have been calculated systematically (for every company's financial statements), another deep RNN architecture will be implemented to produce stock recommendation in a supervised fashion (i.e., each and every company will be labelled 'long' or 'short' in accordance to the F-Score and sentiment results obtained in section 3.3). To assess the performance of the deep learning models used in my project, a series of loss metrics will be used (e.g. Mean-Squared Error – MSE); note the specific metrics are yet to be decided. Furthermore, as the stock recommendation RNN will be supervised, confusion matrix analysis will be conducted.

3.5 Evaluation

The evaluation will be conducted throughout the proposed research project in the form of mathematical/statistical results and visualisations. Results from back-testing the proposed investment strategy will be analysed and compared to those of Petroski's original F-Score investment strategy. Any issues with design/implementation will be thoroughly documented and reasoned in the evaluation. Furthermore, the usability and suitability of the proposed will be examined in detail; questions, such as 'how would the implementation of the proposed strategy be suitable for industry?', will be discussed to explore any possible future work. This process will help to answer the three research questions outlined in section 1.3.

3.6 Ethical, Legal and Professional Issues

In consideration of City University of London's research ethics review questionnaire, which is situated at the end of this proposal, no questions required further action. All the work that is proposed in this report should be successfully completed without any ethical, legal or professional issues.

4 | Work Plan

The organisation of my proposed project is detailed in Figure 3. The project will commence on the 10th June 2019 and finish on the 30th September 2019 with a contingency of one extra week. A period of three weeks has solely been allocated to writing the report, although this will be continually visited whilst the work is being completed.

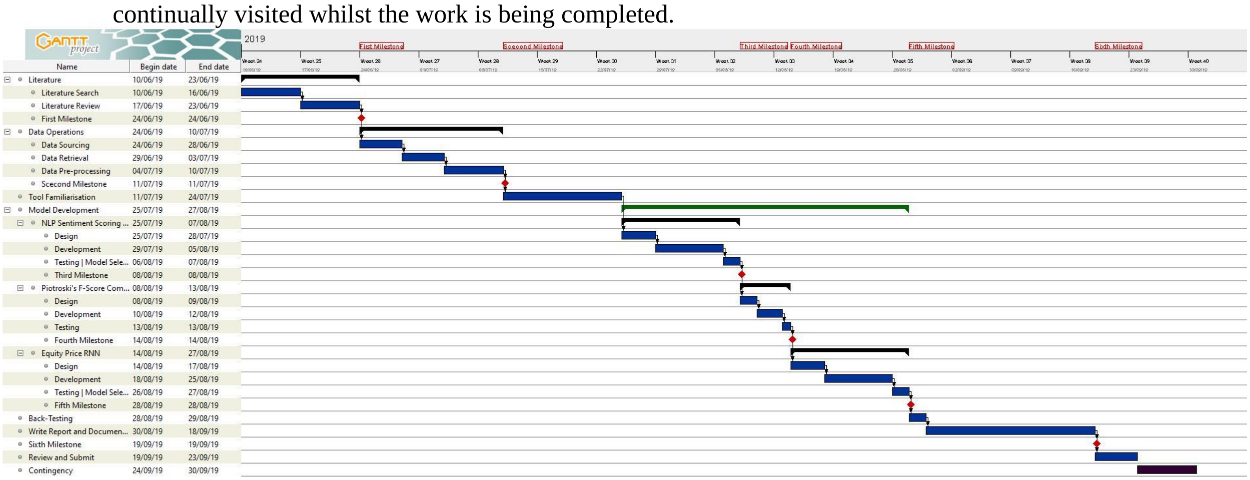

Figure 3: Work Plan Gantt Chart

4 | Risks

Dawson (2009) mentions that “risk management involves the identification of risks at the beginning of a project and the control of those risk as the project progresses”. Therefore, I have produced a risk register for the proposed research project to identify possible risks and mitigations to those risks. The likelihood (1 - unlikely to 3 – very likely), consequence (1 – minimal to 5 – maximum) and impact (likelihood * consequence) of each risk are also considered.

Risk	Likelihood (1-3)	Consequence (1-5)	Impact (L*C)	Mitigation
Incomplete literature search and review	1	4	4	Allocate suitable priority to literature search and review; complete at the beginning of project.
Hardware failure	1	4	4	Use/Purchase alternative hardware; back-up reports, code, etc. on regular basis.
Loss of motivation	1	3	3	Choice of topic is highly of interest.
Sudden illness/injury/accident	1	5	5	Exercise regularly and aim for work-life balance.
Incorrect planning/timing estimations	2	3	6	Notify supervisor ASAP and allocate extra time (7 days of contingency).
Any ethical issues that are missed/not addressed	1	5	5	Address missed issues ASAP and notify supervisor; on revision ask for support.
Incomplete evaluation	1	5	5	Revise project timing and allocate time from contingency time. Solely focus on evaluation until its at a suitable quality.
Missing project task(s)	2	4	8	Revise work plan and add missing task(s) – once again, allow extra time from contingency.
Internet/Network outage	2	3	6	Proceed to complete project at university/local library or purchase 4G ‘dongle’.
Use of wrong methodology/algorithms	1	4	4	Revisit relevant literature and decide on alternative methodology/algorithm.
Inconclusive results	1	5	5	Revise project ideology, search for any ‘gaps’ and rectify if possible, to produce meaningful results. If no issues can be identified, accept result and ensure analysis is solid.

Table 1: Risk Register

Albert, J., and Lipton, Z. C. (2017) ‘Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals’ *Euclidean Technologies Management, LLC.* DOI: https://arxiv.org/abs/arXiv:1711.04837.

Asness, C. *et al.* (2015) ‘Fact, Fiction, and Value Investing’ *The Journal of Portfolio Management*, 42(1), pp. 34-52. DOI: https://doi.org/10.3905/jpm.2015.42.1.034.

Chan, L. K. C., and Lakonishok, J. (2004) ‘Value and Growth Investing: Review and Update’ *Financial Analyst Journal*, 60(1), pp. 71-86. DOI: https://doi.org/10.2469/faj.v60.n1.2593.

Collobert, R. *et al*. (2011) ‘Natural Language Processing (Almost) from Scratch’ *Journal of Machine Learning Research*, 12(1), pp. 2493-2537. DOI: https://arxiv.org/abs/1103.0398.

Dawson, C. W. (2009) *Projects in Computing and Information Systems: Second Edition*. Harlow, United Kingdom: Pearson Education.

Fama, E. F., and French, K. R. (1992) ‘The Cross-Section of Expected Stock Returns’ *The Journal of Finance*, 47(2), pp. 427-465. DOI: https://doi.org/10.2307/2329112.

