# Long-term Stock Selection using Random Forest and LSTM Models for Fundamental Analysis

Morgan Wynne
Supervised by: Ian Nabney

August 29, 2023

**Abstract**

Stock selection for longer term investment horizons is a notoriously challenging task that involves many interrelated factors. Fundamental analysis is commonly used by investors of various types to identify equities that are most likely to outperform their relevant benchmark. Investors are increasingly turning to machine learning methods to model the complex, nonlinear relationships between fundamental indicators and thus find the optimal solution to this problem at the time of investment. In contemporary literature, Random Forest models have demonstrated consistently superior performance on such tasks. Although sequence prediction models have not yet been applied to such tasks, it is hypothesised that incorporating temporal information to model inputs may improve model performance.

This thesis compares the performance of a Long-Short Term Memory (LSTM) model, the outstanding sequence prediction model for short-term stock movement prediction, with the performance of a Random Forest model, for selecting stocks from the entire universe of North American publicly listed companies. The models use 50 years of fundamental data on North American stocks, including company financial ratios and macro-economic indicators to rank stocks by their predicted returns over the subsequent year. The long-term returns generated by a simple buy-and-hold strategy that uses each model's predictions is compared against the S&P 500 benchmark index and the model selections are analysed in detail to ensure validity and applicability to real-world scenarios. Evaluating the models in this way reveals useful insights about the "type" of stocks that each model selects and the significant effect that factors like market capitalisation and diversification have on model performance.

For an investment universe constituting all North American listed stocks, the Random Forest and LSTM models generate impressive returns, outperforming the S&P 500 value-weighted index by 151.50% and 49.44% per year on average. When restricted to the more competitive universe of Mid Cap stocks and larger, both models continue to outperform the benchmark, achieving average excess returns of 9.91% and 10.28% pa respectively.

**Ethics statement:** This project does not require ethics approval, as reviewed by my supervisor Ian Nabney. I have completed the ethics test on Blackboard. My score is 12/12.

**Github code repository:** `https://github.com/mwynne-bristol/dsp-vq22301`
**Onedrive data repository:** here

# Contents

# 1 Introduction

Stocks are one of several major asset classes that buy-side investors use to generate returns on invested capital. They are popular amongst both ***institutional investors*** and ***retail investors*** for their accessibility and potential to generate outsized returns. Active investment, or stock selection, is the act of buying and selling stocks with the aim of outperforming the market. The Efficient Market Hypothesis (EMH), proposed by Eugene Fama (1970), suggests that financial markets assimilate all relevant information, rendering it challenging to consistently attain above-average returns through active investing [1]. Opposing theories, such as that proposed by Robert Shiller (1980), hypothesise that markets are run by people and therefore are governed by human factors [2]. Thus, they are inefficient to the extent that opportunities exist for active investors to generate alpha, the excess return beyond market movements and ***idiosyncratic returns***.

Lass Heje Pedersen (2015) proposed that markets are in fact "efficiently inefficient". Markets must be inefficient enough that active investors are compensated for their costs, but efficient enough to discourage additional active investing [3]. This hypothesis is supported by the evidence that although the majority of active investors fail to beat the market, prominent investors such as Warren Buffett have been able to consistently beat the market over a long period of time. Pedersen further argues that the alpha that active investors can generate come from one of two sources: 1) The reward for taking on ***liquidity risk***, or 2) the reward for having an information advantage over other investors.

Two major approaches exist for the generation of information advantage - fundamental and technical analysis. Technical analysis has a shorter term outlook and assumes that the market price is fully discounted by the stock's fundamentals and movements are governed by supply and demand dynamics [4]. Consequently, previous market behaviour repeats itself and therefore stock price movement and volume information can be used to predict stock price movements in the near future [5]. Fund flow, trend, momentum, volatility, and raw price data are all examples of technical indicators.

Fundamental analysis, on the other hand, is based on the assumption that the value of a stock is determined by its potential profitability [5]. Fundamentalists believe that although ***investor sentiment*** can drive short-term market fluctuations, in the long run, the "weighing machine" causes the company's real value and market price to converge [6]. Fundamental analysis forms the basis of many longer-term investment strategies and is adopted by renowned investors, including Benjamin Graham and Warren Buffett. It includes:

1. macroeconomic analysis, which incorporates factors such as Gross Domestic Product (GDP), inflation, interest rates, etc.,
2. industry analysis, which analyses value by comparing the relative return performance of other stocks in an industry, and
3. company analysis, which analyses various financial ratios that indicate a company's current financial status.

The dataset in this thesis uses the first and third elements. It incorporates monthly data on 70 financial ratios for more than 23,000 unique North American stocks over the period 1970-2022. This data is combined with macroeconomic data describing GDP growth rates, inflation rates, the federal funds rate, and the yield on 10-year government bonds, a recognised proxy for economic confidence.

In Shah et al.'s (2019) taxonomy for stock market prediction techniques, advancements fall under five categories: statistical, pattern recognition, machine learning, sentiment analysis, and hybrid methods [6]. In recent years, machine learning and hybrid methods have dominated the literature. Unlike traditional statistical methods, machine learning can model non-linear functions between input variables that more accurately represent the complex mechanisms of the stock market. Furthermore, statistical methods, e.g. Autoregressive Integrated Moving Average (ARIMA) and Autoregressive Moving Average (ARMA), require strict assumptions regarding the distributions and stationarity of time-series [7]. Machine learning methods, on the other hand, are less affected by such characteristics and evidence suggests that the impact of the non-stationarity on prediction power is negligible.

The majority of contemporary literature that utilise machine learning for stock price prediction focus on short-term ***investment horizons*** of 30 days or less and use technical indicators to model price movements. When applied to active trading scenarios, accurately forecasting short-term stock prices improves performance

on a mulititude of tasks, including High Frequency Trading (HFT) and market impact optimisation. ***Sequence prediction models*** that account for the temporal dimension inherent in stock market time series, such as the classic Recurrent Neural Network (RNN) and its adaptations, are commonly applied to these tasks and achieve consistently impressive performance [7, 8, 9, 10]. Long Short-Term Memory models (LSTM) are a popular variant of the RNN model that consistently outperform other sequence prediction models for short-term stock movement prediction. LSTM is an RNN that uses gates and memory cells to solve the vanishing gradient problem associated with RNNs and capture more complex temporal patterns [8].

Long-term investors, like Warren Buffett, on the other hand, use fundamental indicators to select stocks that they expect to outperform some benchmark index over much longer investment horizons, often greater than 1 year. Application of machine learning techniques to long-term stock selection is less well explored and, to the author's knowledge, LSTM models have not been applied for directly predicting long-term stock returns. In the few papers that adopt this longer-term investment approach, the Random Forest model achieves the best prediction performance on unseen data [11, 12]. Unlike LSTM, which uses a *sequence* of input vectors to predict the output variable, Random Forest, like all other models applied to long-term stock price prediction tasks, makes a prediction based on single, independent input vectors that do not consider the dataset's temporal dimension.

This thesis hypothesises that the innate temporal dimension in fundamental data provides information regarding a stock's improvement or deterioration over time and might help improve long-term stock predictions. An LSTM and a Random Forest model are used to predict 1-year stock returns and each model's performance is benchmarked against the value-weighted S&P 500 index over a 41-year period. Section 2 conducts an extensive literature review, covering both the short-term stock price forecasting and long-term stock selection domains. Section 3 describes the data science process applied to this thesis, offering a comprehensive account and explanation of the data preparation, exploratory data analysis, modelling, and model evaluation decisions. Section 4 dissects model results and evaluates the profitability of model outputs when combined with a simple buy-and-hold trading strategy. Finally, section 5 concludes the thesis by summarising the findings, making recommendations for future work, and reflecting on project decisions. Following Section 5 is a glossary defining words or phrases formatted in ***bold italics***.

# 2 Literature Review

Stock price forecasting and stock selection literature can be generally separated into three categories: intraday, short-term (between 1 and 30 days) using technical analysis, and long-term using fundamental indicators. While this thesis centers on long-term stock selection, the utilisation of machine learning techniques in this field remains relatively unexplored compared to their application to tasks with a shorter investment horizon. However, short-term tasks share similar motivations and challenges, and the datasets exhibit similar characteristics. Therefore, this section will begin by reviewing short-term stock price forecasting literature before proceeding to examine the long-term stock selection literature that more closely resembles the experiment described in this thesis.

## 2.1 Short-term Stock Price Forecasting

### 2.1.1 Recurrent Neural Networks

In the last decade, RNN and its variants, Gated Recurrent Unit (GRU) and LSTM, have been the most popular sequence prediction models for short-term stock price forecasting, featuring in almost all contemporary papers where sequence prediction methods were used [7, 8, 9, 10]. Their ability to model complex, non-linear functions enables them to perform better than traditional machine learning sequence prediction models, such as Hidden Markov Models (HMM) and statistical models, such as ARIMA.

Di Persio and Honchar (2017) compare the performance of RNN, LSTM and GRU for forecasting the GOOGL stock price [8]. They use 5-years of daily OHLCV data with a time window of 30 days to predict the change in stock price over the next day, 5 days, 10 days etc. Both the GRU and LSTM outperformed the RNN on unseen data and both performed better when early stop and dropout were employed. Additionally, the paper provides an interesting examination of the hidden layer activations to analyse where models identified key price trends. Roondiwala et al. (2017) developed a deep LSTM architecture consisting of two LSTM layers and a dense layer, and compared its performance against a classic RNN for predicting the Nifty 50 index [9]. Both papers fail to demonstrate the generalisability of their results to other price forecasting tasks by only applying the model to one asset.

Bao et al. (2017) and Rather et al. (2015) both design hybrid models that utilise LSTM and RNN in combination with other models. Bao et al. (2017) proposes a three-stage approach with denoising via wavelet transform, deep feature extraction using Stock Auto Encoders (SAE), and an LSTM for induction [7]. They test the model using a buy-and-sell strategy and apply a thorough and well-considered scheme for incorporating trading costs, adding credibility and authenticity to their results. Moreover, their experimental set up with a moving window approach to validating and testing ensures robustness over time and makes full use of the dataset. They do note, however, that "deep learning methods are time-consuming" and that future work should look to heterogeneous computing-based methods for deep learning. Rather et al. (2015) propose an interesting hybrid approach, which combines the results of two linear models, ARMA amd Exponential Smoothing (ES) model, and one RNN model, using genetic programming to determine the optimal weights for each model's output [10]. The hybrid model captures sudden spikes in stock price and outperforms both the single-stage RNN and a Multi-Layer Perceptron (MLP) when applied to 25 stocks from various industry sectors. However, the authors observe that although the model outperforms single-stage ANNs, it does not generalise well to all datasets.

### 2.1.2 Hybrid Models and Modelling Innovations

In the short-term price forecasting domain, data scientists are increasingly employing model hybridisation to tackle challenges that are specific to the model or the task at hand. A selection of interesting and innovative examples are covered briefly here.

Genetic algorithms (GA) are frequently combined with other models to improve performance. Several studies apply GA to find globally optimal ANN weights and overcome the established shortcomings of the gradient descent algorithm, including long training times and tendency to converge to local optima. Kim and Han (2000) use GA to optimise the network weights and discretise 10 technical features [13]. Feature discretisation is used to remove noise from quantitative features and imitate the decision process of human investors. GA finds thresholds for discretisation, that, when combined with a feedforward ANN, output optimal results. However,

although it does remove noise from the data, discretisation also removes information from the data, which increases uncertainty.

Asadi et al. (2012) and Hadavandi et al. (2010) propose very similar multi-stage models. Both apply simple stepwise regression to identify useful technical features and utilise genetic algorithms to find an optimal set of initial ANN weights [14, 15]. This application of GA ensures that the feedforward ANNs in both papers find globally optimal solutions. Asadi et al. (2012) further optimise the training process by using the Levenberg–Marquardt (LM) algorithm to train network weights (once the GA has selected initial weights). They reproduced an experiment conducted in several other papers that involved forecasting short-term closing prices for seven Taiwan Stock Exchange indices. The proposed pre-processed evolutionary LM neural networks (PEL-MNN) model outperformed the models applied in each of the other papers, reinforcing the power of hybridisation for stock forecasting.

As illustrated in Rather et al. (2015), one approach to hybridisation is ensemble learning. Hsu (2013) use two methods for feature subset selection: Genetic Programming and a statistical procedure for ranking the information provided by each feature [16]. A BP ANN is then trained using both feature subsets and prediction results are compared and combined. Smith and Jin (2014) propose an interesting ensemble model that uses GA to find an optimal generation (ensemble) of RNNs for time-series prediction [17].

Patel et al. (2015) adopt an innovative perspective for overcoming the temporal gap between features at time $t$ and the target variable as the forecasting interval increases from time $t$ [18]. The first stage uses Support Vector Regression (SVR) to map technical indicators to their future values. The second stage uses these new values for the technical indicators as inputs to an machine learning model, which maps these values to the day's closing price. The two-stage models that use SVR, ANN, and Random Forest as the machine learning model in the second stage outperform their single-stage counterparts for forecasting intervals longer than 4 days. This result demonstrates the ability of hybridisation to solve single-stage model shortcomings.

Finally, Peachavanish (2016) proposes an interesting unsupervised learning method for selecting stocks from the SET100 with optimal technical characteristics, including trend and momentum [4]. K-means clustering was used to identify 10 clusters of stocks with similar characteristics at day $t - 20$ trading days. The profit and loss over the following 20 days for each cluster was then calculated and the most profitable cluster of stocks is selected for the strategy. The strategy outperforms its benchmark index by 75% and the paper gives good analysis of the periods for which the strategy underperforms.

## 2.2 Long-term Stock Selection

### 2.2.1 Overview

There are very few papers that use fundamental data to examine stock selection for investment horizons longer than 30 days. The author identified six papers that use a variety of models and fundamental data to select a portfolio of stocks.

Upadhyay et al. (2012) conduct one of the first long-term stock selection studies that utilise machine learning techniques. They use simple multivariate logistic regression to map fundamental company ratios to a binary target variable describing returns over the following year [19]. Although more sophisticated machine learning techniques and rigorous feature subset selection might be employed, the paper reports a 58.41% accuracy on the test set, significantly outperforming the baseline performance level of 33.3%.

Huang (2012) proposes a wrapper approach to feature subset selection that uses GA as part of a hybrid GA-SVR model [20]. In the first stage, GA is used to optimise the feature subset and Support Vector Regression (SVR) hyperparameters. The SVR component with optimised hyperparameters then induces the 1-year return using the feature subset selected. Huang (2012) reports the model results for several scenarios: with no feature subset selection, with GA for model hyperparameter optimisation only, GA for feature subset selection only, and GA for both feature subset selection and model hyperparameter optimisation. It is clear from the results that using GA for feature subset selection improves model results greatly, highlighting the importance of selecting an appropriate subset of fundamental indicators when constructing feature vectors.

Yu et al. (2014) identify a key characteristic of fundamental data - many company financial sub-ratios provide duplicate information to the model, causing "redundancy and low efficiency; even decreasing the quality of empirical results" [21]. The authors propose a hybrid PCA-SVM model that uses Principal Component Analysis (PCA) to reduce these sub-ratios to a one-dimensional feature for each ratio category, e.g. earnings ability, cash ratios, etc. The lower-dimensional feature vector is then inputted to a SVM model to classify

stocks in the top 25% of returns in the following year. The hybrid model attains an accuracy score of 61.79% on unseen data and outperforms the benchmark index. The authors note that results might be improved by modifying the investment strategy to weight stocks by their risk-return characteristics, rather than weighting stocks equally.

Ballings et al. (2015) also conduct a comparison of multiple models for stock return classification [11]. Inspired by the performance of ensemble classifiers in other fields, they compare the performance of three ensemble methods, Random Forest, AdaBoost and Kernel Factory, against that of four popular single classifier models, ANN, Logistic Regression, SVM and K-Nearest Neighbours. For the dataset used, Random Forest again performs best, and three out of the four best performing models utilised ensemble learning, demonstrating its capability for stock selection tasks. Although the authors use fundamental indicators advocated for in the literature as model inputs, it seems that many of the indicators provide the same or similar information. More sophisticated feature subset selection might have improved generalisation performance of some of the models tested.

Milesovic (2016) compares the performance of several machine learning models for classifying stocks into those which experienced >10% stock price growth over the following year (1) and those that didn't (0) [12]. This simple classification of stocks does not provide a rigorous evaluation of the model's performance and, instead, a strategy should be formulated and performance compared against a benchmark index. Furthermore, the paper conducts feature subset selection by iteratively removing features and assessing the model's test set performance. Again, this is suboptimal and is not guaranteed to provide insightful information about the best features for long-term stock selection. Finally, downsampling is used to tackle issues associated with class imbalance, causing the experiment to lose potentially valuable information from dropped instances. The author might employ upsampling or more sophisticated methods for dealing with class imbalance. Despite these problems with the experimental set-up, Milesovic (2016) finds that Random Forest significantly outperformed other models with an F-score of 0.75, a result consistent with similar papers.

To the author's knowledge, Sun (2019) is the only study that uses sequence prediction models (LSTM and GRU) for longer-term stock selection [22]. Unlike other papers that predict a target variable representing some change in stock price, Sun (2019) predicts the EBIT/EV ratio, a financial ratio commonly used as an indicator for future stock price growth. The LSTM and GRU outperform the standard feedforward ANN but neither significantly outperforms the other on this task. More sophisticated feature subset selection methods could be applied and the author notes that industry-related features or industry-specific models might improve performance.

### 2.2.2 Data Sources and Features

Despite conducting the same general stock selection task, the papers identified above are not directly comparable and it is important to consider their differences when evaluating methods and results. Firstly, each paper uses data on stocks from various markets and stock exchanges. International markets have varying maturity levels that affect transaction costs, efficiency, and volatility - and therefore returns. Secondly, although all papers identified employ fundamental indicators as model inputs, the combinations of indicators selected as feature vectors vary across papers. Finally, each paper applies models to data from different time periods and uses different experimental setups. The result are incomparable models that have limited application and heavily nuanced results. To gain a critical understanding of each paper's limitations, their data sources and experimental setups will be individually outlined and examined in detail.

Upadhyay et al. (2012) use data for the 30 most-traded companies from the Nifty Index in India. They select a four-year period 2005-2008 and map the annual figures for 7 company financial indicators to their categorical target variable. This results in a very limited sample of just 118 distinct company-year observations [19]. By selecting a sample of just 30 companies, the model is likely to incorporate industry bias. Furthermore, the time period selected is very short and the market was notoriously and uniquely turbulent during these years. Therefore, the model's results do not demonstrate efficacy in "normal" market conditions.

Huang (2012) use the 200 largest stocks by ***market capitalisation*** from the Taiwan Stock Exchange as their investment universe [20]. 14 financial ratios are selected from the literature and used as input to the GA component. The model is tested for the period 1996-2010, illustrating the efficacy of the model for selecting stocks in multiple economic environments.

