University of Bath

PHD

Ensemble Machine Learning for Individual Stock Investment

Wichaidit, Siriphan

Award date:
2021

Awarding institution:
University of Bath

Link to publication

Alternative formats
If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 21. Oct. 2021

Ensemble Machine Learning for Individual Stock Investment

submitted by

Siriphan Wichaidit

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Computer Science

September 2019

COPYRIGHT

Attention is drawn to the fact that copyright of this thesis rests with the author. A copy of this thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that they must not copy it or use material from it except as permitted by law or with the consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation with effect from................ (date)

Signed on behalf of the Faculty of Science ....

Acknowledgements

I would like to thank my supervisor, Dr Marina De Vos, for her continuous support, guidance, and engagement throughout this thesis. I also would like to thank my co-supervisors, Dr Julian A Padget and Dr Tom Samuel Fincham Haines, for their support and advice. Without the support and guidance along the way from all of my supervisors, this thesis would not exist.

Beside my supervisors, I wish to express my deepest gratitude to Dr Neill Campbell for taking the time to be my confirmation examiner. During my confirmation exam he encouraged me to do my best.

I am grateful to the Thai Ministry of Science and Technology that is under the Office of Civil Service Commissions for awarding me the opportunity to pursue my doctorate overseas through their scholarship program.

Finally, I especially want to express my biggest thanks to my beloved husband, Mr. Kieran Torbuck, for his love and continuous support and help throughout the long process of my research. I also would like to thank my brother, Mr.Sornpon Wichaidit, and my parents for all their help and for being so supportive. I also would like to thanks my friends for always being there for me and supporting me during the completion of this thesis.

Summary

Stock market activity cycles through periods of trends and fluctuations due to external economic factors and the psychology of participants. Many stock prediction models exist for predicting prices, trends and volatility. However, models focusing on individual or few prediction methods suffer from a lack of adaptability, meaning they perform well at specific stages of the market cycle rather than over the whole range. A prediction model developed and tested during a period of strong growth may perform well under these conditions, but fail during market downturns.

It is desirable that a prediction model adapts to new circumstances so investors can profit across the entire market cycle. For this reason and because of the absence of adaptive models in the literature, this research has developed a dynamic stock investment system that combines the intelligence of multiple predictors using a scoring system to give more weight to predictions that have performed best under recent market conditions and a filtering system to identify the most potentially profitable trades, thereby effectively adapting to market behaviour.

Differently to other research in this area, the performance of the new system is not evaluated based on the accuracy of predictions, but primarily by investment metrics such as profit, drawdown and the Sharpe Ratio, the latter two of which also account for the risk of the system.

The experimental results show that our model works effectively on more than 100 stocks from the UK, US, Chinese and Singaporean markets. These stocks come from more than 10 different market sectors covering a wide-range of market conditions during the testing periods. We concluded that our model shows an excellent capability of handling predictions in fluctuated situations and is effective regardless of the characteristics of the stock data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There are a lot of factors which affect stock prices, such as rates of inflation and interest, the economic outlook, political and economic turmoil etc. These make stock price prediction extremely challenging. Most investors attempt this by analysing the time-series data of stock prices. For this, there are two main methods: *Fundamental* and *Technical* analysis.

Fundamental analysis focuses on investigating the statistics of macroeconomic data such as money supply, interest rates, currency strength, inflation and deflation rates, daily news events and company accounts in order to evaluate the real value of the company. In fundamental analysis, investors are interested in the high-value companies and may try to buy their shares when the price is lower than the perceived value of a company's assets or future cash flow in order to make a profit when the share price increases.

On the other hand, technical analysis is concerned with historical financial time-series data. The prediction of the direction of stock prices is made by identifying patterns or trends within the time-series data and seeing how they correlate with patterns that have emerged in the past. These patterns are shown in charts of prices and volumes. In technical analysis, investers try to make profit whether share price increases or decreases.

Predicting future prices in a stock market is complicated due to the continued fluctuation of prices, which are affected by many factors. However, historical data has shown that there are repetitive, recognisable patterns and relationships within markets that it may be possible to use to predict future stock prices. As such, there are many existing methods available to predict stock prices.

There are two main areas of research into stock prediction. In the first, research is mainly focused on individual stock information [102]. Approaches focus mainly on the history of a stock's price, including financial indicators which are calculated from historical prices. The second takes international stock market linkages into account [86]. Stocks and markets are investigated in order to find relationships in the price movements between them that can be used to predict future price movements. Investigating such relationships is not just limited to predicting the direction of price movements, as it can also applied to a portfolio management model in order to create diversification in a portfolio. Diversification is an important aspect in reducing portfolio risk [85], since too much exposure to any geographical region, market, sector or security could result in large losses. To create a diversified portfolio, an investor must select uncorrelated stocks in the hope that when a macroeconomic event causes a held stock's price to drop, the prices of other stocks will not be affected, therefore minimising losses for the portfolio as a whole.