Graham, B. (1973) *The Intelligent Investor: Fourth Edition*. New York, United States: Harper and Row.

Graham, B. (1976) ‘A conversation with Benjamin Graham’ *Financial Analysts Journal*, 32(5), pp. 20-23. DOI: https://doi.org/10.2469/faj.v32.n5.20.

Graham, B., and Dodd, D. L. (2009) *Security Analysis: Sixth Edition*. New York, United States: McGraw-Hill Education.

Haykin, S.S. (2009) *Neural Networks and Learning Machines: Third Edition*. Upper Saddle River, New Jersey: Pearson Education.

Hirschberg, J., and Manning, C. D. (2015) ‘Advances in natural language processing’ *Science*, 349(6245), pp. 261-266. DOI: https://doi.org/10.1126/science.aaa8685.

Loughran, T., and McDonald, B. (2011) ‘When is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks’ *The Journal of Finance*, 66(1), pp. 35-65. DOI: https://doi.org/10.1111/j.1540-6261.2010.01625.x.

Mikolov, T. *et al*. (2011) ‘Extensions of Recurrent Neural Network Language Model’ *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5528-5531. DOI: https://doi.org/10.1109/ICASSP.2011.5947611.

Nasukawa, T., and Yi, J. (2003) ‘Sentiment Analysis: Capturing favorability using Natural Language Processing’ *K-CAP '03 Proceedings of the 2nd international conference on Knowledge capture*, pp. 70-77. DOI: https://doi.org/10.1145/945645.945658.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986) ‘Learning Representations by Back-Propagation Errors’ *Nature*, 323(6088), pp. 533-536. DOI: https://doi.org/10.1038/323533a0.

Pätäri, E. J. *et al*. (2018) ‘Enhancement of value investment strategies based on financial statement variables: the German evidence’ *Review of Quantitative Finance and Accounting*, 51(3), pp. 813-845. DOI: https://doi.org/10.1007/s11156-017-0689-y.

Piotroski, J. D. (2000) ‘Value Investing: The Use of Historical Financial Statement Information to Separate Winners from Losers’ *Journal of Accounting Research*, 38(1), pp. 1-41. DOI: https://doi.org/10.2307/2672906.

Research Ethics Review Form: BSc, MSc and MA Projects

Computer Science Research Ethics Committee (CSREC)

Undergraduate and postgraduate students undertaking their final project in the Department of Computer Science are required to consider the ethics of their project work and to ensure that it complies with research ethics guidelines. In some cases, a project will need approval from an ethics committee before it can proceed. Usually, but not always, this will be because the student is involving other people ("participants") in the project.

In order to ensure that appropriate consideration is given to ethical issues, all students must complete this form and attach it to their project proposal document. There are two parts:

PART A: Ethics Checklist. All students must complete this part. The checklist identifies whether the project requires ethical approval and, if so, where to apply for approval.

PART B: Ethics Proportionate Review Form. Students who have answered "no" to all questions in A1, A2 and A3 and "yes" to question 4 in A4 in the ethics checklist must complete this part. The project supervisor has delegated authority to provide approval in such cases that are considered to involve MINIMAL risk. The approval may be ***provisional*** – *identifying the planned research as* likely to involve MINIMAL RISK. In such cases you must additionally seek ***full approval*** from the supervisor as the project progresses and details are established. ***Full approval*** must be acquired in writing, before beginning the planned research.

A.1 If you answer YES to any of the questions in this block, you must apply to an appropriate external ethics committee for approval and log this approval as an External Application through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
1.1	Does your research require approval from the National Research Ethics Service (NRES)? *e.g. because you are recruiting current NHS patients or staff?* *If you are unsure try - https://www.hra.nhs.uk/approvals-amendments/what-approvals-do-i-need/*	**NO**
1.2	Will you recruit participants who fall under the auspices of the Mental Capacity Act? *Such research needs to be approved by an external ethics committee such as NRES or the Social Care Research Ethics Committee - http://www.scie.org.uk/research/ethics-committee/*	**NO**
1.3	Will you recruit any participants who are currently under the auspices of the Criminal Justice System, for example, but not limited to, people on remand, prisoners and those on probation? *Such research needs to be authorised by the ethics approval system of the National Offender Management Service.*	**NO**
A.2 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee, you must apply for approval from the Senate Research Ethics Committee (SREC) through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
2.1	Does your research involve participants who are unable to give informed consent? *For example, but not limited to, people who may have a degree of learning disability or mental health problem, that means they are unable to make an informed decision on their own behalf.*	**NO**
2.2	Is there a risk that your research might lead to disclosures from participants concerning their involvement in illegal activities?	**NO**
2.3	Is there a risk that obscene and or illegal material may need to be accessed for your research study (including online content and other material)?	**NO**

2.4	Does your project involve participants disclosing information about special category or sensitive subjects? *For example, but not limited to: racial or ethnic origin; political opinions; religious beliefs; trade union membership; physical or mental health; sexual life; criminal offences and proceedings*	NO
2.5	Does your research involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning that affects the area in which you will study? *Please check the latest guidance from the FCO - http://www.fco.gov.uk/en/*	NO
2.6	Does your research involve invasive or intrusive procedures? *These may include, but are not limited to, electrical stimulation, heat, cold or bruising.*	NO
2.7	Does your research involve animals?	NO
2.8	Does your research involve the administration of drugs, placebos or other substances to study participants?	NO
A.3 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee or the SREC, you must apply for approval from the Computer Science Research Ethics Committee (CSREC) through Research Ethics Online - https://ethics.city.ac.uk/ Depending on the level of risk associated with your application, it may be referred to the Senate Research Ethics Committee.		*Delete as appropriate*
3.1	Does your research involve participants who are under the age of 18?	NO
3.2	Does your research involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)? *This includes adults with cognitive and / or learning disabilities, adults with physical disabilities and older people.*	NO
3.3	Are participants recruited because they are staff or students of City, University of London? *For example, students studying on a particular course or module.* *If yes, then approval is also required from the Head of Department or Programme Director.*	NO
3.4	Does your research involve intentional deception of participants?	NO
3.5	Does your research involve participants taking part without their informed consent?	NO
3.5	Is the risk posed to participants greater than that in normal working life?	NO
3.7	Is the risk posed to you, the researcher(s), greater than that in normal working life?	NO
A.4 If you answer YES to the following question and your answers to all other questions in sections A1, A2 and A3 are NO, then your project is deemed to be of MINIMAL RISK. **If this is the case, then you can apply for approval through your supervisor under PROPORTIONATE REVIEW. You do so by completing PART B of this form.** **If you have answered NO to all questions on this form, then your project does not require ethical approval. You should submit and retain this form as evidence of this.**		*Delete as appropriate*
4	Does your project involve human participants or their identifiable personal data? *For example, as interviewees, respondents to a survey or participants in testing.*	NO