Yu et al. (2014) and Ballings et al. (2015) both select 1 year of data from 2009-2010 and a combination of

company financial indicators from the relevant literature. Yu et al. (2014) choose 675 stocks from the A-share Shanghai Stock Exchange as their dataset and Ballings et al. (2015) 5,767 publicly listed European companies from a range of industries. In the case of both papers, the decision to use just 1 year of stock market data again fails to demonstrate the ability of the models to generate returns in different market conditions.

Milesovic (2016) reduces model bias towards one market by constructing a dataset of 1,739 stocks from different international markets, including the S&P 1000, FTSE 100, and S&P Europe 350 [12]. The authors initially map 28 financial ratios collected at the end of each quarter from Q1 2012 to Q3 2015 to the target variable. After evaluating model performance, the authors remove features iteratively, to construct a final feature vector constituting 11 financial ratios. Although the dataset spans a longer period than that of some other papers, the results are still limited in their applicability to different market conditions.

Sun (2019) examines stock selection in the North American market and constructs a dataset of the 3,000 largest stocks by market capitalisation listed on the NYSE, NASDAQ or AMEX. In line with standard practice, they elect to exclude financial companies from this stock pool. 5 income statement and 9 balance sheet indicators are selected from the literature and normalised by market capitalisation. Sun (2019) adopts the longest time period there is reliable data for, extracting quarterly data from 1960 to 2018. Data is inputted to the sequence prediction models in rolling 2-year periods, yielding 232 input vectors of size 8x14 per stock (if listed for the full 58-year period).

### 2.2.3 Model Application and Evaluation Methods

Due to the practical nature of stock selection, there is a broad spectrum of options available for evaluating models and curating trading strategies to apply them. These options encompass both classical data science methods and performance simulation tests and analysis from the financial domain. As a result, the evaluation methods used in each long-term stock selection paper are reviewed in turn.

Upadhyay et al. (2012), Yu et al. (2014), Ballings et al. (2015) and Milesovic (2016) present the stock selection task as a classification problem, using various models to classify stocks into some variation of "good" selections and "bad" selections. Upadhyay et al. (2012) present the macro-average and class-specific accuracies achieved by their logistic regression model across three stock classes: "POOR", "AVERAGE", and "GOOD". The authors then proceed to use a selection of statistical tests to assess the goodness of fit, appropriateness of the model and significance of each coefficient. Yu et al. (2014) also present the macro-average and class-specific accuracies for their model, which classifies stocks into those in the top 25% of returns and those outside it. They then apply the model to one year of trading and plot the selected portfolio's cumulative returns against the benchmark A-share index of Shanghai Stock Exchange for the subsequent 52 weeks. Although the model outperformed the benchmark by a notable margin, a single testing period is not enough to suggest significant outperformance. Ballings et al. (2015) use the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) metric to evaluate model performance. The authors then employ five times twofold cross-validation and the Friedman test to compare the performance of each of the seven algorithms. Then, to determine which algorithms' performances are significantly different, the Nemenyi post-hoc test is used [11]. To confirm results, the interquartile ranges of the AUC for each model's ten folds are compared as a measure of consistency of goodness. They received the same result for both tests. Finally, Milesovic (2016) evaluates each of their models using precision, recall and F-score. Paired $t$-tests with a 5% significance level are then used to compare the best-performing model to each other model to validate the significance of results.

Huang (2012) and Sun (2019) present the stock selection task as a regression problem, ranking stocks for selection by the continuous output of their models. Huang (2012) showcases the most comprehensive application and evaluation of their hybrid model's performance. Cumulative returns generated by the models are graphed against a benchmark index for the 15-year test period. The model outcomes are showcased with and without incorporating feature selection and model hyperparameter optimisation components, illustrating the augmented performance provided by these elements. Furthermore, Huang (2012) graphs the model's performance with a range of portfolio sizes (10, 20 or 30 stocks) from an investment universe of 200 Taiwanese stocks to examine the effects of diversification on model performance. Huang (2012) also acknowledges the importance of return volatility to potential investors and visualises the yearly volatility of stock returns within each portfolio. Finally, the importance of each feature to the model is identified and presented by analysing the proportion of periods the GA wrapper selected each feature to be part of the optimal subset for that period. In comparison, Sun (2019) provides a relatively superficial approach to application and evaluation, presenting just the Mean Absolute

Percentage Error (MAPE) of their models. The reason for choosing MAPE to evaluate model performance is not justified.

## 2.3 Summary

There are several important takeaways from this literature review:

1. Most of the papers across both short-term and long-term financial domains focus exclusively on the modelling choices and techniques.

2. Sequence prediction models have demonstrated outstanding performance in short-term stock price forecasting tasks. In comparison, they are relatively unused for long-term stock selection tasks, despite the innate temporal dimension in fundamental data.

3. Hybridisation and innovative modelling techniques are becoming increasingly prominent in the short-term literature. After much consideration, it was decided that creating and applying hybrid models fell outside the feasible scope of this thesis. However, their impressive outperformance of single-stage models in the short-term literature should encourage future authors to experiment with applying them to longer-term stock selection tasks.

4. There are very few long-term stock selection papers - just 6 identified by the author. These papers are not very comparable, with each choosing different markets, experimental setups, and evaluation techniques. Moreover, the results incorporate various biases to specific exchanges, industries, and market conditions.

5. The most insightful papers are those that evaluate model performance using techniques from the finance domain. These papers apply models in combination with an investment strategy and thus demonstrate their successful application to active trading.

# 3 Methodology

This section describes and justifies the methodological steps and decisions taken to achieve the results in Section 4. Each subsection examines a step in the data science process adopted in this thesis. The first subsection describes the dataset and the second examines the results of exploratory data analysis. The third subsection forms a large proportion of the methodology and both explains model mechanisms and justifies modelling decisions. The fourth and fifth subsections describe the chosen trading strategy and experimental set up. Finally, the sixth subsection examines performance metrics and methods chosen to evaluate model performance and behaviour.

## 3.1 The Dataset

### 3.1.1 Data Sources

The dataset of financial ratios is extracted from Wharton Research Data Services (WRDS) and contains 70 financial ratios for the entire universe of CRSP Common Stock constitutents, including North American publicly listed stocks from NYSE, NYSE American, NASDAQ, and NYSE Arca stock exchanges. The result is a dataset, which contains financial ratio data for a total of approximately 23,000 unique North American stocks over the period 1970-2022. The dataset includes monthly data for each company - annual and quarterly data are carried forward to the subsequent months before the next data become available. Following standard practice and because high leverage does not carry the same meaning as it does for other firms, finance companies have been excluded from the investment universe [23]. Finally, ratios are categorised into seven categories and a full list of financial ratios and their meanings are included in Appendix A.

The macroeconomic indicators selected for inclusion are Gross Domestic Product (GDP), Consumer Price Index (CPI), Federal Funds Rate, and the yield on the 10-year treasury bond. Each of these factors conveys different information about the current economic conditions and are important considerations for long-term investors. Data are downloaded from the Federal Reserve Bank of St. Louis database.

Lastly, data on stock prices and the S&P 500 index are downloaded from the CRSP database on WRDS.

### 3.1.2 Data Preparation

The first step in preparing the dataset was merging the macroeconomic data to the financial ratios data. The change rate of GDP and CPI indicators are more useful to investors than the raw data. Therefore, the 1-month and 1-year rates of change for each of these features were extracted and the resulting data for 6 macro-economic indicators was left-joined to the dataset of financial ratios.

Next, pricing information, including monthly market capitalisation and price data, needed to be extracted for each [stock, year, month] instance in the dataset. Before this could be accomplished, instances with duplicate entries or missing stock details were dropped.

Finally, to extract the target variable, the 1-year return rate, the following equation was applied to each [stock, year, month] instance:

$$return_t = \frac{price_{t+1} - price_t}{price_t}, \tag{1}$$

where $price_t$ is the current market price and $price_{t+1}$ is the market price for the asset one year in the future. Instances with a public date later than 31-12-2021 were dropped as there was no target variable data for these points.

The resulting final dataset included:

1. 70 financial ratios,
2. 6 macroeconomic indicators,
3. market capitalisation, and
4. the 1-year return

for all publicly-listed North American stocks over the period 1970-2021.

## 3.2 Exploratory Data Analysis

### 3.2.1 Missing Data

Missing data can significantly affect model performance and many machine learning models are unable to handle missing data effectively. The dataset contained significant proportions of missing data for each financial ratio - at least 0.5% of instances. Upon investigation into the nature of these instances, it seemed that the missing data was not erroneous but a feature of the dataset. For example, a stock with missing data for the variable, Debt to Equity ratio, may have no debt on their balance sheet. Therefore, it would be incorrect to impute missing values and dropping indices with one or more missing values would remove valuable information from the dataset and decrease the practical value of models. Decisions and steps taken to handle missing data prior to modelling is discussed in Sections 3.3.1.2 and 3.3.2.5.

### 3.2.2 Financial Ratios - Distributions, Outliers and Correlation

Many company financial indicators are calculated as ratios, e.g. Price-to-Book ratio. As a result, many of the features exhibited heavy-tailed distributions with extreme outliers caused by near-zero denominators. Some machine learning models are better at handling outliers than others and these are dealth with during pre-modelling data preparation.

Histograms for each variable were used to examine the distributions of individual features and each feature was mapped against the target variable using heatmaps to explore any patterns and correlations. After stemming the data at five median absolute deviations either side of the median value, most features demonstrated a generally bell-shaped curve, some a power law relationship, and some more unique distributions, e.g., dividend yield. The target variable exhibited a positively skewed Gaussian distribution. This is expected, because among other reasons, returns exhibit limited downside and potential for unlimited upside.

There is very high correlation between many of the features, with high feature redundancy caused by the fact that many of the features are variants of the same financial ratio. Figure 14 in Appendix B maps these correlations as a correlation matrix. Features are organised into ratio categories (see Appendix A) and it is clear that there is high correlation between features within the same category. Although the Random Forest model can effectively handle features with such characteristics, Section 3.3.2.4 describes the feature subset selection process for modelling using the LSTM.

### 3.2.3 Financial Ratios - Dimensionality Reduction

It was hypothesised that applying unsupervised learning methods to the financial ratio data may identify clusters of stocks that share similar traits. Furthermore, by using dimensionality reduction techniques to map the data to two dimensions and visualising returns using colour, it may have been possible to discern which features of stocks more frequently led to positive returns in the subsequent year.

$t$-Stochastic Neighbourhood Embedding ($t$-SNE) was chosen for this task. $t$-SNE aims to preserve local structure from the original data space when mapped to the latent space. It does this by placing Gaussian distributions centered on each datapoint in the high-dimensional space, and mapping each point to the low dimensional space in a way that creates similar probability distributions.

After experimenting with various perplexity values, the visualisation in Figure 15 in Appendix B depicts the results of $t$-SNE for a bootstrap sample constituting 10% of the unique [company, reporting year] instances in the dataset. There is one cluster clearly separated from the main mass. Upon further inspection, this cluster represents financial sector companies that are still in the dataset. $t$-SNE also did not reveal any clusters that experience a greater proportion of positive returns.

It was further hypothesised that applying $t$-SNE on data from specific years might reveal clusters of stocks with more positive returns. This stemmed from the idea that performance of two popular stock types, ***growth stocks*** and ***value stocks***, varies in different market conditions. However, the experiment yielded similar levels of separation. These results imply that there are few clearly identifiable stock "types" and that supervised learning methods are likely required to identify groups of better-performing stocks in this dataset.

### 3.2.4 Macroeconomic Indicators

Visualising the fluctuation of macroeconomic indicators reveals economic cycles and fluctuations caused by rare, exogenous events. These patterns and abnormalities are important to consider when evaluating model performance. Machine learning models are trained only on historic data and therefore unseen economic conditions may negatively impact model performance.

Figure 16 in Appendix B displays the values for each macroeconomic indicator for the period 1970-2022. A high period of inflation can be seen in the 1970s, closely mirrored by the high inflation seen emerging in 2021-2022. The period of ultra-low inflation between 2010-2017 can be seen also. For the most part, GDP growth rate and both interest rates follow a similar pattern. The exceptions are the crises years, 2008 and 2020, during which the world experienced large economic slowdowns. The largest abnormality can be seen at the right end of the graph, where the economy strongly rebounds following the COVID-19 pandemic. Generally, the economic conditions seen in the dataset appear to move in four-year cycles.

### 3.2.5 Stock Composition

Stock composition is important to consider at this stage. Not only is it important to long-term investors, but these metadata-related factors influence crucial modelling decisions concerning experimental set up and evaluation of results.

There are several reasons why potential investors consider market capitalisation when selecting stocks. Firstly, smaller companies are traded less frequently and therefore markets are characterised by larger ***market impact***, larger ***bid-ask spreads***, and generally higher liquidity risk. Secondly, the combination of this lower liquidity, greater sensitivity to market sentiment, and speculative investor behaviour causes smaller stocks to experience higher return volatility. Finally, due to the generally higher risk that they carry, many funds are restricted from holding small cap stocks, and often larger funds are unable to buy enough shares in a company to justify their exposure.

The market capitalisation of stocks is generally categorised into several types. According to Investopedia, these types are:

- Mega Cap - companies with a market cap of $200 billion or higher,
- Large Cap - companies with a market cap beetween $10 billion and $200 billion,
- Mid Cap - companies with a market cap between $2 billion and $10 billion,
- Small Cap - companies with a market cap between $300 million and $2 billion,
- Micro Cap - companies with a market cap between $50 million and $300 million, and
- Nano Cap - companies with a market cap below $50 million [24].

To explore various market capitalisation restrictions and understand the stock choices made by the models, it is important to categorise each stock into one of the six specified categories. When mapped for each year 1970-2022, the distribution of market capitalisation remains unchanged but, as depicted in Figure 17 in Appendix B, the median value increased from $44.3m in 1970 to $1.1bn. Therefore, the above cap boundaries were scaled for each year using the following equation:

$$boundary_{year} = boundary_{2021} \times \frac{med(marketcapitalisation_{year})}{med(marketcapitalisation_{2021})}, \tag{2}$$

where $med()$ represents the median value. The resulting distribution for market capitalisation post-discretisation is approximately equivalent to the distribution for 2021 and is visualised in Figure 1.

Other factors to consider pre-modelling are the yearly size of the investment universe and the length of time each company is in the dataset for.

The prior is useful for understanding the proportion of available stocks selected each year. Figure 18 in Appendix B displays the number of unique stocks in the dataset per reporting year. The graph indicates that the number of companies triples from 1970 to the mid 1990s and then declines again to 2020. The declining count throughout the 2000s coincides with the dotcom crash and global financial crisis.

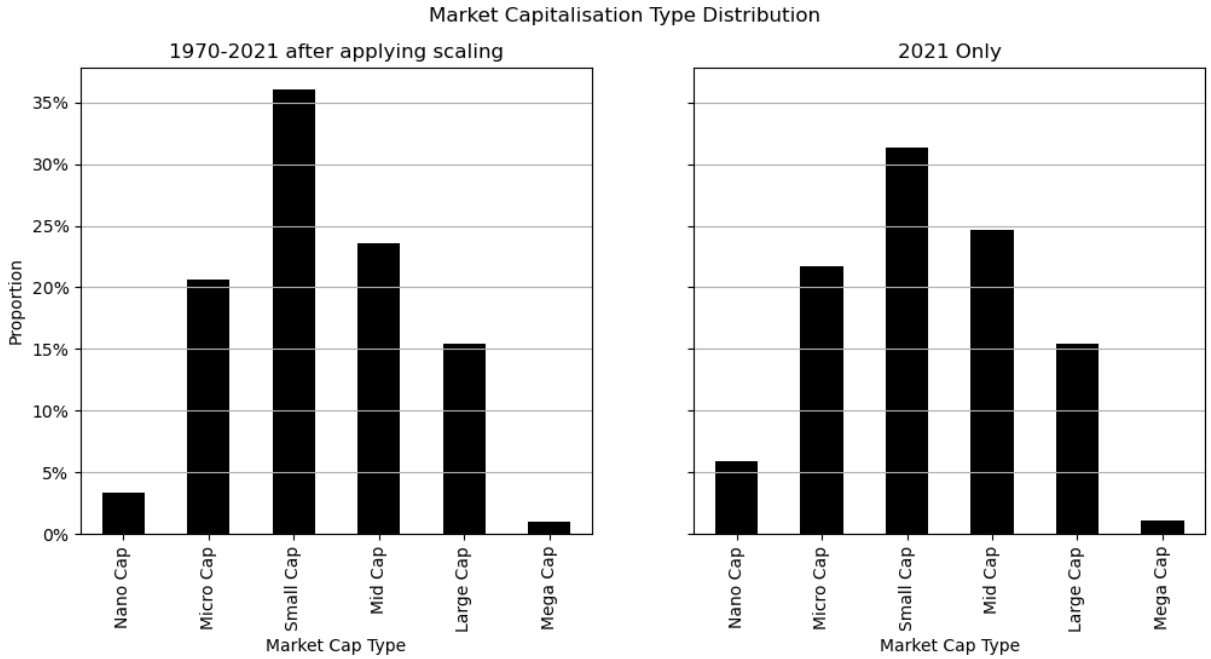


Figure 1: Market Cap Type Distribution.

The length of time stocks are included in the dataset is important for selecting an appropriate sequence length and minimum sequence length when applying sequence modelling techniques. Figure 19 in Appendix B visualises this distribution as a histogram and identifies the mean length of time as 10.5 years. It can also be seen that the distribution follows a power law. Additionally, approximately 23% of stocks are in the dataset for less than one year, while 2.5% are in the dataset for the full duration of 52 years.

#### 3.2.6 Returns

This thesis aims to apply models that outperform average market returns. Market returns fluctuate over time in response to market sentiment and macroeconomic factors. Figure 20 in Appendix B visualises the monthly average for the target variable (1-year return). As expected, the target variable is very volatile and it is clear that periods of strong returns follow distinct declines. The average monthly value for the target variable aggregated across all stocks is 15.14%.

As discussed previously, stocks with smaller market caps are expected to be more volatile. Figure 2 demonstrates the extent of this effect for the North American market, a market with generally strong ***venture capital*** presence. Nano Cap and Micro Cap stocks experience significantly higher average returns, with Nano Cap stocks experiencing average 1-year returns greater than 100% of their equity value for a large number of months. This effect appears especially strong for periods of strong market sentiment and during periods of crisis recovery, e.g. the 1990s and the periods following the dotcom crash (2000-2002), global financial crisis (2008), and COVID-19 pandemic (2020).