Many stock prediction models exist for predicting prices, trends and volatility. Each model makes predictions based on one or a combination of indicators, such as moving averages or relative strength, and the use of machine learning techniques, for example neural networks or genetic algorithms. However, models focusing on individual or few prediction methods suffer from a lack of adaptability, meaning they are often limited to performing well at a specific stage of the market cycle rather than over the whole range.

Much like stock prices, market activity also fluctuates due to external economic and psychological factors such as recessions and their accompanying pessimism. A prediction model developed and tested during a period of economic growth and investor optimism may perform well under these conditions, but fail during inevitable market downturns or when the market is ranging (fluctuating with no clearly-defined trend). Examples of this phenomenon include the k-Nearest Neighbour model, which only performed well during market uptrends [5] and the neural network model [128], which also only performed well when certain patterns appeared in the data.

It is desirable that a prediction model adapts to new circumstances so investors can profit across the entire market cycle. For this reason and because of the absence of adaptive models in the literature, the research direction proposed here is to develop a new, adaptive model that focuses on combining the intelligence of several predictors using a scoring system to give more weight to the predictions that are most appropriate given recent market behaviour and a filtering system to identify the most potentially profitable trades.

From the experimental results, we find that multiple predictors which are trained on different pieces of data work better than a single predictor. However, to make these predictors work together effectively we must have a control system to manage switching between predictors. This research proposes a new predictor selection technique called a scoring system. The model adopts the reward and punishment technique from reinforcement learning, giving or taking away points according to the predictors' past performance. We initially use only the cumulative score as the criteria to select predictors. However, only using the score is not enough, as the predictors which work well build a high cumulative score which takes too much time to decrease when market conditions change, meaning our system cannot switch to a new predictor quickly enough to profit and instead loses money. Therefore, we introduced a second layer to the scoring system, which is not only interested in the cumulative scores, but also other features such as patterns in the scores. This two-layer scoring system performs very well across a range of stocks and market sectors as can be seen in our results from over 100 stocks in chapter six. Besides the scoring system, we also introduce an optimisation method to filter undesirable or weak signals from the scoring system. This makes the system able to be used by individual investors with limited amounts of funds, which is our goal.

This research aims to benefit individual investors for the following reasons: 1) We would like to support traders with limited funds to trade responsibly. We support the idea that everyone should be given the opportunity to make money, even with limited capital to start with. Financial markets are a place where that can be achieved and, as we will discuss shortly, are now accessible to almost everyone. 2) We would like to be able to simulate our trading system as realistically as possible. We realise than designing parameters to mimic institutional trading is difficult for outsiders with no experience. We also relish the challenge of designing a successful trading system despite the limits imposed by focusing on retail traders, such as not being able to profit from a declining share price. Positive results despite missing half of trading opportunities are a vote of confidence in our system, and it can be inferred that profits would be even bigger if short (profiting from decline) trades were included too. 3) We recognise that online trading platforms have been a growing industry for the last two decades or so. Nowadays, there are many choices of online trading platforms and applications which support people who trade from home, allowing people to learn how to trade in their own time and on the back of their own interest. Therefore, we decided to build a system that individual people could realistically follow, people who can connect with an online broker to buy and sell shares but for whom trading may not be a full-time occupation. 4) Selecting individual traders who can only profit from a share price rising as our focus confers a degree of indirect risk control. We did not include in our system elements of trading that might be categorised as risk management or money management. Examples of these elements include decisions about what percentage of one's capital to invest per trade and a predetermined loss at which point shares will be sold (a stop-loss). It is sometimes argued that a blunt-instrument, rules-based approach is safer for this than predictive models. But we did give some thought to risk control in our selection of individual investors, as seen in the following example. While our system recommends selling shares in anticipation of a decline, it is possible that the signal may come too late or not at all in reality. By only being able to buy shares and not short-sell them, the worst possible outcome is if the share price declines to zero and the money invested in that share is lost. But, importantly, the loss is capped, it cannot go beyond the amount invested if the share hits zero. However, if investors can short-sell shares, the price can go up an unlimited amount, potentially causing a loss beyond what they planned to invest if they are unable to exit the position quickly enough. After carefully considering the reasons mentioned above, we decided to build a trading system aimed at individual traders.