Appendix B - Piotroski's Methodology - Back-testing Results - Annualized Excess Returns

Portfolio	Mean	10th Percentile	25th Percentile	Median	75th Percentile	90th Percentile	n
All Companies	1013.853	−0.687	−0.412	−0.118	0.140	0.578	1453
F_SCORE = 0	−0.404	−0.404	−0.404	−0.404	−0.404	−0.404	1
F_SCORE = 1	−0.214	−0.815	−0.610	−0.447	0.029	0.527	7
F_SCORE = 2	0.623	−0.726	−0.405	−0.045	0.267	0.934	70
F_SCORE = 3	3.471	−0.730	−0.476	−0.118	0.317	1.016	184
F_SCORE = 4	4907.126	−0.765	−0.460	−0.131	0.160	0.673	300
F_SCORE = 5	0.894	−0.610	−0.365	−0.100	0.119	0.449	354
F_SCORE = 6	0.102	−0.646	−0.361	−0.094	0.136	0.375	285
F_SCORE = 7	−0.126	−0.684	−0.434	−0.188	0.064	0.408	175
F_SCORE = 8	−0.151	−0.602	−0.362	−0.160	0.085	0.282	70
F_SCORE = 9	0.001	−0.302	−0.156	0.099	0.156	0.216	9
High F_SCORE	−0.134	−0.609	−0.350	−0.149	0.100	0.282	79
Low F_SCORE	−0.237	−0.799	−0.541	−0.426	−0.053	0.444	8

Table 16: Piotroski's Methodology Returns - Long Position Annualized Excess Returns for High B/M Portfolios (to 3.d.p.)

Portfolio	Mean	10th Percentile	25th Percentile	Median	75th Percentile	90th Percentile	n
All Companies	0.935	−0.300	−0.105	0.103	0.568	1.681	1453
F_SCORE = 0	0.521	0.521	0.521	0.521	0.521	0.521	1
F_SCORE = 1	1.100	−0.246	0.019	0.635	1.596	3.147	7
F_SCORE = 2	0.872	−0.401	−0.185	0.041	0.496	1.859	70
F_SCORE = 3	1.214	−0.403	−0.194	0.092	0.617	2.134	184
F_SCORE = 4	1.366	−0.318	−0.117	0.122	0.736	2.357	300
F_SCORE = 5	0.689	−0.265	−0.093	0.089	0.450	1.286	354
F_SCORE = 6	0.727	−0.229	−0.099	0.078	0.501	1.405	285
F_SCORE = 7	1.024	−0.233	−0.043	0.186	0.627	1.776	175
F_SCORE = 8	0.389	−0.184	−0.067	0.178	0.477	1.177	70
F_SCORE = 9	0.089	−0.136	−0.097	−0.072	0.144	0.394	9
High F_SCORE	0.355	−0.184	−0.071	0.144	0.430	1.127	79
Low F_SCORE	1.028	−0.213	0.087	0.578	1.197	2.959	8

Table 17: Piotroski's Methodology Returns - Short Position Annualized Excess Returns for High B/M Portfolios (to 3.d.p.)

Appendix C - Grid Search Results for MLP Models

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
2	adam	relu	1.00000	0.01316	1.00000	0.00809	0.99887	0.01471
4	adam	relu	1.00000	0.01547	1.00000	0.01046	0.99887	0.01769
3	nadam	relu	0.99699	0.01584	1.00000	0.01070	0.99887	0.01681
2	nadam	relu	0.99749	0.01810	1.00000	0.01334	0.99887	0.02076
3	adam	relu	0.99850	0.01860	1.00000	0.01380	0.99887	0.02136
4	nadam	relu	0.99749	0.01930	1.00000	0.01447	0.99887	0.02213
1	nadam	relu	0.99900	0.01920	1.00000	0.01451	1.00000	0.02170
1	adam	relu	1.00000	0.01925	1.00000	0.01453	0.99887	0.02146
4	adamax	relu	0.99950	0.01951	1.00000	0.01476	1.00000	0.02177
3	adamax	relu	1.00000	0.02252	1.00000	0.01811	0.99887	0.02535
2	adamax	relu	0.99850	0.02753	1.00000	0.02348	0.99887	0.03103
1	adamax	relu	0.99699	0.03972	1.00000	0.03616	0.99775	0.04415
4	nadam	sigmoid	0.99649	0.04273	1.00000	0.03914	0.99550	0.04730
4	adam	sigmoid	0.99599	0.04275	0.99850	0.03917	0.99437	0.04726
3	adam	sigmoid	0.99599	0.04580	0.99850	0.04221	0.99324	0.05042
3	nadam	sigmoid	0.99399	0.04580	0.99850	0.04224	0.99437	0.05055
2	adam	sigmoid	0.99449	0.05140	0.99850	0.04786	0.99437	0.05647
2	nadam	sigmoid	0.99549	0.05156	0.99700	0.04797	0.99324	0.05643
1	adam	sigmoid	0.99299	0.06617	0.99700	0.06250	0.99324	0.07153
1	nadam	sigmoid	0.99248	0.06635	0.99700	0.06264	0.99324	0.07158
4	adamax	sigmoid	0.99048	0.08568	0.99399	0.08182	0.99212	0.09128
3	adamax	sigmoid	0.98998	0.09219	0.99099	0.08822	0.99212	0.09781
2	adamax	sigmoid	0.98998	0.10544	0.99099	0.10118	0.99099	0.11099
1	adamax	sigmoid	0.98898	0.14024	0.99099	0.13520	0.99099	0.14547
4	sgd	relu	0.98347	0.16416	0.99700	0.15895	0.99324	0.17025
3	sgd	relu	0.98046	0.17649	0.99700	0.17081	0.99324	0.18227
1	sgd	relu	0.98246	0.19447	0.99700	0.18848	0.99324	0.20035
2	sgd	relu	0.97896	0.19845	0.99700	0.19221	0.99324	0.20413
4	sgd	sigmoid	0.98647	0.33245	0.99099	0.32199	0.98761	0.33505
1	sgd	sigmoid	0.98347	0.35305	0.99099	0.34257	0.98761	0.35554
2	sgd	sigmoid	0.98397	0.36194	0.99099	0.35116	0.99212	0.36467
3	sgd	sigmoid	0.98998	0.36752	0.99099	0.35649	0.99099	0.37009

Table 18: F_ROA MLP model grid search results (to 5.d.p.)