### 3.3 Modelling

#### 3.3.1 Random Forest

Chosen for its superior performance in contemporary literature, the first model used to predict 1-year returns and select stocks is a random forest regressor. Random forest models are ensemble learning models that use multiple decision tree base learners to make predictions. To understand the mechanisms of random forests, one must first understand decision trees.

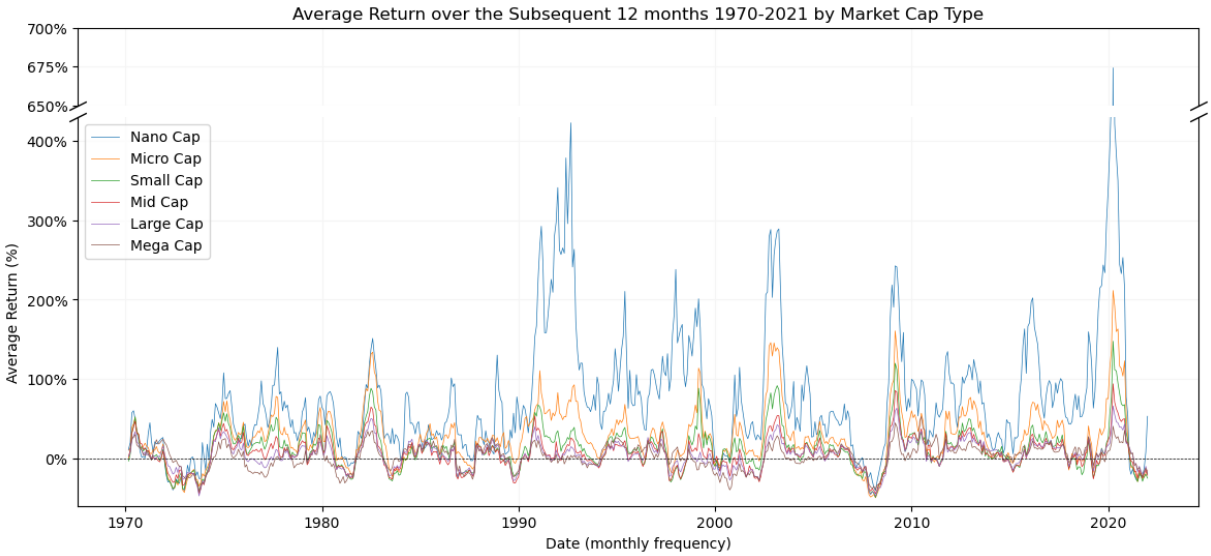


Figure 2: Target variable (1-year return) 1970-2021 by market capitalisation type.

Decision trees are a popular non-parametric machine learning model that can be used for classification or regression. Decision trees recursively partition the dataset into smaller subsets using a purity function at each node to identify the optimal input feature and value to split the dataset on. This process continues until either a stopping criterion is met, such as a predefined maximum tree depth, or until each leaf contains just one class in the case of classification, or a very small number of instances in the case of regression. The result is a hierarchical tree structure where leaf nodes contain the final predictions for classification or regression tasks.

Random forest models combine the predictions of multiple decision trees to reduce overfitting, improve generalisation, and take advantage of "the Law of Large Numbers". A predefined number of decision trees are initiated. A ***bootstrap sample*** is selected for input to the first node in each tree and each tree is constructed independently. This improves generalisation performance. At each node, instead of every feature being considered, a random subset of features are evaluated, helping to decorrelate the results of each tree and introduce variation to the tree predictions. Random forest regressors make predictions by calculating the mean of results from all individual tree predictors. It is possible to assess the importance of individual features by ranking each feature by its average impurity reduction across decision tree learners.

The non-parametric nature of decision tree and random forest models make them more resistant to datasets with features that exhibit non-normal distributions and outliers. Furthermore, by utilising multiple decision trees and decorrelating their output, random forests generalise well to unseen data and can ably handle noisy datasets [25].

There are several limitations to random forest regressors. They are very computationally intensive, especially to train on large datasets. Furthermore, they are more complex than the decision tree learners that they are based on, which makes interpreting why the model made specific predictions much more difficult. Finally, due to the deterministic nature of decision trees, the model does not extrapolate well to data that are dissimilar to the training set. This problem is exacerbated by the fact that extreme values are often normalised through the averaging of individual tree outputs.

Further to its outperformance in the contemporary literature, there are several additional reasons why it is hypothesised that a random forest regressor is well-suited for application to this long-term stock selection task. The dataset used in this thesis constitutes high-dimensional feature vectors, with redundancy caused by high feature correlations. Furthermore, a proportion of the company financial features exhibit unique, non-normal distributions and many of the ratios data contain extreme outliers. The model's non-parametric nature suggests that it should model data with these characteristics well. However, there are a few concerns. The dataset contains many missing values, which need to be handled before input to the random forest. Secondly, the financial markets are affected by real-world events and thus experience novel statistical patterns. On one hand, the model's inability to extrapolate may inhibit its performance in such events. On the other hand, the

averaging of a sufficiently large number of decorrelated decision trees may improve generalisation in such events and have the opposite effect.

#### 3.3.1.1 Software Implementation

Sci-kit Learn's RandomForestRegressor class is used to apply random forest regression to the dataset. The class uses the Classification And Regression Trees (CART) algorithm to construct each decision tree. This is a greedy algorithm that applies a binary split at each node based on the best feature and threshold to split on. The default setting for split criterion is applied, which selects features that minimise the mean squared error (MSE) at the next node. To improve generalisation and reduce computation time to a reasonable window, the size of the bootstrap sample at each tree is 10% of the entire dataset. To introduce heterogeneity across individual learners, the subset of features that a split can be selected from is equal to the square root of the total number of features.

#### 3.3.1.2 Data Preparation

Random forest models cannot take missing data as input and, as discussed in Section 3.2.1, there are many instances of meaningful missing data in the dataset. First, it was hypothesised the normalising each feature using Max-Min normalisation and replacing null values with an arbitrarily large negative number might enable the model to split on missing data. A second hypothesis involved dropping features that contained greater than 120,000 missing values and then dropping instances with any missing data. By handling missing data in this way, 41 features were still retained and only 11% of instances were dropped. Both methods were tested for several periods, and results suggested that the second method greatly outperformed the first. Thus, missing data in both the train and test sets was handled in this way for each simulation period.

#### 3.3.1.3 Hyperparameter Optimisation

The two most important hyperparameters for random forest models are the number of weak tree learners and the maximum allowable depth of each tree. More decision trees improves generalisation, reduces variance of predictions, and increases resistance to noise in the dataset. However, computational complexity is increased, slowing the training process. Moreover, there are diminishing returns to increasing the number of trees and, if the dataset is insufficiently large, then there is potential for overfitting. Setting a maximum tree depth is similar to pruning individual decision trees insofar as it helps to reduce overfitting to the training set. When a maximum tree depth is set, the individual learners are more likely to base their splits on meaningful patterns, rather than increasingly arbitrary patterns found in the bootstrap samples as the number of splits increases.

Grid search is applied to select optimal model hyperparameters. The entire dataset contains approximately 2.4 million instances. Therefore, to reduce computational complexity whilst retaining results validity, 5-fold cross-validation is used on a bootstrap sample totalling 10% of the dataset. The grid search space constitutes maximum tree depth in the range 3-13 and number of trees in the range 50-350 with a step size of 25. Mean Squared Error (MSE) is used to measure the performance of each model, given by the equation:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3}$$

where $n$ is the number of data points, $y_i$ are the observed values for $y$, and $\hat{y}_i$ are the predicted values for $y$. Figure 3 graphs the mean MSE performance across folds for each iteration.

It can be seen that MSE decreases from 1.92 to 1.80 as model complexity increases. Sudden increases in the MSE are symptomatic of the grid search process and mark the first iteration for a new maximum tree depth and therefore a reduction from 350 trees to just 50. The diminishing return to adding more trees can be seen by the line plateauing following this increase, and it seems that, considering the trade-off between performance improvements and computational complexity, 350 trees is the near-optimal number of trees. Increasing maximum depth improves performance. However, once again, performance plateaus above a certain threshold. After reviewing the graph, it was decided the the optimal model hyperparameters are 350 trees with a maximum depth of 13.

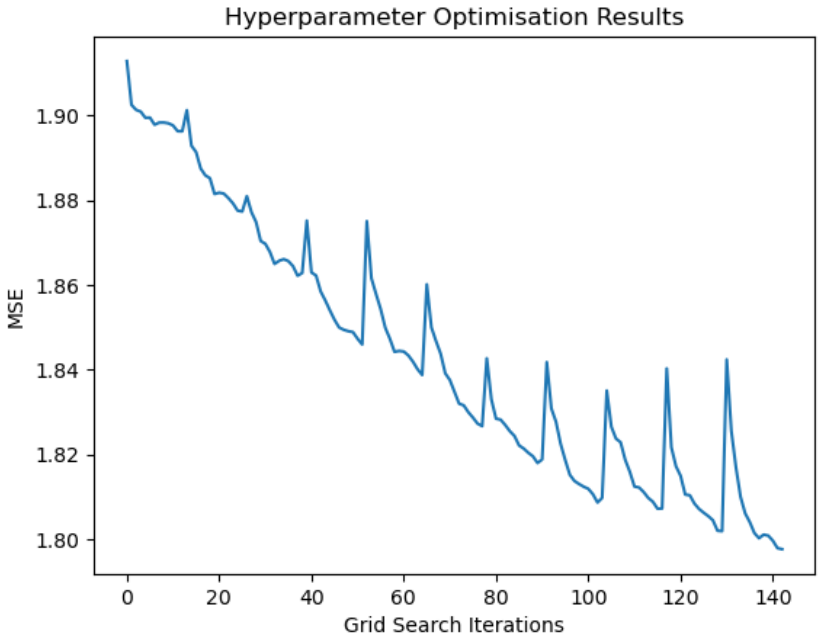


Figure 3: Random Forest Hyperparameter Optimisation.

#### 3.3.2 LSTM

LSTM networks are a popular sequence prediction neural network that aim to solve the vanishing gradient problem associated with traditional RNNs. To understand how an LSTM makes predictions, one must first understand RNNs.

##### 3.3.2.1 RNN Structure and Algorithms

RNNs are a class of neural network designed to handle sequential data. They utilise repeated cells that have the same weight matrix and number of neurons. A hidden state is maintained and updated after each cell, using the equation:

$$\mathbf{a}^t = g_1(\mathbf{W}_{aa}\mathbf{a}^{t-1} + \mathbf{W}_{ax}\mathbf{x}^t + \mathbf{b}_a), \tag{4}$$

where $\mathbf{a}^t$ and $\mathbf{a}^{t-1}$ are hidden states, $g_1$ is an activation function, $\mathbf{W}_{aa}$ and $\mathbf{W}_{ax}$ are weight matrices that are unchanged across cells, $\mathbf{x}^t$ is the input vector at time $t$, and $\mathbf{b}_a$ is the bias vector. The hidden state is used to calculate the output from each cell and is passed as input to the next cell. This enables the network to "remember" information from previous layers and incorporate it into the calculations for the current layer. The following equation is used to generate the output $y^t$ from one such cell:

$$\mathbf{y}^t = g_2(\mathbf{W}_{ya}\mathbf{a}^t + \mathbf{b}_y), \tag{5}$$

where $g_2$ is an activation function, $\mathbf{W}_{ya}$ is the weight matrix, and $\mathbf{b}_y$ is the bias vector. For a regression task, such as the one described in this paper, only the output from the final cell in the recurrent layer is used to make predictions. Figure 4 features a diagram of one such cell [26].

For regression tasks that use mini-batch variants of the gradient descent algorithm (such as this one), some loss function is calculated, usually MSE (see equation 3), at each time step for each batch of training instances. The loss, $L$, for all time steps is then calculated using the equation:

$$L = \frac{1}{T}\sum_{t=1}^{T} MSE_t, \tag{6}$$

where $T$ is the total number of time steps and $MSE_t$ is the MSE at time step $t$. Back Propagation Through Time (BPTT), a variant of the standard Back Propagation algorithm for training RNNs, propagates this loss

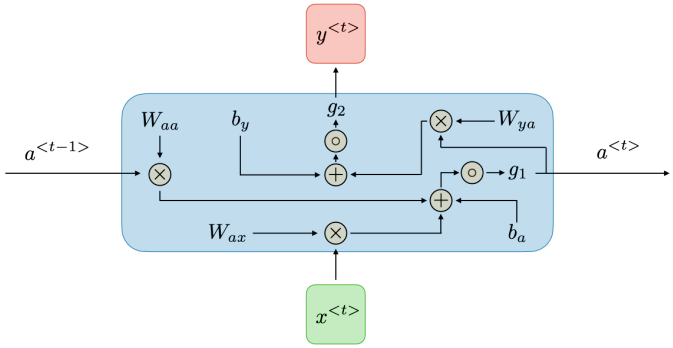


Figure 4: Diagram of a recurrent layer cell [26].

back through each time step to calculate gradients with respect to the loss for each weight matrix at each timestep ($\frac{\partial L^{(T)}}{\partial W}$). Some variation of the gradient descent (GD) algorithm uses these gradients to update each weight matrix.

GD iteratively adjusts weight matrices in the opposite direction to the gradient of the cost function, thus moving the weights closer to their optimal values with respect to the loss. There are several variants of GD, including stochastic gradient descent (SGD) and mini batch gradient descent (MBGD). SGD calculates loss gradients and adjusts the weights after each randomly selected training instance. This introduces noise to the parameter optimisation, which reduces speed of convergence but ensures that the algorithm does not get stuck in local optima. Furthermore, it is more computational efficient than traditional GD. MBGD updates parameters after calculating the loss for batches, usually of size 32, 64 or 128 training instances. It provides a balance between the computational efficiency of SGD and the stability of the traditional GD algorithm.

Learning rate is a model hyperparameter that specifies the step size for updating weight matrices, influencing speed and stability of convergence. The chosen algorithm continues this optimisation process over the training set for a predefined number of ***epochs***.

#### 3.3.2.2 LSTM and Gated Units

As mentioned previously, RNNs suffer from the vanishing gradient problem. This occurs when the gradients of the loss function with respect to the weight matrices become very small as they are backpropagated through the RNN cells. As the gradient vanishes, the weights are updated by smaller amounts, causing the algorithm to converge slowly and find suboptimal solutions. Furthermore, the weight parameters in RNNs perform two functions simultaneously: they provide information for the current decision and decide which information is carried forward to future cells [27]. Both these problems mean that RNNs generally struggle to retain long-term information.

This is a problem for long-term stock selection: the full history of economic conditions and stock performance patterns throughout the sequence should impact the context in which the model interprets the conditions at the time of investment.

LSTMs solve this problem by introducing three gates to the traditional RNN cell that help the model to control how much information should be kept in memory, forgotten, and passed on as cell output [28]. Figure 5 shows a diagram of an LSTM memory cell, where:

- $\mathbf{x}_t$ represents the input vector for time step $t$,
- $\mathbf{h}_{t-1}$ and $\mathbf{h}_t$ represent the previous hidden state and new hidden state,
- $\mathbf{c}_{t-1}$ and $\mathbf{c}_t$ represent the previous context and updated context,
- $\mathbf{f}$, $\mathbf{i}$, and $\mathbf{o}$ represent the forget, input, and output gates, and
- $\odot$ represents the element-wise multiplication of two vectors or matrices (sometimes called ***Hadamard product***).

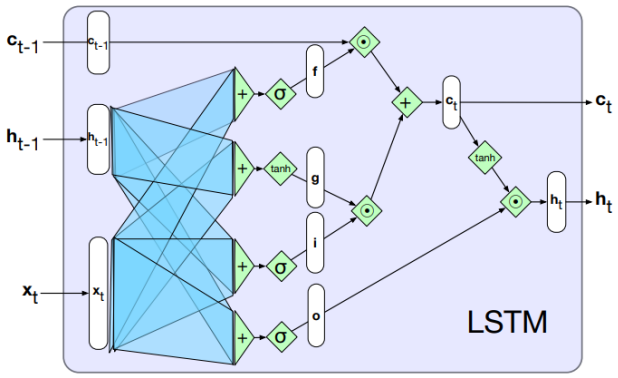


Figure 5: Diagram of a LSTM memory cell [27].

LSTM gates have similar structures, consisting of a feedforward layer that passes a weighted sum of the previous state's hidden layer and current input through a sigmoid function with equations:

$$\mathbf{f}_t = \sigma(\mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{W}_f \mathbf{x}_t), \tag{7}$$

$$\mathbf{i}_t = \sigma(\mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{W}_i \mathbf{x}_t), \tag{8}$$

$$\mathbf{o}_t = \sigma(\mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{W}_o \mathbf{x}_t). \tag{9}$$

The sigmoid function,

$$\sigma(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}}}, \tag{10}$$

ensures that the elements of the output vector for each gate are pushed towards either 0 or 1, and, when multiplied element-wise, perform an effect similar to a ***binary mask***. The role of each gate output will be covered in turn.

As depicted in Figure 5, the Hadamard product of the forget gate output, $\mathbf{f}_t$, and the context from the previous cell, $\mathbf{c}_{t-1}$, pushes the value of $\mathbf{c}_{t-1}$ towards either 0 or 1 to delete information from the context that is no longer needed:

$$\mathbf{k}_t = \mathbf{c}_{t-1} \odot \mathbf{f}_t \tag{11}$$

Next, the input gate is used to select information to add to the current context, $\mathbf{c}_t$. The output from the input gate is multiplied element-wise with $\mathbf{g}_t$, where:

$$\mathbf{g}_t = \tanh(\mathbf{U}_g \mathbf{h}_{t-1} + \mathbf{W}_g \mathbf{x}_t). \tag{12}$$

This is the same equation used to calculate the hidden state in a traditional RNN cell. The resulting equation is

$$\mathbf{j}_t = \mathbf{g}_t \odot \mathbf{i}_t, \tag{13}$$

and this is added to the modified context from the previous cell, $\mathbf{k}_t$, to get the current context vector,

$$\mathbf{c}_t = \mathbf{j}_t + \mathbf{k}_t. \tag{14}$$

Finally, the output gate is used to decide which information is needed for the current hidden state, which is calculated as:

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{15}$$

The LSTM structure solves the problems associated with the traditional RNN by distributing the task of deciding which information to retain across time steps across multiple gates; each with their own weight matrices that are optimised using some variant of GD.

This thesis hypothesises that applying sequence prediction models to the long-term stock selection task for the first time will improve investment outcomes. There is an innate temporal dimension to fundamental data that incumbent models, like the random forest, fail to account for in their predictions. However, prudent human investors utilise this historical data in their decision-making process in several ways:

- Improvement or deterioration in financial performance can only be ascertained by assimilating company results over a period of time.
- Market sentiment and consequently stock performance depends on macroeconomic conditions, which are not determined simply be indicators at the time of investment, but by the stage in the economic cycle and the direction that economic indicators are moving. This can only be deduced from the sequential macroeconomic data.