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
4	nadam	relu	0.99800	0.00844	1.00000	0.00745	1.00000	0.00653
3	nadam	relu	0.99800	0.00881	1.00000	0.00793	1.00000	0.00697
4	adam	relu	0.99900	0.00989	1.00000	0.00900	1.00000	0.00799
2	nadam	relu	0.99800	0.01025	1.00000	0.00951	1.00000	0.00845
3	adam	relu	0.99850	0.01029	1.00000	0.00962	1.00000	0.00857
2	adam	relu	0.99800	0.01068	1.00000	0.01006	1.00000	0.00896
1	nadam	relu	0.99800	0.01190	1.00000	0.01151	1.00000	0.01027
1	adam	relu	0.99800	0.01193	1.00000	0.01154	1.00000	0.01029
4	adamax	relu	0.99800	0.01358	1.00000	0.01323	1.00000	0.01200
3	adamax	relu	0.99800	0.01571	1.00000	0.01568	1.00000	0.01426
2	adamax	relu	0.99800	0.01891	1.00000	0.01936	1.00000	0.01764
4	nadam	sigmoid	0.99800	0.01885	1.00000	0.01969	1.00000	0.01770
4	adam	sigmoid	0.99800	0.01885	1.00000	0.01972	1.00000	0.01770
3	adam	sigmoid	0.99850	0.02537	1.00000	0.02670	1.00000	0.02462
3	nadam	sigmoid	0.99850	0.02540	1.00000	0.02674	1.00000	0.02467
1	adamax	relu	0.99800	0.02676	1.00000	0.02822	1.00000	0.02595
2	adam	sigmoid	0.99850	0.02829	1.00000	0.02981	1.00000	0.02764
2	nadam	sigmoid	0.99850	0.02840	1.00000	0.02993	1.00000	0.02776
1	nadam	sigmoid	0.99900	0.03729	1.00000	0.03906	1.00000	0.03673
1	adam	sigmoid	0.99850	0.03736	1.00000	0.03911	1.00000	0.03678
4	adamax	sigmoid	0.99900	0.03817	1.00000	0.04022	1.00000	0.03753
3	adamax	sigmoid	0.99900	0.05604	0.99700	0.05724	0.99887	0.05554
2	adamax	sigmoid	0.99699	0.06490	0.99700	0.06572	0.99887	0.06404
1	sgd	relu	0.99699	0.08264	0.98949	0.08345	0.99099	0.08060
2	sgd	relu	0.99549	0.09115	0.98799	0.09122	0.99099	0.08850
1	adamax	sigmoid	0.99248	0.09600	0.99399	0.09535	0.99324	0.09377
3	sgd	relu	0.99098	0.10966	0.98498	0.10837	0.98649	0.10587
4	sgd	relu	0.98898	0.13035	0.97748	0.12763	0.97860	0.12529
4	sgd	sigmoid	0.69389	0.59575	0.66216	0.61399	0.70158	0.58874
3	sgd	sigmoid	0.69389	0.60944	0.66216	0.62939	0.70045	0.60371
1	sgd	sigmoid	0.69389	0.61050	0.66216	0.63054	0.70045	0.60487
2	sgd	sigmoid	0.69389	0.61100	0.66216	0.63075	0.70045	0.60542

Table 19: F_CFO MLP model grid search results (to 5.d.p.)

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
4	adamax	relu	0.99499	0.04241	0.99700	0.04774	0.98986	0.04760
3	adamax	relu	0.99349	0.04287	0.99850	0.04842	0.98986	0.04790
2	adamax	relu	0.99449	0.04448	0.99700	0.05026	0.98986	0.04988
4	adam	relu	0.99349	0.04680	0.99850	0.05316	0.99212	0.05190
1	nadam	relu	0.99148	0.05299	0.99550	0.05965	0.98536	0.05941
2	nadam	relu	0.99198	0.05449	0.99700	0.06120	0.98874	0.06043
1	adam	relu	0.99148	0.05537	0.99550	0.06219	0.98536	0.06187
4	nadam	relu	0.99299	0.05593	0.99850	0.06266	0.98986	0.06162
3	adam	relu	0.98798	0.05826	0.99700	0.06506	0.98874	0.06436
2	adam	relu	0.99048	0.05893	0.99700	0.06562	0.98536	0.06530
3	nadam	relu	0.98898	0.05863	0.99700	0.06569	0.98986	0.06487
1	adamax	relu	0.98998	0.06513	0.99249	0.07225	0.98423	0.07301
4	adam	sigmoid	0.98747	0.07068	0.98949	0.07899	0.98423	0.07956
4	nadam	sigmoid	0.98798	0.07078	0.98949	0.07917	0.98423	0.07974
3	adam	sigmoid	0.98697	0.07538	0.98649	0.08383	0.98198	0.08506
3	nadam	sigmoid	0.98547	0.07552	0.98649	0.08392	0.98198	0.08518
2	adam	sigmoid	0.98246	0.08375	0.98498	0.09246	0.97748	0.09410
2	nadam	sigmoid	0.98397	0.08384	0.98498	0.09256	0.97748	0.09429
1	nadam	sigmoid	0.98096	0.10411	0.97898	0.11316	0.97410	0.11603
1	adam	sigmoid	0.98046	0.10441	0.98048	0.11347	0.97748	0.11609
4	adamax	sigmoid	0.97295	0.13026	0.97598	0.13881	0.96847	0.14471
3	adamax	sigmoid	0.96994	0.14021	0.97598	0.14849	0.96396	0.15543
2	adamax	sigmoid	0.96794	0.15741	0.97447	0.16512	0.96284	0.17348
1	adamax	sigmoid	0.95792	0.19638	0.96096	0.20263	0.95045	0.21395
3	sgd	relu	0.94840	0.22254	0.97598	0.22917	0.96396	0.23918
4	sgd	relu	0.94339	0.24629	0.96847	0.25138	0.95608	0.26391
2	sgd	relu	0.94038	0.25669	0.95946	0.26051	0.95045	0.27421
1	sgd	relu	0.93637	0.26115	0.95796	0.26481	0.95045	0.27873
4	sgd	sigmoid	0.88126	0.38683	0.85886	0.38380	0.85135	0.41025
1	sgd	sigmoid	0.86924	0.40508	0.85285	0.40168	0.84459	0.42759
2	sgd	sigmoid	0.85671	0.41095	0.84835	0.40689	0.84009	0.43409
3	sgd	sigmoid	0.85371	0.41531	0.84384	0.41090	0.82095	0.43878

Table 20: F_CROA MLP model grid search results (to 5.d.p.)

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
4	adam	sigmoid	0.99749	0.03284	1.00000	0.03054	0.99662	0.03370
4	nadam	sigmoid	0.99649	0.03294	1.00000	0.03058	0.99662	0.03373
3	nadam	sigmoid	0.99599	0.03588	0.99850	0.03308	0.99550	0.03672
3	adam	sigmoid	0.99599	0.03606	0.99850	0.03320	0.99437	0.03694
4	adamax	sigmoid	0.99749	0.06987	1.00000	0.06238	0.99662	0.06976
3	adamax	sigmoid	0.99900	0.10176	0.99249	0.09262	0.99662	0.10151
2	adamax	sigmoid	0.99549	0.11495	0.98799	0.10530	0.99662	0.11472
2	nadam	sigmoid	0.75100	0.56080	0.72222	0.59019	0.74099	0.57098
2	adam	sigmoid	0.75100	0.56069	0.72222	0.59020	0.74099	0.57094
1	adamax	sigmoid	0.75100	0.56212	0.72222	0.59065	0.74099	0.57197
1	nadam	sigmoid	0.75100	0.56298	0.72222	0.59090	0.74099	0.57254
1	adam	sigmoid	0.75100	0.56292	0.72222	0.59092	0.74099	0.57248
2	adam	relu	0.75050	0.56293	0.72222	0.59208	0.74099	0.57265
2	nadam	relu	0.75050	0.56338	0.72222	0.59213	0.73986	0.57285
3	adam	relu	0.75050	0.56350	0.72072	0.59223	0.73986	0.57290
3	nadam	relu	0.75050	0.56324	0.72072	0.59245	0.73986	0.57284
1	adam	relu	0.75050	0.56381	0.72072	0.59280	0.73986	0.57321
1	nadam	relu	0.75050	0.56315	0.72072	0.59287	0.73986	0.57294
2	sgd	sigmoid	0.75100	0.56305	0.72222	0.59295	0.74099	0.57315
4	sgd	relu	0.75100	0.56137	0.72222	0.59300	0.74099	0.57230
3	sgd	relu	0.75100	0.56146	0.72222	0.59313	0.74099	0.57239
2	sgd	relu	0.75100	0.56159	0.72222	0.59317	0.74099	0.57246
3	sgd	sigmoid	0.75100	0.56300	0.72222	0.59318	0.74099	0.57318
1	sgd	relu	0.75100	0.56167	0.72222	0.59320	0.74099	0.57250
4	nadam	relu	0.75050	0.56201	0.72222	0.59347	0.74099	0.57270
4	adam	relu	0.75050	0.56258	0.72222	0.59369	0.73986	0.57298
1	sgd	sigmoid	0.75100	0.56439	0.72222	0.59381	0.74099	0.57425
3	adamax	relu	0.75050	0.56339	0.72072	0.59411	0.73986	0.57342
2	adamax	relu	0.74950	0.56429	0.72072	0.59474	0.73986	0.57403
1	adamax	relu	0.74900	0.56514	0.72072	0.59515	0.73986	0.57457
4	adamax	relu	0.74950	0.56565	0.72072	0.59552	0.73986	0.57489
4	sgd	sigmoid	0.75100	0.56926	0.72222	0.60184	0.74099	0.57993

Table 21: F_ACCRUAL MLP model grid search results (to 5.d.p.)

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
2	nadam	sigmoid	0.98848	0.07994	0.99399	0.05905	0.98761	0.07555
4	adam	sigmoid	0.98948	0.08027	0.99700	0.05947	0.99099	0.07569
3	nadam	sigmoid	0.98848	0.08065	0.99700	0.05980	0.99099	0.07599
2	adam	sigmoid	0.98848	0.08122	0.99550	0.06036	0.98874	0.07662
3	adam	sigmoid	0.98747	0.08341	0.99550	0.06208	0.98874	0.07848
4	nadam	sigmoid	0.98898	0.08599	0.99550	0.06422	0.98874	0.08074
1	adam	sigmoid	0.98948	0.09451	0.99550	0.07197	0.98874	0.08829
1	nadam	sigmoid	0.98597	0.09468	0.99399	0.07208	0.98874	0.08846
4	adamax	sigmoid	0.98547	0.11888	0.99249	0.09303	0.98761	0.11026
3	adamax	sigmoid	0.98547	0.12251	0.99249	0.09638	0.98536	0.11352
2	adamax	sigmoid	0.98397	0.13537	0.99249	0.10849	0.98536	0.12516
1	adamax	sigmoid	0.98347	0.16689	0.99249	0.13882	0.98536	0.15457
4	sgd	sigmoid	0.94489	0.36223	0.96697	0.33619	0.93806	0.34923
3	sgd	sigmoid	0.93637	0.38895	0.95495	0.36444	0.93243	0.37717
2	sgd	sigmoid	0.92585	0.40762	0.95045	0.38424	0.92793	0.39667
1	sgd	sigmoid	0.89228	0.45538	0.93544	0.43520	0.90428	0.44631
4	nadam	relu	0.54910	0.68836	0.56006	0.68618	0.55743	0.68669
3	nadam	relu	0.54910	0.68837	0.56006	0.68619	0.55743	0.68670
2	nadam	relu	0.54910	0.68837	0.56006	0.68619	0.55743	0.68670
3	adam	relu	0.54910	0.68839	0.56006	0.68623	0.55743	0.68669
3	adamax	relu	0.54910	0.68839	0.56006	0.68624	0.55743	0.68670
2	adamax	relu	0.54910	0.68843	0.56006	0.68626	0.55743	0.68669
3	sgd	relu	0.54910	0.68849	0.56006	0.68628	0.55743	0.68668
2	adam	relu	0.54910	0.68846	0.56006	0.68629	0.55743	0.68673
1	adam	relu	0.54910	0.68849	0.56006	0.68631	0.55743	0.68670
1	nadam	relu	0.54910	0.68847	0.56006	0.68632	0.55743	0.68671
1	adamax	relu	0.54910	0.68852	0.56006	0.68639	0.55743	0.68670
2	sgd	relu	0.54910	0.68857	0.56006	0.68639	0.55743	0.68668
4	adam	relu	0.54910	0.68856	0.56006	0.68639	0.55743	0.68671
4	adamax	relu	0.54910	0.68859	0.56006	0.68641	0.55743	0.68670
4	sgd	relu	0.54910	0.68865	0.56006	0.68645	0.55743	0.68671
1	sgd	relu	0.54910	0.68861	0.56006	0.68646	0.55743	0.68669

Table 22: F_CLEVER MLP model grid search results (to 5.d.p.)