By applying the LSTM model to long term stock selection, it is hoped that the information above is captured by the model and investment outcomes echo the model's success in contemporary short-term literature.

#### 3.3.2.3 Software Implementation

The python library, PyTorch, was used to construct and run the LSTM model. The MSE loss function and Adam optimizer were used to optimise the model. The Adam optimisation algorithm is a more advanced variation of gradient descent that adjusts learning rates for each parameter individually, improving performance for most optimisation problems.

#### 3.3.2.4 Feature Subset Selection

As discussed in Section 3.2.2, there is high correlation between features. LSTM models require large amounts of training data to optimise the weight matrices and prevent overfitting. After sequencing the input vectors for input to the LSTM, the size of the training set is reduced greatly and totals just 4,000 instances approximately for the earlier test years. The ***curse of dimensionality*** means that, if it has too many parameters to estimate from limited training data, the model may overfit the training set and struggle to generalise effectively. Furthermore, ***multicollinearity*** in the variables can destabilise parameter estimates.

To address these issues without excessively diminishing information from the input data, for each test period and corresponding trained random forest model, the 12 features with the greatest feature importances were used to construct input matrices for the LSTM.

#### 3.3.2.5 Data Preparation

Data preparation for the LSTM model consisted of five consecutive steps:

1. **Resample the data from monthly to quarterly frequency**: As many of the features are extracted from quarterly or annual results and are filled forward in the dataset, the values for many features remain constant across time steps. Therefore, it is found that the LSTM performed better after resampling the data to quarterly and dropping excess instances. The resulting LSTM inputs had time steps at 31/03, 30/06, 30/09, and 31/12 annually.
2. **Winsorisation**: The data contains extreme outliers, as discussed in Section 3.2.2. Generalised Linear Models (GLM), including many neural networks, do not handle outliers effectively. In their paper exploring the effects of outliers on neural networks, Khamis et al.(2005) found that increasing the quantity and magnitude of outliers in both the training and test data yielded statistically significant increases in MSE [29]. Winsorisation is a data transformation that caps values above a certain predefined threshold. It was applied to each input variable and the upper bound, $UB$, and lower bound, $LB$, thresholds were calculated as:
$$UB, LB = med(\mathbf{x}) \pm (5 \times MAD(\mathbf{x})), \tag{16}$$
where $med(\mathbf{x})$ is the median and $MAD(\mathbf{x})$ is the median absolute deviation of the variable. ***Robust statistics*** were selected over mean and standard deviation for their robustness to outliers.
3. **Normalisation**: Results in the literature suggest that the best results when applying neural networks are achieved when all input features are in the same order of magnitude, especially if in the order of one [30]. Therefore, model input is normalised using min-max scaling, with the equation,
$$\mathbf{X}_{norm} = \frac{\mathbf{X} - \mathbf{X}_{min}}{\mathbf{X}_{max} - \mathbf{X}_{min}}. \tag{17}$$

Due to the high number of outliers for the target variable, caused by Nano Cap stock instances with extremely high positive returns, the target variable is also winsorised at the $5 \times MAD(\mathbf{x})$ threshold and normalised using min-max scaling.

4. **Missing values**: A relatively simple approach is adopted for handling missing values. Missing values are replaced with the value, $-1$, which lies outside the [0, 1] range exhibited by input variables post-normalisation. As discussed in 3.2.1, the missing data is not erroneous, but a feature of the dataset. Therefore, it is hoped that the network treats the values as missing and learns their meaning as such.

5. **Padding**: Sequences that have length greater than or equal to the minimum length for LSTM sequences, but less than the maximum length are padded using using a tensor of shape $m \times n$, where $m$ is the number of features and $n$ is difference between the length of the sequence and the maximum sequence length. Each element in the tensor is equal to $-1$.

#### 3.3.2.6 Hyperparameter Optimisation

There are multiple hyperparameters to be optimised when constructing an LSTM model. Given the computation time required to prepare training and testing datasets for the LSTM, it is not feasible to use grid search and employ $k$-fold cross validation to optimise every hyperparameter. Therefore, it was decided that a grid search be conducted over the following four hyperparameters with variable ranges:

- maximum sequence length between 1-5 years with a step size of 1 year,
- minimum sequence length between 1 year and the maximum sequence length,
- number of hidden layers between 1-3,
- number of hidden units in each hidden layer between 10-50 units with a step size of 10 units.

Each instance in the dataset constitutes a sequence of length,

$$sequencelength_{min} < sequencelength_{dataset} < sequencelength_{max}.$$

Therefore, the dataset is different for each pairwise [maximum sequence length, minimum sequence length] combination. To reduce computation time, a bootstrap sample totalling 20% of this dataset is used to test the two model hyperparameters relating to network architecture. To decrease computation time without reducing performance, a learning rate scheduler with a multiplicative learning rate is employed and training is stopped when MSE on the validation set increases post-epoch. The LSTM is tested using a train/test split of 80/20 and 18 features used most in other papers. This ensures comparability between iterations but avoids the problems discussed in Section 3.3.2.4.

Performances on the test set returned the results in Figure 6. It is clear that k-fold cross-validation might have improved the consistency of results. Longer sequence lengths seem to reduce MSE but there is no clear patterns in any of the other variables. Despite this, a sequence length of 5 years and minimum sequence length of 2 years seems to perform best. Following the principle of ***Occam's Razor***, values selected for the other two hyperparameters are 1 LSTM layer of 10 hidden units.

During simulation, the size of the training set varies greatly with the size of the expanding window (see section 3.5), and thus the optimal learning rate and early stopping point changes. A constant learning rate of $5 \times 10^{-5}$ and an early stop of 8 epochs that increases linearly with the size of the training set is found to be sufficient.

## 3.4 Trading Strategy

A simple ***buy-and-hold strategy*** is used in conjunction with the model to assess the profitability of model selections. Although a ***long-short equity strategy*** was considered, a buy-and-hold strategy better reflects the behaviour of long-term investors like ***mutual fund*** managers. Although mutual funds are not restricted from ***short-selling***, short-sales have unlimited downside and so expose funds to risk that is often considered excessive by long-term investors.

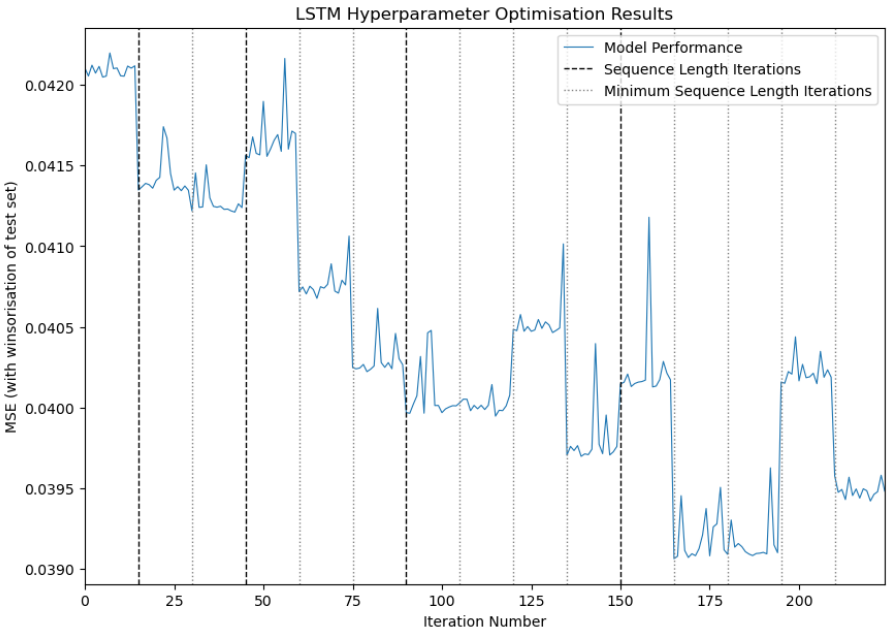


Figure 6: LSTM hyperparameter optimisation results.

The $n$ stocks that are predicted by each regression model to have the highest value for the target variable (1-year return) are purchased in equal weights. To evaluate performance for various levels of ***portfolio diversification***, as in Huang (2012), $n$ can take values from 30, 50, 100, and 200 stocks. The strategy sells all stocks currently held at 31st December each year and purchases a new portfolio of stocks, reinvesting all profits from the sale of assets. Although the restriction that the portfolio is rebalanced only once per year is a simplification and superior performance might be achieved by rebalancing the portfolio using a more intelligent mechanism, it is sufficient to directly compare the performance of each model.

***Trading costs*** are ignored in this thesis. This is because the portfolio is rebalanced so infrequently that trading costs are likely to be negligible. Furthermore, implementing a trading cost scheme would be unnecessary insofar as to compare the models' performances.

## 3.5 Experimental Setup

To assess the profitability of each model over time, 41 test periods from 1981-2022 were used. Models were trained using an expanding window, as depicted in Figure 7. This enables the models to make predictions for each period based on all historical data from 1970 onwards. The first investment date is 31/12/1981, 10 years after the first date in the dataset, to ensure that the models are trained on a sufficiently large training set for the first few test periods.

In live deployment to real-world scenarios, stock selection models like the ones described in this paper only have access to data available at that point in time. It is a common pitfall when ***backtesting*** prospective models to incorporate lookahead bias into the results. Lookahead bias refers to the unintentional inclusion of future information or data that was not available at the time of analysis. It can lead to overstated results and misplaced confidence that can lead to potentially devastating loss of funds when applied to real trading scenarios. For each test period, the most recent data point used in the training set is the 31st December one year prior to the date of investment. This ensures that values for the target variable (1-year return) in the training set do not allow the model to look ahead.

Given that the LSTM is a sequence prediction model and the random forest a static prediction model, the training sets and test set are defined differently to account for this distinction.

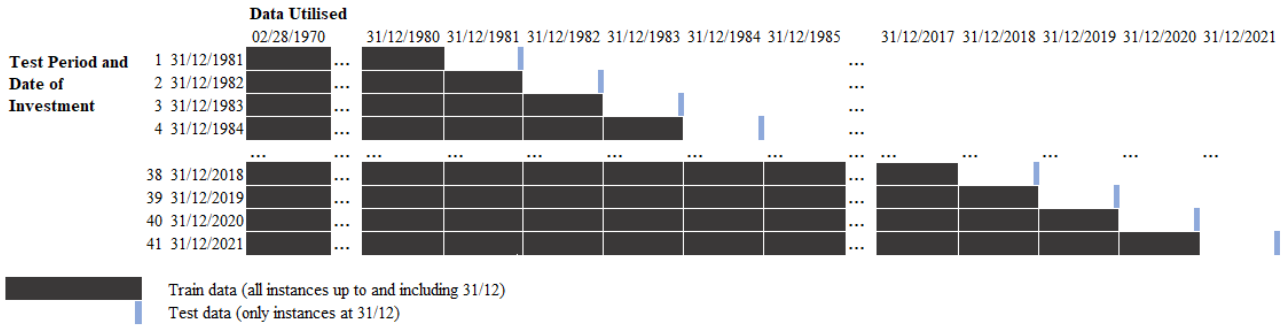


Figure 7: Diagram showing the experimental setup for each test period.

### 3.5.1 Random Forest

The random forest model is trained on all monthly instances up to and including the last 31st December of the training period, and the test set constitutes all input vectors with date equal to the investment date for that test period.

### 3.5.2 LSTM

The LSTM model is trained on the full sequence per stock, $sequence_{stock}$, where:

$$len(sequence_{stock}) \geq len(sequence_{LSTM})_{MIN} \tag{18}$$

$$sequence\ end\ date_{stock} \leq training\ set\ end\ date \tag{19}$$

In the preceding conditions, $len()$ is a function that returns the length of a sequence, $len(sequence_{LSTM})_{MIN}$ is the minimum number of time steps for LSTM inputs, $sequence\ end\ date_{stock}$ is the date corresponding to the last step in the stock's sequence, and $training\ set\ end\ date$ is the final 31st December in the training set. If $len(sequence_{stock}) > len(sequence_{LSTM})_{MAX}$, where $len(sequence_{LSTM})_{MAX}$ is the maximum sequence length for LSTM inputs, then a sequence is included in the training set for each full-length $sequence_{LSTM}$ that can be "fit" in $sequence_{stock}$. This is done by rolling through the time steps of $sequence_{stock}$.

The test set constitutes the full sequence per stock, $sequence_{stock}$, for every stock that fulfils:

$$sequence\ end\ date = investment\ date \tag{20}$$

If $len(sequence_{stock}) > len(sequence_{LSTM})_{MAX}$, the sequence is truncated so that the sequence starts at time step, $t = len(sequence_{stock}) - len(sequence_{LSTM})_{MAX}$, where $t = 0$ is the first time step in $sequence_{stock}$.

## 3.6 Evaluation

Traditional data science performance metrics for regression tasks, such as MSE variants (see equation 3), are not useful for evaluating the performance of models for stock selection. This is because, for profitable stock selection, only the ordering of returns for each stock is important, and not the magnitude. Therefore, a model with better MSE may achieve much worse return performance. Consequently, it is essential to evaluate model performance using metrics from the financial domain.

Because the trading strategy prescribes that each stock is bought with equal weight, the annual portfolio return for each test period, $r_t$, is calculated as:

$$r_t = \frac{1}{n} \sum_{x=1}^{n} y_n, \tag{21}$$

where $y_0, ..., y_n$ are the actual target variable (1-year return) values for the stocks that correspond to the top $n$ predicted returns, $\hat{y}$. $n$ is either 30, 50, 100 or 200 depending on the portfolio size.

The portfolio's cumulative performance from 1981 to period $t$, $p_t$, given initial capital $IC$, is calculated using the equation,

$$p_t = IC \times (1 + r_1) \times (1 + r_2) \times ... \times (1 + r_t) \tag{22}$$

$IC$ is set to the arbitrary quantity, \$100, and both $r_t$ and $p_t$ for each model are calculated and compared to the performance of the S&P 500 value-weighted index for the period 1981-2022.

In active trading, fund performance over an arbitrary number of periods is evaluated relative to some benchmark index. Therefore, it is necessary to use the mean and standard deviation of excess returns relative to some index to evaluate the return and volatility performance of a portfolio. The mean excess return, $\overline{ER}$, over $T$ periods is calculated using the equation:

$$\overline{ER} = \frac{1}{T} \sum_{t=1}^{T} ER_t, \tag{23}$$

where

$$ER_t = (r_t - r_b), \tag{24}$$

where $r_b$ is the return of the benchmark index for period $t$. The standard deviation of excess returns for the same period, $\sigma_{ER}$, is calculated using the equation:

$$\sigma_{ER} = \sqrt{\frac{1}{T-1} \sum_{t=1}^{T} (ER_t - \overline{ER})^2}, \tag{25}$$

A popular metric that uses both the mean and standard deviation of excess returns relative to some benchmark is the information ratio, $IR$, which is given by the equation:

$$IR = \frac{\overline{ER}}{\sigma_{ER}}, \tag{26}$$

It can be interpreted as the amount of excess returns, given the risk taken to achieve them, and gives prospective investors a better idea of a portfolio's ability to consistently pick good stocks.

Although this is a commonly-used ratio for evaluating portfolio performance, it assumes that stock returns follow a normal distribution and thus unfairly penalises portfolios for volatile positive excess returns. Moreover, most investment managers with long-term investment horizons do not care about the volatility of positive excess returns. Therefore, the proposed performance metric for evaluating model performance in this thesis is the Sortino-modified Information Ratio, $SIR$. It incorporates the idea of downside risk from the well-known ***Sortino Ratio***, and combines it with the Information Ratio by measuring excess returns relative to some benchmark index. Downside standard deviation of excess returns, $\sigma_{DER}$, for $T$ periods, is given by the equation:

$$\sigma_{DER} = \sqrt{\frac{1}{T-1} \sum_{t=1}^{T} \min(0, ER_t)^2}, \tag{27}$$

and $SIR$ is given as:

$$SIR = \frac{\overline{ER}}{\sigma_{DER}}. \tag{28}$$

The final evaluation method for model performance is the one-tailed Welch's $t$-test, which compares the mean yearly return for each portfolio with the mean return for the benchmark index. An F-test is used to determine whether the equal variance assumption can be made. The null hypothesis is that the average return of the model portfolio is equal to the return of the benchmark index, and the alternative hypothesis proposes that the model actually outperforms the benchmark. If the resulting $p$-value for the one-tailed Welch's $t$-test $<0.05$, the null hypothesis can be rejected with 95% confidence.

# 4 Results

## 4.1 Return Performance

### 4.1.1 Unrestricted Investment Universe

Figure 9 compares the performance of all portfolio sizes for both models against the chosen benchmark index, the S&P 500 Value-Weighted Index. Notice the y-axis for cumulative returns (views 1 and 2) uses an exponential scale. It is clear that both models significantly outperform the index, with all portfolios generating exponentially greater returns over the entire 41-year period. Views 3 and 4 plot the corresponding mean yearly returns for each model. Although they demonstrate similar outperformance with average yearly returns of 161.13% and 55.9% for the 30-stock portfolios, it reveals that both exhibit high return volatility. Furthermore, until 1990, both models fail to outperform the benchmark. This may be because the training set is smaller for these earlier years. Both models experience negative average annual returns following the dot-com crash in 2000-2002 and the global financial crisis in 2008, and the LSTM model also experiences similar underperformance during the COVID-19 pandemic. This implies that the models struggle during abnormally ***bearish*** market conditions.

A combination of economic stability, the strength of the US Dollar, and reliable monetary policy make the North American market the most popular and therefore the most efficient market globally. Therefore, it is incredibly surprising to see such consistently high returns. A closer examination of portfolio composition exposes the reason for this.

Figure 8 displays the distribution of the market capitalisation type for selected stocks in each portfolio for both the random forest and LSTM models. The majority of stocks picked by both models are in the Nano Cap or Micro Cap categories. The problem with this result is twofold. Firstly, as discussed in Section 3.2.5, although stocks with lower market capitalisation can yield much greater returns (see Figure 2), they also carry much greater risk. Investing large amounts of capital in such small stocks results in high trading costs, resulting from large bid-ask spreads and market impact. Therefore, the actual price paid for such stocks is likely to be much more than the mid-price at the time of investment. These trading costs are unpredictable and not accounted for in the model results. The second reason why such a market cap distribution is a problem is that the cumulative results in Figure 9 assume that all accumulated capital can be invested in selected stocks. The total market capitalisation of all 30 stocks selected by the random forest model in the 30 stock portfolio in 2021, the final investment date, is $1bn. The results, on the other hand, assume that the total capital, which is 2.7m times that amount, can be invested equally in those same stocks, which is obviously impossible.

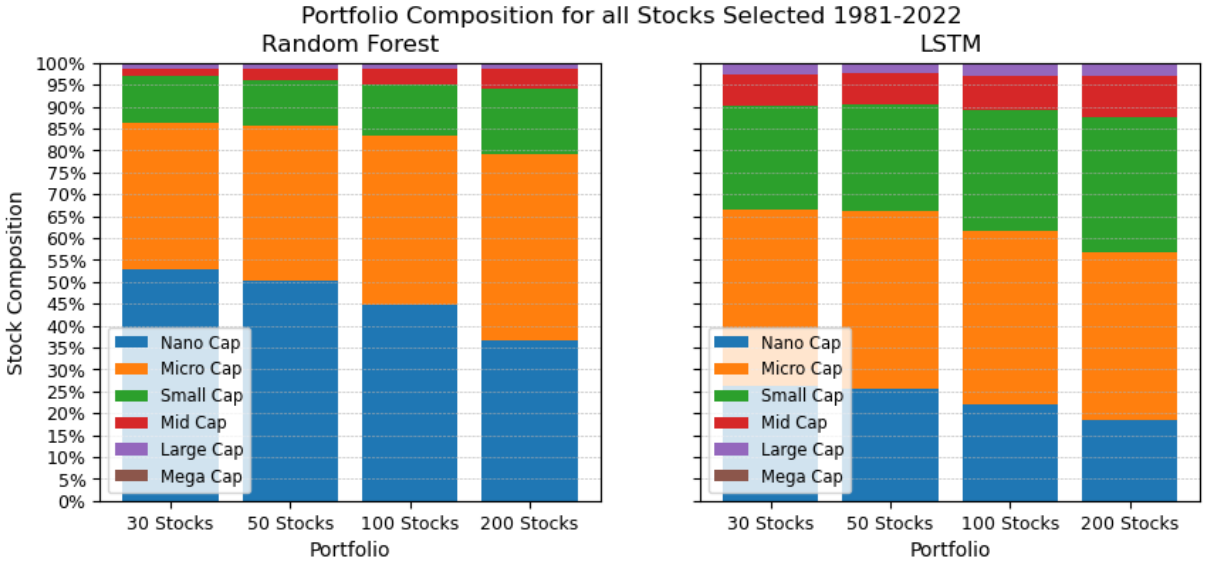


Figure 8: Stock composition for all stocks selected by each model for the entire period 1981-2022.

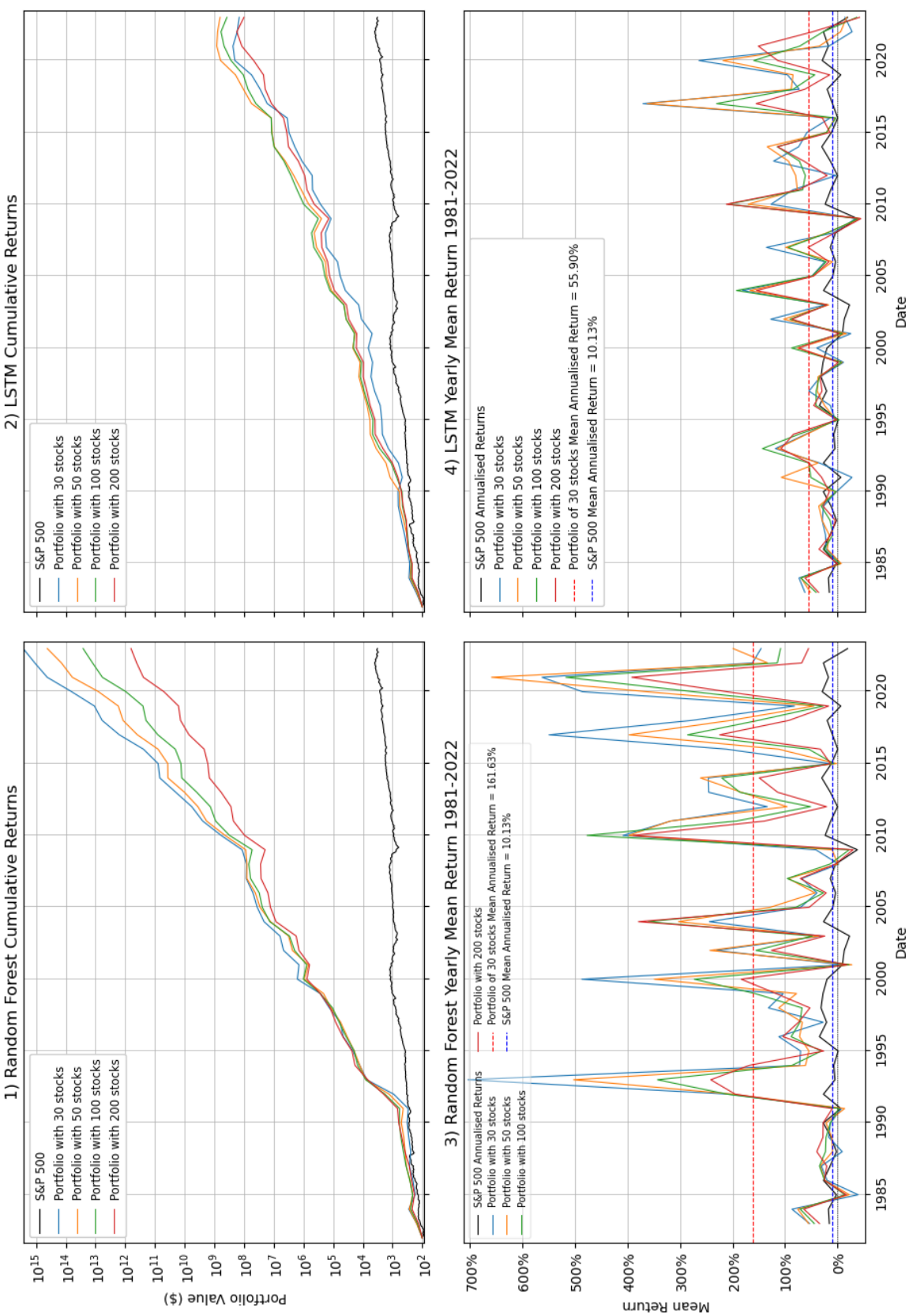


Figure 9: Model return performance (cumulative and yearly average) 1981-2022. The y-axis for cumulative returns uses an exponential scale.

| Model | Portfolio Size ($n$) | Mean Return | $\sigma$ Return | $\overline{ER}$ | $\sigma_{DER}$ | SIR | 1-T WT ($p$) |
|---|---|---|---|---|---|---|---|
| Random Forest | 30 stocks | 161.63% | 180.27% | 151.50% | 12.723% | 22.035 | 0.0000 |
| | 50 stocks | 143.40% | 156.47% | 133.26% | 7.5461% | 26.964 | 0.0000 |
| | 100 stocks | 120.54% | 130.78% | 110.41% | 5.4510% | 29.672 | 0.0000 |
| | 200 stocks | 96.033% | 106.51% | 85.899% | 6.9279% | 27.031 | 0.0000 |
| LSTM | 30 stocks | 55.896% | 78.243% | 45.763% | 16.095% | 3.9595 | 0.0004 |
| | 50 stocks | 59.573% | 73.206% | 49.440% | 11.880% | 5.8631 | 0.0001 |
| | 100 stocks | 55.212% | 60.692% | 45.079% | 11.786% | 6.5091 | 0.0000 |
| | 200 stocks | 49.001% | 53.338% | 38.867% | 10.374% | 6.5975 | 0.0000 |

Table 1: Model performance metrics on entire investment universe.

The above criticisms do not invalidate the average returns generated by both models and each still demonstrates consistent ability to select stocks that significantly outperform the market. Such recommendations would certainly be useful for venture capital and other such smaller-cap investors. Table 1 displays the performance metrics for each model. All portfolios for both models outperform the S&P 500 at the 99% significance level, and each achieve extremely high SIR scores, with the 100-stock random forest portfolio providing the greatest excess returns per unit of downside risk with an SIR score of 29.672.

Although both models demonstrate impressive ability to identify high-performing growth stocks, the liquidity problems associated with investing in such small stocks mean that a limited amount of capital can be invested into such strategies. Thus, the models were run again for a more restricted investment universe of only stocks that have a market capitalisation of Mid Cap or greater. Mid Cap stocks are generally characterised by much more liquid markets with high daily trading volume and lower transaction costs. Although this reduces the size of the investment universe by >50%, as shown in Figure 1, there should still be sufficient training data for both models.

#### 4.1.2 Restricted Investment Universe

Figure 10 displays the cumulative and average performance for both models when the investment universe is restricted to Mid Cap stocks and larger, and Figure 21 (Appendix C) displays the market capitalisation distribution. It is immediately evident that both models do not generate the same outsized returns. This is not surprising, because, as Figure 2 exhibits, investors are not compensated for liquidity risk and thus receive smaller returns on average. Furthermore, the market for larger North American stocks is famously competitive and selecting a portfolio of stocks form this universe that outperforms the S&P 500 benchmark index is unachievable for the majority of contemporary active investors [31]. Nonetheless, views 1 and 2 show that both models outperform the benchmark cumulatively over the 41-year period. Moreover, the LSTM model seems to outperform the random forest model, with the 30-stock portfolio generating more than three times more profit than the same-sized random forest portfolio.

The average yearly returns for each model depicted in views 3 and 4 reveal that, despite the cumulative outperformance, returns seem more volatile than the benchmark index, and there are significantly more periods of negative average returns than there was in Figure 9. Both models appear to mirror the return fluctuations of the benchmark index for periods 1981-1996, with both models exhibiting greater volatility than the S&P 500. The difference in performance between 2000 and 2005 explains the rift in cumulative performance. While the LSTM model greatly outperformed the benchmark during the dotcom crash, the random forest portfolios lost more than half their value. Both models struggled to select performant stocks during the global financial crisis but during the subsequent bull run of 2010, the LSTM model 30-stock portfolio generated a return of >250%, vastly outperforming the same random forest portfolio.

Table 2 displays the performance metrics for each model. As expected, the 30-stock LSTM portfolio achieves the highest mean return and SIR. However, the 1-tailed Welch's $t$-test result is insignificant at the 95% significance level, implying that the portfolio's mean annual return is not significantly higher than that of the benchmark index. However, an SIR >1 when compared with the S&P 500 index is still very impressive, and a more complex trading strategy may reduce return volatility and thus yield a statistically significant result.

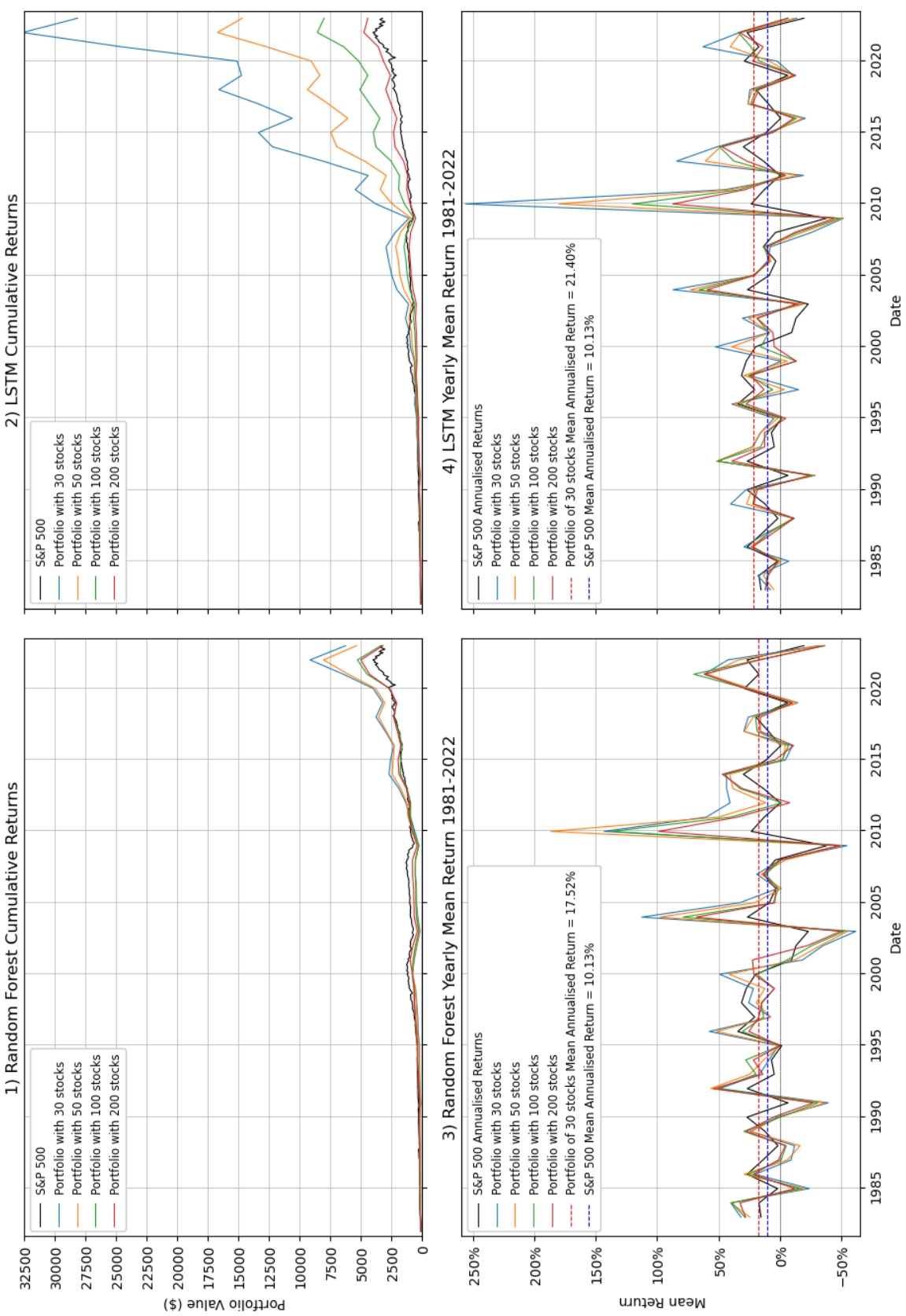


Figure 10: Model return performance (cumulative and yearly average) with Mid Cap restriction 1981-2022.

| Model | Portfolio Size ($n$) | Mean Return | $\sigma$ Return | $\overline{ER}$ | $\sigma_{DER}$ | SIR | 1-T WT ($p$) |
|---|---|---|---|---|---|---|---|
| Random Forest | 30 stocks | 17.521% | 39.476% | 7.3876% | 9.9122% | 0.7332 | 0.1388 |
| | 50 stocks | 16.638% | 40.872% | 6.5042% | 8.4902% | 0.6982 | 0.1764 |
| | 100 stocks | 13.780% | 33.643% | 3.6469% | 8.5899% | 0.4349 | 0.2691 |
| | 200 stocks | 12.728% | 28.869% | 2.5946% | 7.1192% | 0.3488 | 0.3103 |
| LSTM | 30 stocks | 21.396% | 47.324% | 11.263% | 10.278% | 1.1385 | 0.0801 |
| | 50 stocks | 17.490% | 35.772% | 7.3564% | 9.0810% | 0.8708 | 0.1200 |
| | 100 stocks | 14.295% | 27.033% | 4.1610% | 8.7125% | 0.5348 | 0.2025 |
| | 200 stocks | 11.965% | 22.626% | 1.8319% | 7.8844% | 0.2465 | 0.3385 |

Table 2: Model performance metrics on entire investment universe.

Another key observation that can be made from Table 2 is the effect of diversification on the performance of the LSTM model. The SIR for the 30-stock portfolio is nearly five times better than that for the 200-stock portfolio, which exhibits the worst performance of all portfolios for the restricted universe. The effect is not mirrored in the random forest results.

Although analysing stock returns is informative, before deploying a machine learning model to active trading, it is important to investigate model behaviour and predictions. Future market conditions are likely to change and understanding stock composition and why the model is picking certain stocks is imperative for preventing unexpected losses.

## 4.2 Feature Importance

Figure 11 presents the feature importances as determined by the random forest models before and after market cap restriction (see Appendix A for a list of variables and their formulae). The top view displays the feature importances for the model trained on the entire universe of North American listed stocks and the bottom view the same for only stocks Mid Cap and larger. As expected, market cap was the by far most important feature for predicting returns in the first instance. For the restricted universe, market cap was much less important, and instead, valuation ratios and economic indicators were the most important for predicting 1-year returns.

While this reveals the most important features for predicting 1-year returns, for a thorough comprehension and understanding of model predictions, it is essential to also compare the stock composition of portfolios selected by each model.

## 4.3 Model Stock Composition Comparison

### 4.3.1 Principal Component Analysis

Dimensionality reduction is a useful tool for visualising model classifications. Prinicipal Component Analysis (PCA), unlike $t$-SNE used in Section 3.2.3, is a *linear* dimensionality reduction technique that transforms the original data while preserving the most important information. It does this by calculating and identifying the top $k$ eigenvectors, or principal components, that explain the most variance in the high-dimensional space. Figure 12 shows the result of PCA when $k = 2$ and points are mapped onto a two-dimensional scatterplot. The variance, as a percentage of total variance, explained by the first two principal components is 43.1%. Each dot is a stock at one of the 41 investment dates and each is coloured by whether they were selected for investment or not. It is clear to see that the models pick stocks with different features. The stocks selected by the random forest model are more spread out and can be found at the left and right extremes of the projection space. On the other hand, those selected by the LSTM are more clustered in the centre, and, relative to the two directions of most variance in the dataset, certainly appear to have less extreme feature values. Figure 22 shows the same plot for the unrestricted investment universe. Although the same conclusions hold about the LSTM selections, the stocks selected by the random forest model are much more clustered at the right side of the projection space.

### 4.3.2 Comparison of Means

To gain a deeper understanding of the "type" of stocks that each model is selecting, a comparison of means is used to contrast the mean values of each feature for selected and unselected stocks, for each model. Figure 13 shows the results of this analysis for the restricted investment universe. Before separating the two classes (selected and not selected), data is scaled using Z-score normalisation to ensure that the results are interpretable and comparable across features. This analysis assumes that each feature follows a normal distribution and so a handful of less important features, including dividend yield, that did not follow a normal distribution, were excluded. Next, the difference in the means was calculated. The Student's $t$-distribution was used to calculate 95% confidence intervals around each difference.