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
2	adam	relu	0.99950	0.00600	1.00000	0.00455	1.00000	0.00671
3	adam	relu	0.99950	0.00595	1.00000	0.00460	1.00000	0.00667
2	nadam	relu	0.99950	0.00611	1.00000	0.00463	1.00000	0.00683
1	nadam	relu	0.99950	0.00619	1.00000	0.00464	1.00000	0.00700
3	nadam	relu	0.99950	0.00610	1.00000	0.00467	1.00000	0.00681
4	nadam	relu	0.99950	0.00639	1.00000	0.00494	1.00000	0.00710
4	adam	relu	0.99950	0.00665	1.00000	0.00513	1.00000	0.00737
1	adam	relu	0.99950	0.00689	1.00000	0.00514	1.00000	0.00781
4	adamax	relu	0.99950	0.00762	1.00000	0.00593	1.00000	0.00857
3	adamax	relu	0.99950	0.00856	1.00000	0.00665	1.00000	0.00967
4	adam	sigmoid	0.99950	0.00928	1.00000	0.00677	1.00000	0.01047
4	nadam	sigmoid	0.99950	0.00935	1.00000	0.00681	0.99887	0.01051
3	adam	sigmoid	0.99950	0.00991	1.00000	0.00729	1.00000	0.01112
3	nadam	sigmoid	0.99950	0.00991	1.00000	0.00730	0.99887	0.01113
2	adamax	relu	0.99950	0.01039	1.00000	0.00811	1.00000	0.01168
2	nadam	sigmoid	0.99950	0.01085	1.00000	0.00812	0.99887	0.01212
2	adam	sigmoid	0.99950	0.01087	1.00000	0.00813	1.00000	0.01214
1	adam	sigmoid	0.99950	0.01321	1.00000	0.01016	1.00000	0.01447
1	nadam	sigmoid	0.99950	0.01327	1.00000	0.01020	1.00000	0.01453
1	adamax	relu	0.99950	0.01442	1.00000	0.01143	1.00000	0.01579
4	adamax	sigmoid	0.99950	0.02001	1.00000	0.01580	1.00000	0.02101
3	adamax	sigmoid	0.99950	0.02108	1.00000	0.01672	1.00000	0.02201
2	adamax	sigmoid	0.99950	0.02365	1.00000	0.01895	1.00000	0.02440
1	adamax	sigmoid	0.99950	0.02958	1.00000	0.02426	1.00000	0.03000
3	sgd	relu	0.99749	0.03783	0.99850	0.03051	1.00000	0.03714
2	sgd	relu	0.99850	0.03794	0.99700	0.03055	1.00000	0.03725
1	sgd	relu	0.99699	0.04107	0.99700	0.03324	1.00000	0.04009
4	sgd	relu	0.99699	0.04138	0.99700	0.03346	1.00000	0.04037
2	sgd	sigmoid	0.99449	0.08333	0.99700	0.06953	0.99212	0.07878
1	sgd	sigmoid	0.99499	0.08347	0.99700	0.07053	0.99212	0.07940
4	sgd	sigmoid	0.99449	0.08549	0.99700	0.07117	0.99212	0.08067
3	sgd	sigmoid	0.99449	0.08549	0.99700	0.07119	0.99212	0.08066

Table 23: F_CLIQUID MLP model grid search results (to 5.d.p.)

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
3	adam	sigmoid	0.99499	0.02383	0.99399	0.01812	0.99550	0.02045
4	nadam	sigmoid	0.99549	0.02421	0.99399	0.01835	0.99550	0.02075
1	nadam	sigmoid	0.99499	0.02432	0.99399	0.01892	0.99550	0.02099
1	adam	sigmoid	0.99499	0.02441	0.99399	0.01902	0.99550	0.02110
2	nadam	sigmoid	0.99549	0.02545	0.99399	0.01933	0.99550	0.02194
2	adam	sigmoid	0.99499	0.02561	0.99399	0.01943	0.99550	0.02209
4	adam	relu	0.99599	0.02629	0.99399	0.01960	0.99550	0.02265
3	adam	relu	0.99699	0.02712	0.99550	0.02004	0.99775	0.02316
4	adam	sigmoid	0.99599	0.02713	0.99399	0.02023	0.99550	0.02338
3	nadam	sigmoid	0.99499	0.02714	0.99399	0.02033	0.99550	0.02341
4	adamax	sigmoid	0.99499	0.02753	0.99399	0.02046	0.99550	0.02407
3	nadam	relu	0.99649	0.02845	0.99550	0.02085	0.99775	0.02444
3	adamax	relu	0.99499	0.02703	0.99399	0.02088	0.99550	0.02341
3	adamax	sigmoid	0.99499	0.02905	0.99399	0.02205	0.99550	0.02570
2	adamax	sigmoid	0.99499	0.03100	0.99399	0.02363	0.99550	0.02753
1	adamax	sigmoid	0.99800	0.03724	0.99700	0.02902	0.99775	0.03307
2	sgd	relu	0.99950	0.04164	0.99850	0.03043	1.00000	0.03723
3	sgd	relu	0.99950	0.04164	0.99850	0.03044	1.00000	0.03724
1	sgd	relu	1.00000	0.04164	0.99850	0.03044	1.00000	0.03724
4	sgd	relu	0.99950	0.04164	0.99850	0.03045	1.00000	0.03728
4	sgd	sigmoid	0.97244	0.06155	0.98348	0.04425	0.98423	0.05546
3	sgd	sigmoid	0.97295	0.06146	0.98348	0.04452	0.98423	0.05547
2	sgd	sigmoid	0.97345	0.06138	0.98348	0.04517	0.98423	0.05565
1	sgd	sigmoid	0.97495	0.06480	0.98498	0.04984	0.98536	0.05942
4	adamax	relu	0.62425	0.66207	0.61712	0.66545	0.64640	0.65151
1	adamax	relu	0.62425	0.66206	0.61712	0.66546	0.64640	0.65143
2	adam	relu	0.62425	0.66207	0.61712	0.66546	0.64640	0.65140
1	adam	relu	0.62425	0.66210	0.61712	0.66546	0.64640	0.65138
2	nadam	relu	0.62425	0.66208	0.61712	0.66546	0.64640	0.65138
1	nadam	relu	0.62425	0.66205	0.61712	0.66548	0.64640	0.65126
2	adamax	relu	0.62425	0.66205	0.61712	0.66548	0.64640	0.65138
4	nadam	relu	0.62425	0.66206	0.61712	0.66549	0.64640	0.65129

Table 24: F_EQ_OFFER MLP model grid search results (to 5.d.p.)