Several key observations can be made from Figure 13. Firstly, for the majority of features, mean differences for both models share the same sign, i.e. a positive or negative difference between selected and unselected stocks. Another somewhat surprising similarity between model predictions are the tendency to pick stocks with stereotypically worse capitalisation and solvency metrics, including capitalisation ratio (debt_capital) and debt to assets ratio (debt_assets). There are two notable differences in stocks selected by the models. Firstly, although the Random Forest model selects stocks with significantly higher average book-to-market (bm) ratios, the LSTM model selects stocks with extremely high values for book-to-market on average. This difference is reinforced by the disparity between model selections for price-to-book ratio (ptb). Secondly, the Random Forest model seems to select stocks with very low profitability ratios, with eight of the ten profitability metrics exhibiting mean differences of less than -0.75$\sigma$. Although stocks selected by the LSTM tell a similar story, the differences are not as significant. Both models select stocks with a lower average market cap, but this difference is very small. Figure 23 in Appendix C displays the same analysis for the unrestricted investment universe. As expected, mean difference in market cap for both models is much greater, with both models selecting stock which are more than 0.5$\sigma$ smaller on average.

In general, Figure 13 implies that the Random Forest model tends to select stocks which are not yet profitable, have high balance sheet debt and equity relative to income, but have decent valuation metrics, including book-to-market (bm) and price-to-cash-flow (pcf). These characteristics emulate the typical growth stock archetype. This archetype is reinforced in Figure 23 and explains some of the return performance behaviour. The North American market is famous for providing access to capital to smaller-cap growth stocks, enabling them to generate extremely impressive returns over relatively short time spans during times of strong investor sentiment. This explains the significant outperformance of the Random Forest model when allowed access to Nano and

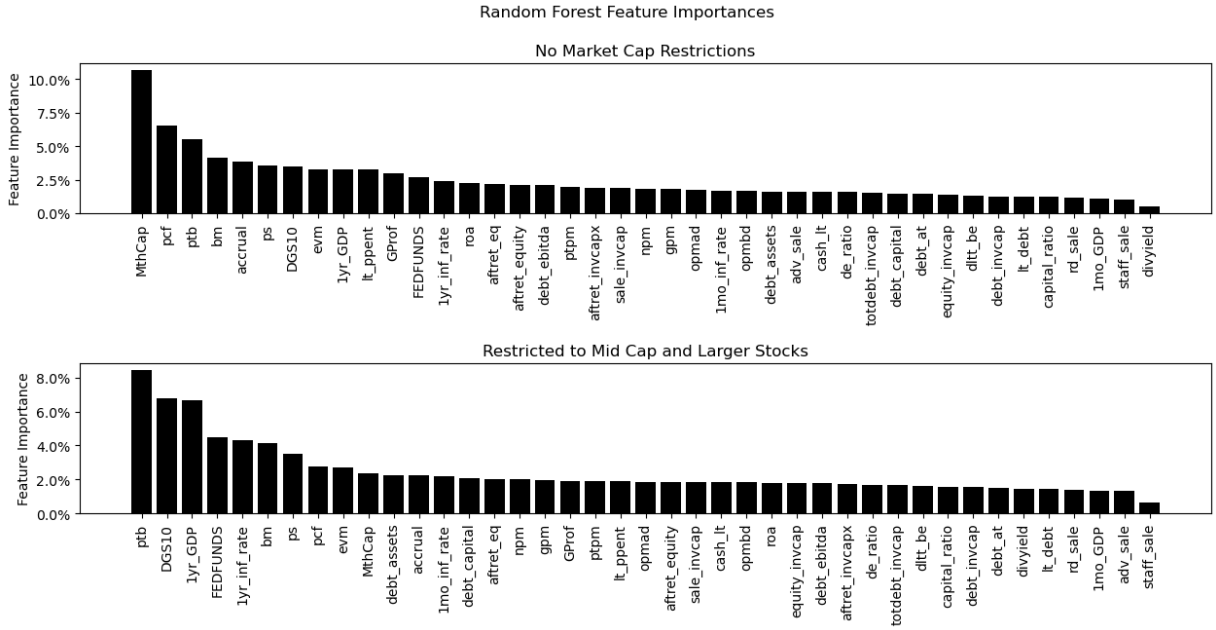


Figure 11: Ordered feature importances from the random forest model before and after market cap restrictions.

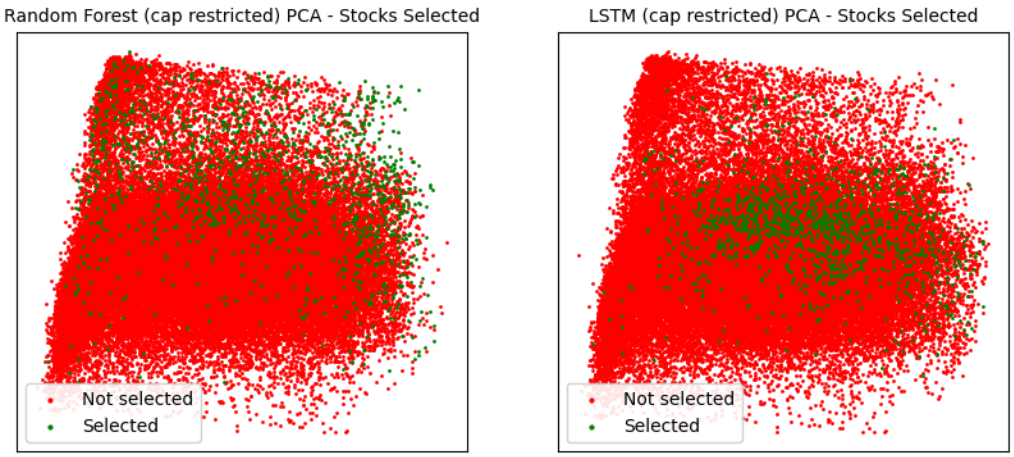


Figure 12: PCA mapping of stocks selected in the 30-stock portfolio by each model (restricted to Mid Cap stocks and larger) for the entire period 1981-2022.

Micro Cap stocks. It also explains the significant underperformance of the model during each financial crisis when restricted to Mid Cap stocks and larger. When investor confidence falls, growth stocks usually experience the greatest decline in prices.

On the other hand, the LSTM seems to select stocks that exhibit exceptional performance for the valuation ratios, and thus follows more of a value strategy. Value investors generally try to identify stocks are undervalued by the market, with the expectation that their true value will be recognised eventually and the investments will yield a profit.

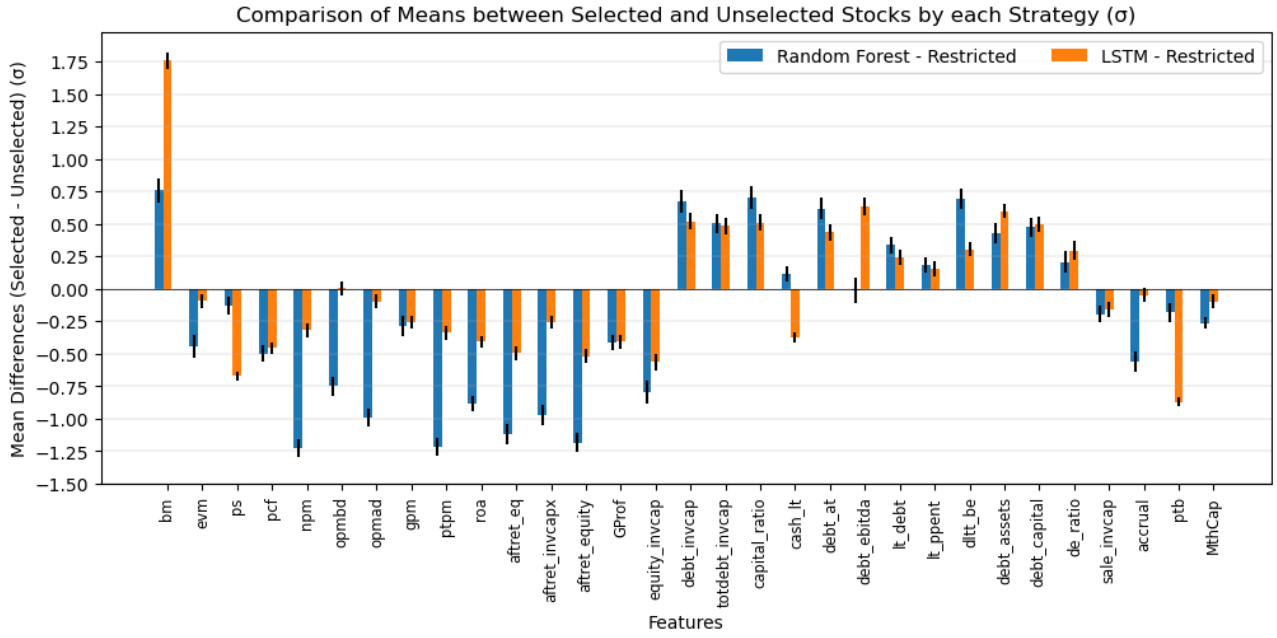


Figure 13: The mean of each feature for selected stocks minus the mean for stock not selected by the model (restricted to Mid Cap stocks and larger). The y-axis is in standard deviations of the feature for the entire investment universe. Black error lines represent a 95% confidence interval.

# 5 Conclusions and Future Work

## 5.1 Conclusions

The main contribution of this thesis is applying sequence prediction models to long term stock selection tasks for the first time. Results suggest that the LSTM model, superior in the short term stock price forecasting literature, is a viable model for long term stock selection and achieves performance that can not only outperform benchmark indices, but compete with the incumbent leading model, the Random Forest. In fact, when the investment universe was restricted to larger stocks with more efficient markets, the LSTM actually generated greater cumulative returns and achieved an SIR of 1.138 versus an SIR of 0.733 for the Random Forest model.

This thesis proposes more realistic and thorough techniques for simulating and evaluating stock selection models compared to other long-term stock selection papers. This involves choosing a long time period for model testing, removing lookahead bias, and applying models using an appropriate trading strategy. Moreover, models should be evaluated comprehensively, using comparison of means and dimensionality reduction techniques to understand the general "type" of stocks that each model selects, rather than simply visualising returns.

Applying such techniques revealed that the models employed in this thesis select different stock archetypes. PCA showed that the Random Forest model selected stocks with more extreme values for many of the financial ratios. Furthermore, the model tended to prefer growth stocks that are generally smaller, less profitable, and have high return potential. On the other hand, the LSTM selects stocks that perform well on valuation metrics, like the book-to-market ratio. Future studies might seek to discover whether this is a feature of non-parametric and parametric models generally, or whether it is idiosyncratic to this thesis. By analysing models in this way, a deeper understanding of model and strategy performance can be attained.

Model performance differences between the unrestricted and restricted investment universes demonstrated that compensation for liquidity risk is a significant factor in North American markets. Models performed significantly worse when restricted from accessing smaller stocks, implying that statistical patterns and characteristics that help predict returns are different across market cap types. Future long-term stock selection papers should consider the market capitalisation spread of their investment universe and perhaps develop individual models for each market cap type.

Finally, as Huang (2012) discovered in their analysis of Taiwanese stock markets, more diversified portfolios generally yield higher average returns but with greater return volatility. Therefore, it is important to consider this trade-off using appropriate variations of the Sharpe or Information ratio, and statistical tests like the 1-tailed Welch's $t$-test, when deciding the optimal diversification level for a proposed strategy.

## 5.2 Future Work

This thesis uses an investment universe constituting only North American listed stocks. As discussed in Section 2.2.2, global markets exhibit varying levels of maturity and efficiency, and are subject to different economic and political factors. Therefore, the results of this paper cannot be generalised to all global markets. Instead, future work might replicate this experiment for other markets to determine whether the LSTM is a viable induction model for long term stock selection generally.

A shallow LSTM model was used in this thesis, constituting just one LSTM layer. As the quantity of historical data increases, future work might explore incorporating feedforward layers as well, or, following the success of transformers and self-attentive layers in large language modelling, apply more complex deep architectures. Alternatively, hybrid models are becoming increasingly popular for short term stock price forecasting and future work could experiment with more creative hybrid solutions for selecting stocks.

Lastly, this paper employs a relatively simple trading strategy that rebalances the portfolio once annually. This results in relatively spiky results that yield low statistical significance when compared with the benchmark index. Future work might experiment with more sophisticated trading strategies that use machine learning methods to optimise when and how to rebalance portfolios. This would help smooth returns and likely improve return volatility and cumulative performance.

## 5.3 Reflection

The original project plan centred around comparing the random forest and LSTM models with a GA feature subset selection component, as proposed in Huang (2012). During modelling, this was an overly complicated

solution to what was a relatively insignificant problem. The random forest handles a large number of features with high redundancy very well and including a GA wrapper yielded insignificant improvements relative to the additional computational intensity it provided. Therefore, there was some time wasted developing and testing this component, and it was decided that using feature importances from the random forest model would be much more feasible for feature subset selection for the LSTM. If I had more thoroughly considered the mechanisms of both models during the project planning phase, then it may have saved time during execution.

The project plan did not allow enough time for data preparation, cleaning, and exploratory analysis. Despite utilising AWS Cloud to deploy increased compute power, each step in these stages took longer to execute due to the size of the dataset. For future projects, I will consider the computational intensity of workloads more when allocating time to project stages.

Finally, developing the LSTM experimental setup and scripts in PyTorch required more time and experimentation than expected. In future, when applying an unfamiliar library or more complex modelling process, it would be better to have a detailed plan for testing various hypotheses and tracking the results for each experiment.

# References


[1] E. F. Fama, "Random walks in stock market prices," *Financial Analysts Journal*, vol. 51, pp. 75–80, 1970. [Online]. Available: https://doi.org/10.2469/faj.v51.n1.1861

[2] R. J. Shiller, "Do stock prices move too much to be justified by subsequent changes in dividends?" *Cambridge: National Bureau of Economic Research*, 1981. [Online]. Available: https://www.aeaweb.org/aer/top20/71.3.421-436.pdf?mod=article_inline

[3] L. H. Pedersen, *Efficiently Inefficient: How Smart Money Invests and Market Prices Are Determined*. Princeton University Press, April 2015.

[4] R. Peachavanish, "Stock selection and trading based on cluster analysis of trend and momentum indicators," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, March 2016. [Online]. Available: https://www.iaeng.org/publication/IMECS2016/IMECS2016_pp317-321.pdf

[5] Y. Hu, K. Liu, X. Zhang, L. Su, E. Ngai, and M. Liu, "Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review," *Applied Soft Computing*, vol. 36, pp. 534–551, November 2015. [Online]. Available: https://doi.org/10.1016/j.asoc.2015.07.008

[6] D. Shah, H. Isah, and F. Zulkernine, "Stock market analysis: A review and taxonomy of prediction techniques," *International Journal of Financial Studies*, vol. 7, May 2019. [Online]. Available: https://doi.org/10.3390/ijfs7020026

[7] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLOS ONE*, July 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0180944

[8] L. D. Persio and O. Honchar, "Recurrent neural networks approach to the financial forecast of google assets," *International Journal of Mathematics and Computers in Simulation*, vol. 11, 2017. [Online]. Available: https://iris.univr.it/handle/11562/959057

[9] M. Roondiwala, H. Patel, and S. Varma, "Predicting stock prices using lstm," *International Journal of Science and Research*, vol. 6, April 2017. [Online]. Available: https://www.researchgate.net/publication/327967988_Predicting_Stock_Prices_Using_LSTM

[10] A. M. Rather, A. Agarwal, and V. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Systems with Applications*, vol. 42, pp. 3234–3241, April 2015. [Online]. Available: https://doi.org/10.1016/j.eswa.2014.12.003

[11] M. Ballings, D. V. den Poel, N. Hespeels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert Systems with Applications*, vol. 42, pp. 7046–7056, November 2015. [Online]. Available: https://doi.org/10.1016/j.eswa.2015.05.013

[12] N. Milosevic, "Equity forecast: Predicting long term stock price movement using machine learning," March 2016. [Online]. Available: https://doi.org/10.48550/arXiv.1603.00751

[13] K. jae Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Systems with Applications*, vol. 19, pp. 125–132, August 2000. [Online]. Available: https://doi.org/10.1016/S0957-4174(00)00027-0

[14] S. Asadi, E. Hadavandi, F. Mehmanpazir, and M. M. Nakhostin, "Hybridization of evolutionary levenberg–marquardt neural networks and data pre-processing for stock market prediction," *Knowledge-Based Systems*, vol. 35, pp. 245–258, November 2012. [Online]. Available: https://doi.org/10.1016/j.knosys.2012.05.003

[15] E. Hadavandi, A. Ghanbari, and S. Abbasian-Naghneh, "Developing an evolutionary neural network model for stock index forecasting," *International Conference on Intelligent Computing*, vol. 93, pp. 407–415, 2010. [Online]. Available: https://doi.org/10.1007/978-3-642-14831-6_54

[16] C.-M. Hsu, "A hybrid procedure with feature selection for resolving stock/futures price forecasting problems," *Neural Computing and Applications*, vol. 22, pp. 651–671, August 2011. [Online]. Available: https://www.researchgate.net/publication/241055123_A_hybrid_procedure_with_feature_selection_for_resolving_stockfutures_price_forecasting_problems

[17] C. Smith and Y. Jin, "Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction," *Neurocomputing*, vol. 143, pp. 302–311, November 2014. [Online]. Available: https://doi.org/10.1016/j.neucom.2014.05.062

[18] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, vol. 42, pp. 2161–2172, March 2015. [Online]. Available: https://doi.org/10.1016/j.eswa.2014.10.031

[19] A. Upadhyay, G. Bandyopadhyay, and A. Dutta, "Forecasting stock performance in indian market using multinomial logistic regression," *Journal of Business Studies Quarterly*, vol. 3, March 2012. [Online]. Available: https://www.researchgate.net/publication/265939310_Forecasting_Stock_Performance_in_Indian_Market_using_Multinomial_Logistic_Regression

[20] C.-F. Huang, "A hybrid stock selection model using genetic algorithms and support vector regression," *Applied Soft Computing*, vol. 12, pp. 807–818, February 2012. [Online]. Available: https://doi.org/10.1016/j.asoc.2011.10.009

[21] H. Yu, R. Chen, and G. Zhang, "A svm stock selection model within pca," *Procedia Computer Science*, vol. 31, pp. 406–412, 2014. [Online]. Available: https://doi.org/10.1016/j.procs.2014.05.284

[22] J. Sun, "A stock selection method based on earning yield forecast using sequence prediction models," *arxiv.org - Neural and Evolutionary Computing*, May 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1905.04842

[23] E. F. Fama and K. R. French, "The cross-section of expected returns," *The Journal of Finance*, vol. 47, June 1992.

[24] S. Seth. (2022) Market capitalization: What it is, formula for calculating it. Investopedia. [Online]. Available: https://www.investopedia.com/investing/market-capitalization-defined/

[25] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[26] A. Amidi and S. Amidi. Recurrent neural networks cheatsheet. Stanford University. [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[27] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., January 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/9.pdf

[28] C. Fjellström, "Long short-term memory neural network for financial time series," *arxiv.org - Statistical Finance*, January 2022. [Online]. Available: https://doi.org/10.1109/BigData55660.2022.10020784

[29] A. Khamis, Z. Ismail, and H. Khalid, "The effects of outliers data on neural network performance," *Journal of Applied Sciences*, vol. 5, pp. 1394–1398, January 2005. [Online]. Available: https://ui.adsabs.harvard.edu/link_gateway/2005JApSc...5.1394K/doi:10.3923/jas.2005.1394.1398

[30] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Transactions on Nuclear Science*, vol. 44, June 1997. [Online]. Available: https://doi.org/10.1109/23.589532

[31] C. Newlands and M. Marriage. (2016, October) 99% of actively managed us equity funds underperform. Financial Times. [Online]. Available: https://www.ft.com/content/e139d940-977d-11e6-a1dc-bdf38d484582

# Glossary

**backtesting** the process of evaluating a trading or investment strategy by applying it to historical data to assess its performance.

**bearish** a pessimistic or negative outlook on the future performance of financial markets or assets, indicating an expectation of declining price.

**bid-ask spread** the difference between the highest price a buyer is willing to pay (bid) and the lowest price a seller is willing to accept (ask) for a particular financial instrument, representing a cost for executing trades in the market.

**binary mask** a binary matrix where each element has a value of either 0 or 1, typically used for filtering, masking, or selecting specific features in a data array.

**bootstrap sample** a random subset of data created by randomly selecting data points from a larger dataset with replacement.

**buy-and-hold strategy** an investment approach where an investor purchases securities and holds them for an extended period, typically with the belief that they will appreciate in value over time.

**curse of dimensionality** the challenges and issues that arise in high-dimensional spaces, where data becomes sparse, distances lose meaning, and computational complexity increases significantly.

**epoch** a single iteration during training where a machine learning algorithm processes the entire training dataset.

**growth stocks** shares in companies expected to experience substantial revenue and earnings increases, often trading at higher valuations due to their potential for future expansion.

**Hadamard product** also known as element-wise multiplication, it is an operation that takes two matrices or vectors of the same dimensions and produces a new matrix or vector in which each element is the product of the corresponding elements in the input matrices or vectors.

**idiosyncratic returns** unique investment performance fluctuations unrelated to overall market trends.

**institutional investors** large organizations that pool funds to invest in various financial assets and securities on behalf of clients or stakeholders.

**investment horizon** the planned or expected length of time an investor intends to hold an investment before selling or liquidating it.

**investor sentiment** the overall mood, emotions, and opinions of investors regarding the financial markets or specific assets, which can influence their buying and selling decisions.

**liquidity risk** the potential of not being able to quickly and easily convert an asset into cash without significant loss in value.

**long-short equity strategy** an investment approach where an investor simultaneously holds long positions in stocks they expect to rise in value and short positions in stocks they anticipate will decline, aiming to profit from both upward and downward market movements.

**market capitalisation** often abbreviated to market cap, the market capitalisation is the total value of a company's outstanding shares of stock, reflecting its size in the financial market.

**market impact** the effect that a large trade or transaction has on the supply and demand for a particular asset, leading to price fluctuations due to the significant buying or selling activity.

**multicollinearity** a statistical issue where two or more independent variables in a regression model are highly correlated, making it difficult to isolate their individual effects on the dependent variable.

**mutual fund** a professionally managed investment vehicle that pools money from multiple investors to invest in a diversified portfolio of stocks, bonds, or other securities.

**Occam's Razor** In machine learning, Occam's Razor suggests that simpler models, with fewer parameters and complexity, are preferred if they achieve comparable performance to more complex models.

**portfolio diversification** the practice of spreading investments across various assets or asset classes to reduce risk by not relying heavily on a single investment.

**retail investors** individual investors who buy and sell financial securities for personal investment purposes rather than on behalf of institutions or organizations.

**robust statistics** analytical methods that are resilient to the influence of outliers or deviations from standard statistical assumptions, providing stable and reliable results.

**sequence prediction model** a type of machine learning model that takes a sequence of input data and generates a sequence of output predictions, often used in tasks such as natural language processing and time series forecasting.

**short-selling** a trading strategy where an investor borrows and sells a security they don't own, with the intention of buying it back at a lower price to profit from a price decline.

**Sortino Ratio** a financial metric that evaluates the risk-adjusted return of an investment by focusing on downside volatility, measuring the return per unit of downside risk.

**trading costs** the expenses associated with buying and selling financial assets, including commissions, spreads, and fees.

**value stocks** shares in companies considered undervalued by the market, typically trading at lower price-to-earnings ratios, making them attractive to investors seeking bargains.

**venture capital** form of private equity funding that investors provide to startups and small businesses with high growth potential in exchange for equity ownership.

# A Financial Ratios

| Variable Name | Financial Ratio | Category | Formula |
|---|---|---|---|
| capital_ratio | Capitalisation Ratio | Capitalisation | Total Long-term Debt as a fraction of the sum of Total Long-term Debt, Common/Ordinary Equity and Preferred Stock. |
| equity_invcap | Common Equity/Invested Capital | Capitalisation | Common Equity as a fraction of Invested Capital |
| debt_invcap | Long-term Debt/Invested Capital | Capitalisation | Long-term Debt as a fraction of Invested Capital |
| totdebt_invcap | Total Debt/Invested Capital | Capitalisation | Total Debt (Long-term and Current) as a fraction of Invested Capital |
| at_turn | Asset Turnover | Efficiency | Sales as a fraction of the average Total Assets based on the most recent two periods |
| inv_turn | Inventory Turnover | Efficiency | COGS as a fraction of the average Inventories based on the most recent two periods |
| pay_turn | Payables Turnover | Efficiency | COGS and change in Inventories as a fraction of the average of Accounts Payable based on the most recent two periods |
| rect_turn | Receivables Turnover | Efficiency | Sales as a fraction of the average of Accounts Receivables based on the most recent two periods |
| sale_equity | Sales/Stockholders Equity | Efficiency | Sales per dollar of total Stockholders' Equity |
| sale_invcap | Sales/Invested Capital | Efficiency | Sales per dollar of Invested Capital |
| sale_nwc | Sales/Working Capital | Efficiency | Sales per dollar of Working Capital, defined as difference between Current Assets and Current Liabilities |
| invt_act | Inventory/Current Assets | Financial Soundness | Inventories as a fraction of Current Assets |
| rect_act | Receivables/Current Assets | Financial Soundness | Accounts Receivables as a fraction of Current Asset |
| fcf_ocf | Free Cash Flow/Operating Cash Flow | Financial Soundness | Free Cash Flow as a fraction of Operating Cash Flow, where Free Cash Flow is defined as the difference between Operating Cash Flow and Capital Expenditures |
| ocf_lct | Operating CF/Current Liabilities | Financial Soundness | Operating Cash Flow as a fraction of Current Liabilities |
| cash_debt | Cash Flow/Total Debt | Financial Soundness | Operating Cash Flow as a fraction of Total Debt |
| cash_lt | Cash Balance/Total Liabilities | Financial Soundness | Cash Balance as a fraction of Total Liabilities |
| cfm | Cash Flow Margin | Financial Soundness | Income before Extraordinary Items and Depreciation as a fraction of Sales |
| short_debt | Short-Term Debt/Total Debt | Financial Soundness | Short-term Debt as a fraction of Total Debt |

| | | | |
|---|---|---|---|
| profit_lct | Profit Before Depreciation/Current Liabilities | Financial Soundness | Operating Income before depreciaton and amortisation as a fraction of Current Liabilities |
| curr_debt | Current Liabilities/Total Liabilities | Financial Soundness | Current Liabilities as a fraction of Total Liabilities |
| debt_ebitda | Total Debt/EBITDA | Financial Soundness | Gross Debt as a fraction of EBITDA |
| dltt_be | Long-term Debt/Book Equity | Financial Soundness | Long-term Debt to Book Equity |
| int_debt | Interest/Average Long-term Debt | Financial Soundness | Interest as a fraction of average Long-term debt based on most recent two periods |
| int_totdebt | Interest/Average Total Debt | Financial Soundness | Interest as a fraction of average Total Debt based on most recent two periods |
| lt_debt | Long-term Debt/Total Liabilities | Financial Soundness | Long-term Debt as a fraction of Total Liabilities |
| lt_ppent | Total Liabilities/Total Tangible Assets | Financial Soundness | Total Liabilities to Total Tangible Assets |
| cash_conversion | Cash Conversion Cycle (Days) | Liquidity | Inventories per daily COGS plus Account Receivables per daily Sales minus Account Payables per daily COGS |
| cash_ratio | Cash Ratio | Liquidity | Cash and Short-term Investments as a fraction of Current Liabilities |
| curr_ratio | Current Ratio | Liquidity | Current Assets as a fraction of Current Liabilities |
| quick_ratio | Quick Ratio (Acid Test) | Liquidity | Quick Ratio: Current Assets net of Inventories as a fraction of Current Liabilities |
| accrual | Accruals/Average Assets | Other | Accruals as a fraction of average Total Assets based on most recent two periods |
| rd_sale | Research and Development/Sales | Other | Research and Development expenses as a fraction of Sales |
| adv_sale | Avertising Expenses/Sales | Other | Advertising Expenses as a fraction of Sales |
| staff_sale | Labor Expenses/Sales | Other | Labor Expenses as a fraction of Sales |
| efftax | Effective Tax Rate | Profitability | Income Tax as a fraction of Pretax Income |
| gprof | Gross Profit/Total Assets | Profitability | Gross Profitability as a fraction of Total Assets |
| aftret_eq | After-tax Return on Average Common Equity | Profitability | Net Income as a fraction of average of Common Equity based on most recent two periods |
| aftret_equity | After-tax Return on Total Stockholders' Equity | Profitability | Net Income as a fraction of average of Total Shareholders' Equity based on most recent two periods |
| aftret_invcapx | After-tax Return on Invested Capital | Profitability | Net Income plus Interest Expenses as a fraction of Invested Capital |
| gpm | Gross Profit Margin | Profitability | Gross Profit as a fraction of Sales |
| npm | Net Profit Margin | Profitability | Net Income as a fraction of Sales |

| | | | |
|---|---|---|---|
| opmad | Operating Profit Margin After Depreciation | Profitability | Operating Income After Depreciation as a fraction of Sales |
| opmbd | Operating Profit Margin Before Depreciation | Profitability | Operating Income Before Depreciation as a fraction of Sales |
| pretret_earnat | Pre-tax Return on Total Earning Assets | Profitability | Operating Income After Depreciation as a fraction of average Total Earnings Assets (TEA) based on most recent two periods, where TEA is defined as the sum of Property Plant and Equipment and Current Assets |
| pretret_noa | Pre-tax return on Net Operating Assets | Profitability | Operating Income After Depreciation as a fraction of average Net Operating Assets (NOA) based on most recent two periods, where NOA is defined as the sum of Property Plant and Equipment and Current Assets minus Current Liabilities |
| ptpm | Pre-tax Profit Margin | Profitability | Pretax Income as a fraction of Sales |
| roa | Return on Assets | Profitability | Operating Income Before Depreciation as a fraction of average Total Assets based on most recent two periods |
| roce | Return on Capital Employed | Profitability | Earnings Before Interest and Taxes as a fraction of average Capital Employed based on most recent two periods, where Capital Employed is the sum of Debt in Long-term and Current Liabilities and Common/Ordinary Equity |
| roe | Return on Equity | Profitability | Net Income as a fraction of average Book Equity based on most recent two periods, where Book Equity is defined as the sum of Total Parent Stockholders' Equity and Deferred Taxes and Investment Tax Credit |
| de_ratio | Total Debt/Equity | Solvency | Total Liabilities to Shareholders' Equity (common and preferred) |
| debt_assets | Total Debt/Total Assets | Solvency | Total Debt as a fraction of Total Assets |
| debt_at | Total Debt/Total Assets | Solvency | Total Liabilities as a fraction of Total Assets |
| debt_capital | Total Debt/Capital | Solvency | Total Debt as a fraction of Total Capital, where Total Debt is defined as the sum of Accounts Payable and Total Debt in Current and Long-term Liabilities, and Total Capital is defined as the sum of Total Debt and Total Equity (common and preferred) |
| intcov | After-tax Interest Coverage | Solvency | Multiple of After-tax Income to Interest and Related Expenses |
| intcov_ratio | Interest Coverage Ratio | Solvency | Multiple of Earnings Before Interest and Taxes to Interest and Related Expenses |
| dpr | Dividend Payout Ratio | Valuation | Dividends as a fraction of Income Before Extra Items |
| PEG_trailing | Trailing P/E to Growth (PEG) ratio | Valuation | Price-to-Earnings, excl. Extraordinary Items (diluted) to 3-Year pas EPS Growth |
| bm | Book/Market | Valuation | Book Value of Equity as a fraction of Market Value of Equity |
| CAPEI | Shillers Cyclically Adjusted P/E Ratio | Valuation | Multiple of Market Value of Equity to 5-year moving average of Net Income |
| divyield | Dividend Yield | Valuation | Indicated Dividend Rate as a fraction of Price |

| | | | |
|---|---|---|---|
| evm | Enterprise Value Multiple | Valuation | Multiple of Enterprise Value to EBITDA |
| pcf | Price/Cash flow | Valuation | Multiple of Market Value of Equity to Net Cash Flow from Operating Activities |
| pe_exi | P/E (Diluted, Excl. EI) | Valuation | Price-to-Earnings, excl. Extraordinary Items (diluted) |
| pe_inc | P/E (Diluted, Incl. EI) | Valuation | Price-to-Earnings, incl. Extraordinary Items (diluted) |
| pe_op_basic | Price/Operating Earnings (Basic, Excl. EI) | Valuation | Price to Operating EPS, excl. Extraordinary Items (Basic) |
| pe_op_dil | Price/Operating Earnings (Diluted, Excl. EI) | Valuation | Price to Operating EPS, excl. Extraordinary Items (Diluted) |
| ps | Price/Sales | Valuation | Multiple of Market Value of Equity to Sales |
| ptb | Price/Book | Valuation | Multiple of Market Value of Equity to Book Value of Equity |

# B Exploratoratory Data Analysis Visualisations

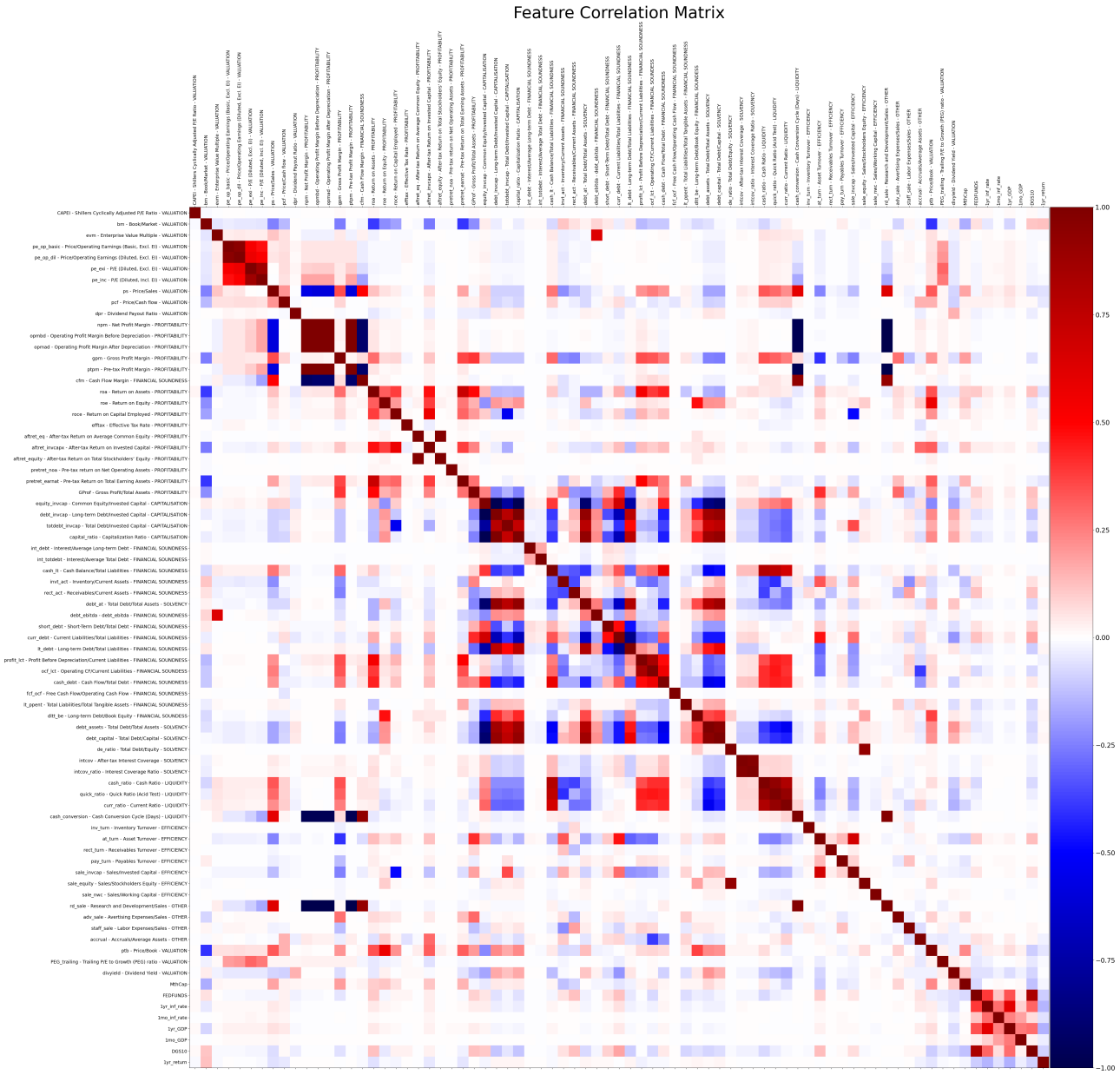


Figure 14: Input feature correlation matrix.