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
4	adamax	relu	0.99599	0.02960	0.99249	0.03163	0.99662	0.02912
2	adam	relu	0.99299	0.03968	0.99099	0.03990	0.99550	0.03839
1	nadam	relu	0.99299	0.04449	0.98949	0.04411	0.99437	0.04316
2	adamax	relu	0.99399	0.04543	0.98799	0.04514	0.99099	0.04495
3	adamax	relu	0.99449	0.04578	0.98949	0.04568	0.99324	0.04496
2	nadam	relu	0.99098	0.04813	0.98799	0.04741	0.99324	0.04680
3	nadam	relu	0.99349	0.04807	0.98949	0.04767	0.99550	0.04662
1	adam	relu	0.99148	0.04982	0.98799	0.04868	0.99099	0.04857
4	nadam	relu	0.99098	0.05068	0.98949	0.05001	0.99324	0.04938
4	adam	relu	0.98898	0.05416	0.98949	0.05313	0.99437	0.05254
3	adam	relu	0.98798	0.05799	0.98799	0.05611	0.99212	0.05596
4	nadam	sigmoid	0.98898	0.06213	0.98799	0.05860	0.98649	0.06097
3	nadam	sigmoid	0.98647	0.06586	0.98649	0.06189	0.98423	0.06489
1	adamax	relu	0.98848	0.06811	0.98799	0.06487	0.98649	0.06703
2	adam	sigmoid	0.98547	0.07161	0.98498	0.06696	0.98198	0.07059
2	nadam	sigmoid	0.98647	0.07169	0.98649	0.06704	0.98423	0.07051
3	adam	sigmoid	0.98747	0.07234	0.98649	0.06771	0.98423	0.07084
4	adam	sigmoid	0.98647	0.07285	0.98649	0.06815	0.98423	0.07144
1	nadam	sigmoid	0.98246	0.08455	0.98348	0.07855	0.97973	0.08321
1	adam	sigmoid	0.98397	0.08452	0.98348	0.07868	0.98086	0.08307
4	adamax	sigmoid	0.97094	0.10731	0.97598	0.09932	0.96509	0.10700
3	adamax	sigmoid	0.96994	0.11253	0.97297	0.10423	0.96284	0.11218
2	adamax	sigmoid	0.96192	0.12150	0.96997	0.11279	0.95946	0.12127
1	adamax	sigmoid	0.95892	0.14249	0.96547	0.13313	0.95045	0.14259
4	sgd	relu	0.92335	0.19674	0.94444	0.18639	0.92342	0.19743
1	sgd	relu	0.91232	0.20696	0.93694	0.19617	0.91892	0.20803
3	sgd	relu	0.91132	0.20798	0.93844	0.19753	0.92117	0.20890
2	sgd	relu	0.91082	0.21165	0.93544	0.20083	0.91554	0.21285
1	sgd	sigmoid	0.90030	0.23726	0.91892	0.22331	0.88176	0.23839
2	sgd	sigmoid	0.89329	0.23948	0.90991	0.22451	0.87162	0.24051
4	sgd	sigmoid	0.89028	0.24185	0.90841	0.22639	0.87162	0.24284
3	sgd	sigmoid	0.88928	0.24320	0.90841	0.22774	0.87162	0.24412

Table 25: F_CMARGIN MLP model grid search results (to 5.d.p.)

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
4	adamax	relu	0.99850	0.01511	0.99850	0.01518	0.99662	0.02025
2	nadam	relu	0.99900	0.01570	0.99850	0.01565	0.99662	0.02135
1	nadam	relu	0.99900	0.01626	0.99850	0.01612	0.99662	0.02199
3	nadam	relu	0.99950	0.01657	0.99850	0.01646	0.99662	0.02225
1	adam	relu	0.99850	0.01664	0.99850	0.01649	0.99662	0.02254
2	adam	relu	0.99850	0.01672	0.99850	0.01658	0.99662	0.02252
3	adamax	relu	0.99850	0.01779	0.99850	0.01768	0.99662	0.02354
4	adam	relu	0.99900	0.01908	0.99850	0.01891	0.99662	0.02547
3	adam	relu	0.99950	0.01939	0.99700	0.01917	0.99550	0.02599
2	adamax	relu	0.99850	0.02201	0.99850	0.02159	0.99662	0.02887
4	nadam	relu	0.99800	0.02319	0.99700	0.02272	0.99324	0.03063
4	nadam	sigmoid	0.99749	0.02769	0.99850	0.02637	0.99662	0.03641
4	adam	sigmoid	0.99749	0.02772	0.99700	0.02640	0.99550	0.03647
3	adam	sigmoid	0.99749	0.02957	0.99850	0.02805	0.99662	0.03850
3	nadam	sigmoid	0.99749	0.02965	0.99850	0.02815	0.99662	0.03864
2	adam	sigmoid	0.99749	0.03267	0.99700	0.03099	0.99550	0.04227
2	nadam	sigmoid	0.99749	0.03280	0.99700	0.03108	0.99324	0.04250
1	adamax	relu	0.99749	0.03283	0.99850	0.03157	0.99662	0.04209
1	nadam	sigmoid	0.99749	0.04001	0.99700	0.03787	0.99324	0.05068
1	adam	sigmoid	0.99749	0.04008	0.99700	0.03794	0.99324	0.05080
4	adamax	sigmoid	0.99599	0.05359	0.99550	0.05076	0.99099	0.06636
3	adamax	sigmoid	0.99599	0.05627	0.99550	0.05329	0.99099	0.06920
2	adamax	sigmoid	0.99599	0.06238	0.99550	0.05920	0.98986	0.07579
4	sgd	relu	0.99549	0.07528	0.98949	0.07162	0.98761	0.09023
2	sgd	relu	0.99399	0.07578	0.99249	0.07187	0.98986	0.08988
1	adamax	sigmoid	0.99449	0.07711	0.99249	0.07348	0.98986	0.09116
3	sgd	relu	0.98948	0.08536	0.99700	0.08167	0.99550	0.09964
1	sgd	relu	0.99248	0.09363	0.98348	0.08959	0.97523	0.11025
1	sgd	sigmoid	0.97445	0.16229	0.97297	0.15575	0.95158	0.17795
2	sgd	sigmoid	0.96743	0.16443	0.97147	0.15750	0.94820	0.18003
4	sgd	sigmoid	0.96794	0.16636	0.96396	0.15932	0.94707	0.18201
3	sgd	sigmoid	0.96493	0.16816	0.96396	0.16105	0.94595	0.18379

Table 26: F_CTURN MLP model grid search results (to 5.d.p.)