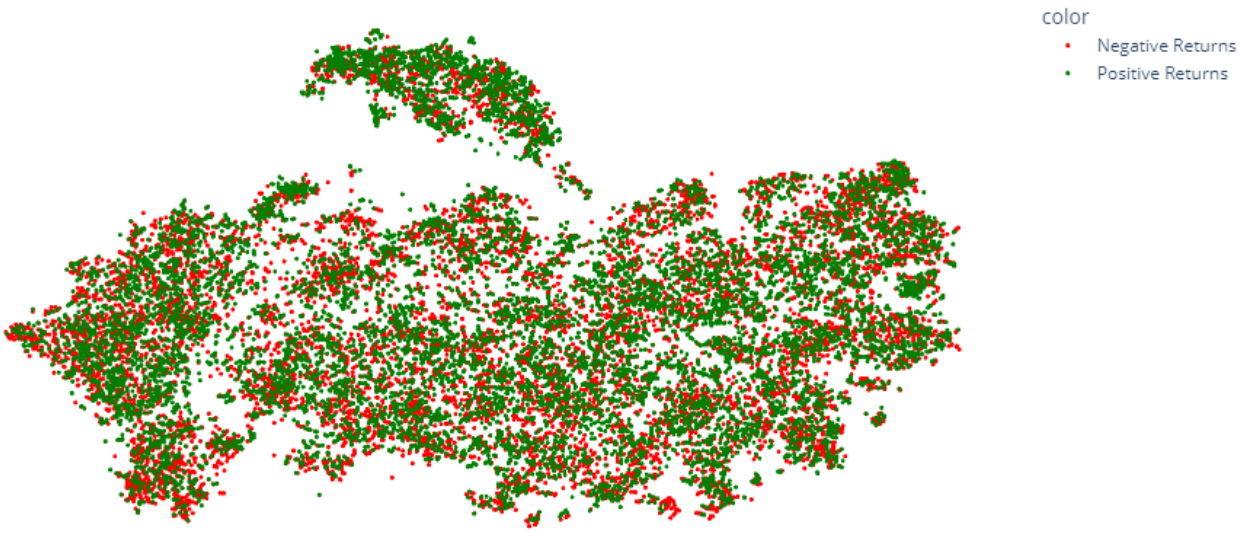


Figure 15: Results of $t$-SNE on Financial Ratios.

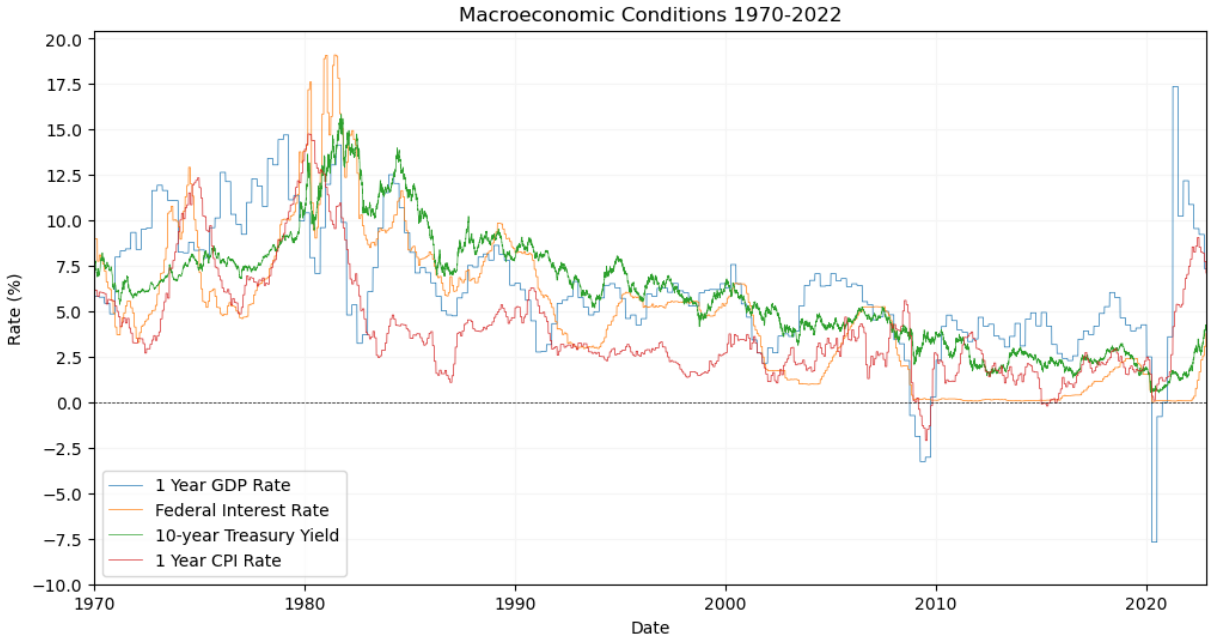


Figure 16: Macroeconomic indicators 1970-2022.


Figure 17: Median Market Capitalisation 1970-2022 (000s).

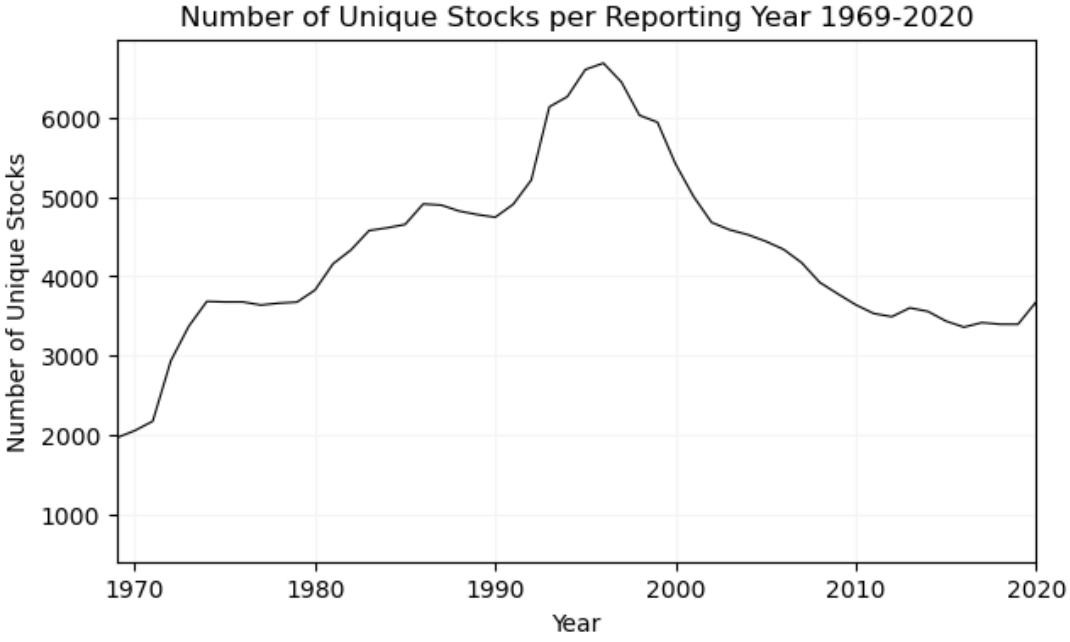


Figure 18: Number of unique stocks for reporting years 1969-2020.

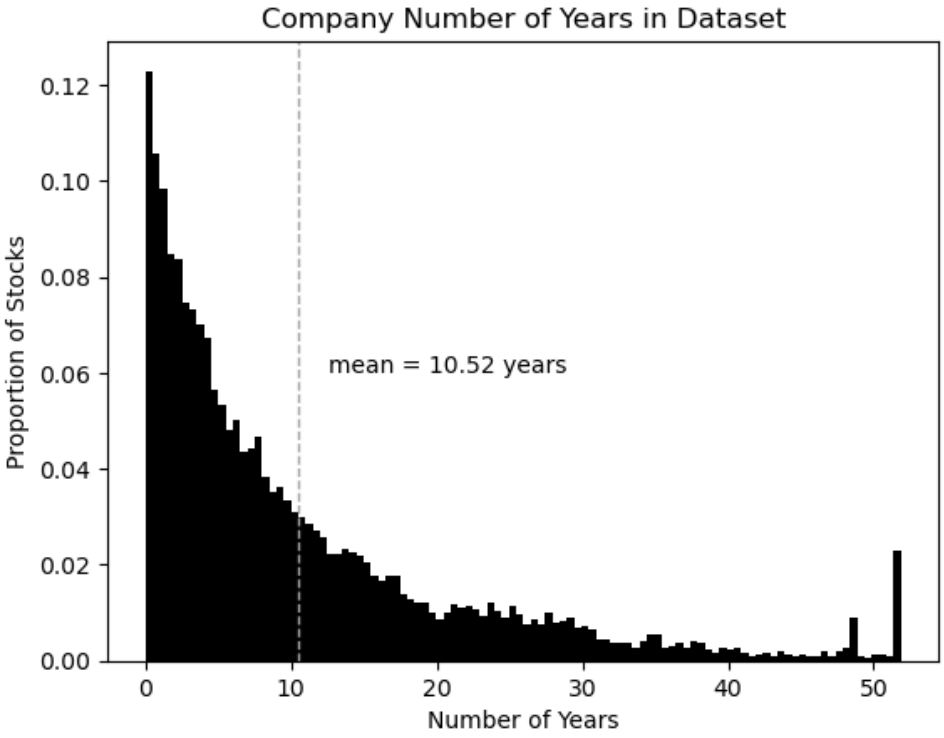


Figure 19: Number of years each company exists for in the dataset. Each bin represents half a year.

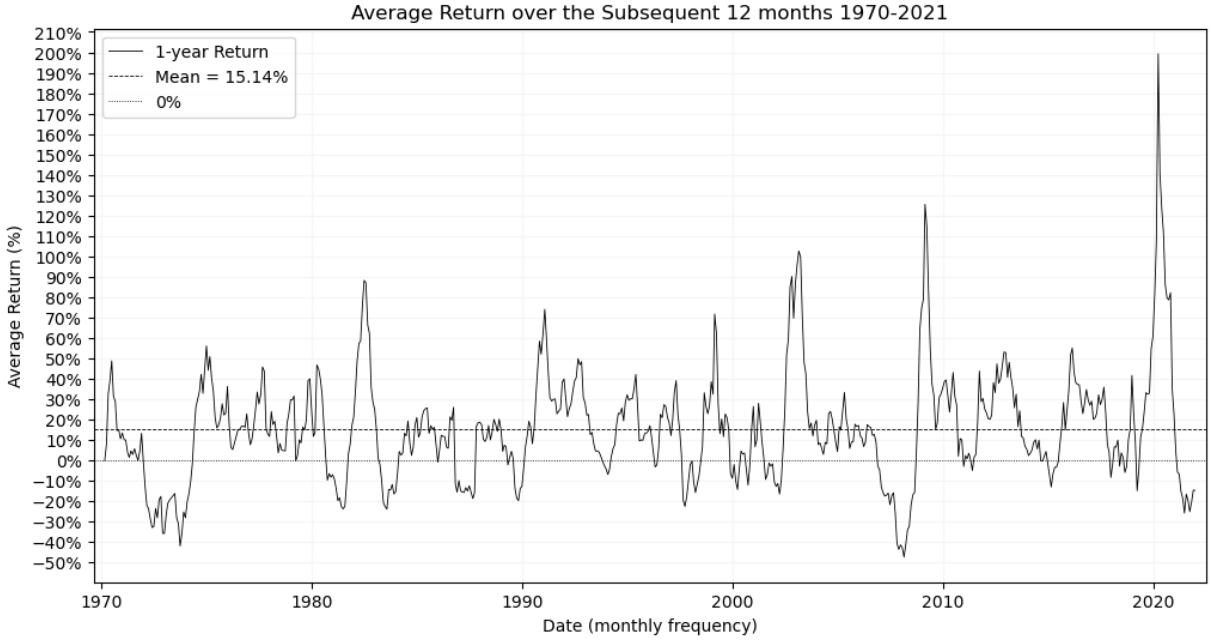


Figure 20: Target variable (1-year return) 1970-2021.

# C Results Visualisations

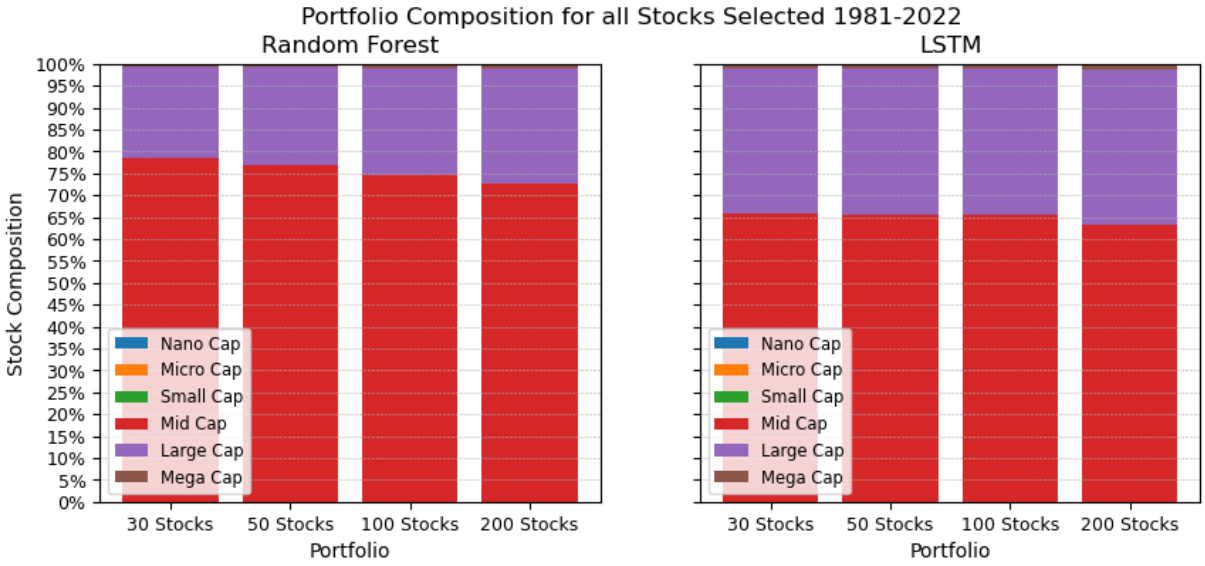


Figure 21: Stock composition for all stocks selected by each model for the entire period 1981-2022 when restricted to stocks with a market cap of Mid Cap or larger.

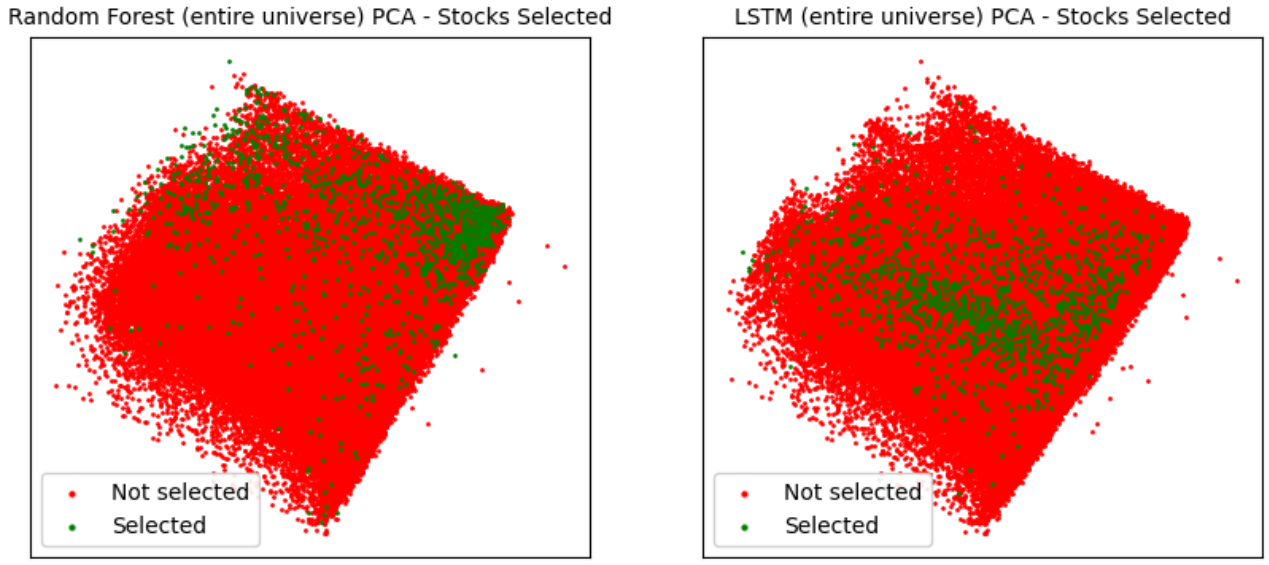


Figure 22: PCA mapping of stocks selected in the 30-stock portfolio by each model (no market cap restriction) for the entire period 1981-2022.

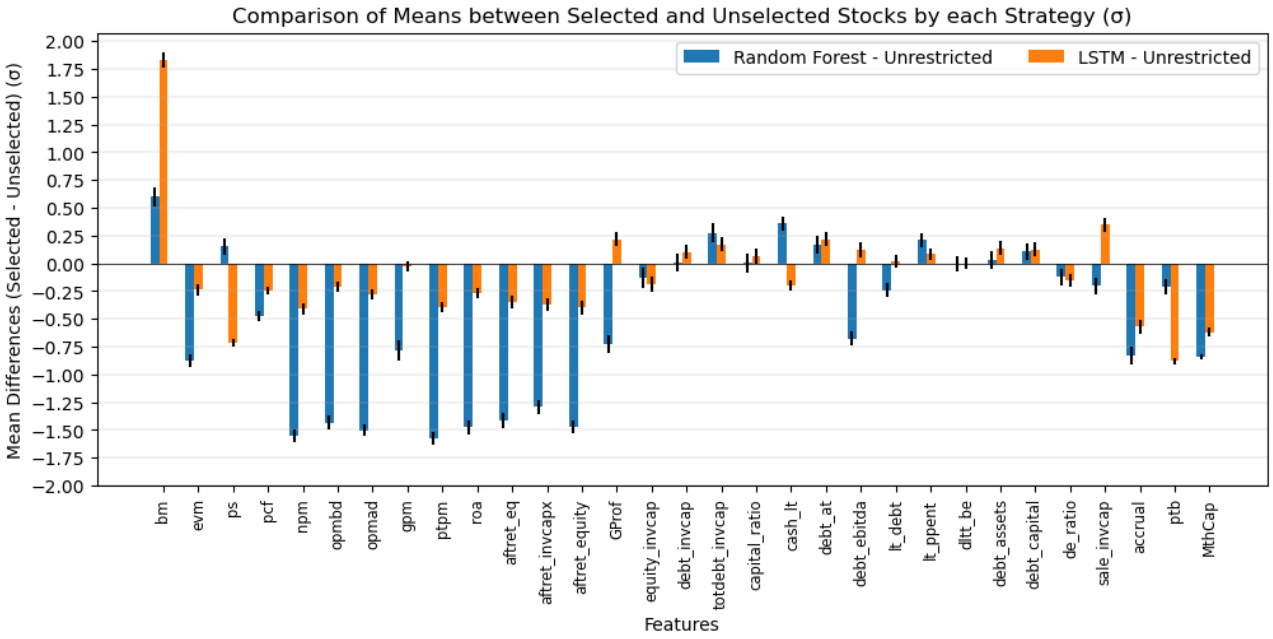


Figure 23: The mean of each feature for selected stocks minus the mean for stock not selected by the model (no market cap restriction). The y-axis is in standard deviations of the feature for the entire investment universe. Black error lines represent a 95% confidence interval.