Number_Of_Hidden_Neurons	Optimizer	Hidden_Activation_Functions	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
27	adam	sigmoid	0.99950	0.01197	1.00000	0.01460	1.00000	0.01178
27	nadam	sigmoid	0.99950	0.01535	1.00000	0.01832	1.00000	0.01484
18	nadam	sigmoid	0.99900	0.01369	0.99700	0.01973	0.99887	0.01376
18	adam	sigmoid	0.99900	0.02170	0.99700	0.02810	0.99887	0.02108
9	nadam	relu	0.99950	0.02297	0.99700	0.03655	0.99775	0.02563
9	adam	relu	0.99950	0.02547	0.99700	0.04092	0.99775	0.02836
36	nadam	sigmoid	0.99900	0.03831	0.99700	0.04094	0.99887	0.03484
9	nadam	sigmoid	0.99900	0.04232	0.99700	0.04530	0.99887	0.03755
9	adam	sigmoid	0.99900	0.04262	0.99700	0.04949	0.99887	0.03922
27	adam	relu	0.99900	0.03824	0.99550	0.05066	0.99324	0.05145
27	adamax	relu	0.99900	0.04288	0.99249	0.06256	0.99437	0.05485
36	adam	sigmoid	0.99900	0.06539	0.99099	0.06704	0.99887	0.06071
18	adamax	relu	0.99900	0.05799	0.99700	0.07295	0.99775	0.05980
27	nadam	relu	0.99850	0.06370	0.98649	0.07924	0.99212	0.07651
36	nadam	relu	0.99749	0.04830	0.98498	0.07966	0.99212	0.06054
36	adam	relu	0.99599	0.04993	0.98348	0.08126	0.99099	0.06008
18	adam	relu	0.99950	0.07048	0.99700	0.09037	0.99775	0.07489
36	adamax	relu	0.99699	0.07544	0.98498	0.09469	0.99212	0.07991
18	nadam	relu	0.99749	0.07033	0.98649	0.09472	0.99324	0.07473
9	adamax	relu	0.99599	0.14334	0.99249	0.15457	0.99662	0.14847
36	adamax	sigmoid	0.99900	0.16518	0.99099	0.16362	0.99887	0.16052
27	adamax	sigmoid	0.99649	0.20795	0.98649	0.20653	0.99437	0.20519
18	adamax	sigmoid	0.98848	0.29533	0.98649	0.29210	0.99212	0.29274
9	adamax	sigmoid	0.95892	0.44935	0.94595	0.45045	0.96396	0.43789
27	sgd	relu	0.90080	0.64175	0.89640	0.65596	0.88401	0.64981
18	sgd	relu	0.88026	0.65646	0.87988	0.66493	0.90878	0.64760
36	sgd	relu	0.87475	0.65047	0.87688	0.66588	0.87613	0.65195
36	sgd	sigmoid	0.86172	0.83813	0.86937	0.83195	0.86599	0.82880
27	sgd	sigmoid	0.85271	0.85528	0.86486	0.84850	0.85135	0.84479
9	sgd	relu	0.82365	0.89685	0.84685	0.87540	0.84347	0.87718
18	sgd	sigmoid	0.86824	0.89530	0.86637	0.88144	0.88063	0.88138
9	sgd	sigmoid	0.85521	0.97082	0.84835	0.95725	0.87387	0.96118

Table 27: F_SCORE MLP model grid search results (to 5.d.p.)

Appendix D - Loughran and Mcdonald Sentiment Labeling - Back-testing Results - Annualized Excess Returns

Portfolio	Mean	10th Percentile	25th Percentile	Median	75th Percentile	90th Percentile	n
All Companies	1722.497	-0.686	-0.399	-0.106	0.168	0.598	855
Positive Sentiment	-0.140	-0.701	-0.433	-0.106	0.061	0.380	38
Negative Sentiment	1802.619	-0.685	-0.399	-0.106	0.169	0.600	817
High F_SCORE with Positive Sentiment	0.105	-0.171	-0.067	0.105	0.278	0.235	2
High F_SCORE & F_SCORE = 7 with Positive Sentiment	-0.135	-0.552	-0.221	-0.110	0.034	0.258	6

Table 28: Loughran and Mcdonald Returns - Long Position Annualized Excess Returns for High B/M Portfolios (to 3.d.p.)

Portfolio	Mean	10th Percentile	25th Percentile	Median	75th Percentile	90th Percentile	n
All Companies	0.984	-0.310	-0.121	0.090	0.541	1.661	855
Positive Sentiment	0.556	-0.223	-0.045	0.097	0.614	1.757	38
Negative Sentiment	1.004	-0.311	-0.125	0.088	0.540	1.632	817
Low F_SCORE with Negative Sentiment	0.754	-0.044	0.000	0.077	0.606	2.582	10
High F_SCORE & F_SCORE = 2 with Negative Sentiment	0.615	-0.365	-0.163	0.052	0.515	1.772	54

Table 29: Loughran and Mcdonald Returns - Short Position Annualized Excess Returns for High B/M Portfolios (to 3.d.p.)

Appendix E - Long Position Annualized Holding Period Returns-based Sentiment Labeling - Back-testing Results - Annualized Excess Returns

Portfolio	Mean	10th Percentile	25th Percentile	Median	75th Percentile	90th Percentile	n
All Companies	1722.497	-0.686	-0.399	-0.106	0.168	0.598	855
Positive Sentiment	3449.468	-0.082	0.014	0.169	0.451	1.086	427
Negative Sentiment	-0.439	-0.813	-0.617	-0.399	-0.220	-0.131	428
High F_SCORE with Positive Sentiment	0.179	-0.032	0.069	0.145	0.255	0.446	20

Table 30: Long Position Annualized Holding Period Returns for High B/M Portfolios (to 3.d.p.)

Portfolio	Mean	10th Percentile	25th Percentile	Median	75th Percentile	90th Percentile	n
All Companies	0.984	-0.310	-0.121	0.090	0.541	1.661	855
Positive Sentiment	-0.160	-0.443	-0.262	-0.120	-0.012	0.069	427
Negative Sentiment	2.126	0.136	0.256	0.541	1.190	2.752	428
Low F_SCORE with Negative Sentiment	0.878	0.000	0.000	0.337	0.757	2.582	10

Table 31: Short Position Annualized Holding Period Returns for High B/M Portfolios (to 3.d.p.)

Appendix F - Grid Search Results for LSTM Model

Number_Of_Hidden_Neurons	Optimizer	Training_Accuracy	Training_Loss	Validation_Accuracy	Validation_Loss	Test_Accuracy	Test_Loss
27	sgd	0.52083	0.69259	0.49068	0.69312	0.46262	0.69506
18	sgd	0.51667	0.69317	0.49068	0.69392	0.46262	0.69569
27	adamax	0.55625	0.67296	0.49689	0.69460	0.44860	0.69851
18	adamax	0.61875	0.60167	0.50311	0.70205	0.46262	0.72834
18	adam	0.90208	0.42183	0.47205	0.74698	0.50467	0.74747
18	nadam	0.91458	0.39486	0.47826	0.77719	0.51402	0.74417
27	nadam	0.97083	0.20288	0.45342	0.96088	0.51869	0.89896
27	adam	0.97708	0.14607	0.44099	1.24915	0.53738	1.06991

Table 32: Sentiment analysis LSTM model grid search results (to 5.d.p.)