We believe our selection of individual investors adds value to this research since trading systems focused on or usable for individual retail investors have been largely absent from the research literature. However, with the growth of online trading platforms and even mobile applications, such traders may be coming to represent a larger share of participants in the market and we might expect some portion of these to turn to computational research in order to gain an edge that will generate consistent profits. We hope and expect to see more research applicable to individual investors in the future. Moreover, our research demonstrates that increasing the sophistication of machine learning algorithms is not the only way to improve their performance on the stock prediction problem. Counter-intuitively, we find that simpler machine learning approaches can work well when several are combined and managed effectively, though there is room in our model to include more complex algorithms too. In this research we review the efficient market hypothesis, discovering that as markets become increasingly efficient, techniques for prediction become obsolete as other market participants expect people to trade using them. The investment of resources into developing a complex new machine learning approach could be wasted if market efficiency diminishes its predictive value. Although this may not be permanent, it could leave one unable to profit and unable to know when their predictor might work again. A system such as ours is necessary in the long term, as it removes signals from out-of-favour predictors and selects better ones until they change places, aiming to exploit short-lived inefficiencies for consistent profits. The core idea of this research is this effective management of multiple machine learning algorithms, which we achieve with our novel two-layer scoring system. We believe this is a strong contribution to the study of machine learning as applied to stock price prediction, as the existing ensemble approaches we reviewed included fewer predictors and did not attempt to select the best predictor for a specific stock and time. Instead they tended to use the output from one algorithm as the input for the other which would make the prediction. Although we selected individual investors as our focus, in principle our scoring-system approach can be readily adapted for institutional trading. With the ability to short-sell and more money to trade, it is possible the results for institutional traders may be even better than the individual trader results reported here. We consider this adaptability a strength of our research.

This report begins with the financial and machine learning backgrounds in chapters two and three. Chapter two, financial background, discusses fundamental topics in finance related to our research, while chapter three reviews multiple related machine learning areas, such as machine learning algorithms used in the stock market, ensemble models, prediction and common frameworks of trading systems using machine learning. Chapter four describes the full design of our machine learning ensemble trading system. Our system follows six main steps which will be explained in this chapter and referred to in the following chapters for a better understanding of our experimental design. Chapter five discusses a number of experiments we conducted in order to obtain the complete system as described in chapter three. There are four main experiments in this chapter. The first three experiments relate to the main design of our system and will be discussed in order, while the last experiment is about small adjustments of our system which do not effect the main design but can improve the results. Chapter six shows our system's performance. In this chapter our system will be tested on various stocks from different markets and business sectors. The results will be compared to the buy & hold and the market benchmark. Then, some of the results will be compared to other trading systems. To compare our results, we mainly consider the Sharpe Ratio as this value accounts for both profit and risk. Next to the Sharpe Ratio, we consider profit as this is the main desire for investors. Please note that in this research we do not compare accuracy as a main criteria. We found from our experiments that accuracy does not lead to profit since in finance, it is not enough to get the direction correct only, but to get it correct on high-profit days and to make sure losses from incorrect predictions are kept small. The details of this issue will be discussed in chapter five, together with the real trading examples. Finally, in the last chapter, we discuss conclusions and recommendations for future work.

Chapter 2

Financial Background

This chapter will provide the reader with some brief background information about financial markets, including definitions of terminology that will be used throughout the dissertation.

As with material goods such as clothes and books, stocks and other types of financial instruments are traded on *Electronic Markets* where buyers and sellers can execute transactions on computers over the internet. The most well-known type of financial markets is probably the stock market. The stocks are released to the market by companies that wish to raise capital, also known as an IPO, and following this, market participants such as private investors, hedge funds, banks and, more recently, computers acting on behalf of market participants, can trade these stocks among each other. During these secondary sales of the stocks, participants aim to make a profit by buying the stocks and waiting for an increased demand in a particular stock before selling. Alternatively, they may profit by borrowing the stock to sell it at a high price, then awaiting a lower demand and subsequent lower price before buying it back and returning it to the original owner. Below are a few definitions of stock market terminology.

2.1 Efficient Market Hypothesis

The Efficient Market Hypothesis, also know as EMH, is an economic theory first introduced in 1965 by [36].It has been influential in both academic and commercial contexts since its introduction. It is a necessary consideration for any research attempting to model or trade financial markets, since in its strongest form the hypothesis concludes that prediction of stock prices is impossible. As we will discuss, the EMH rests on some assumptions that may not necessarily hold in reality, but the principle that the predictive value of information can be lost as markets become more efficient could be important to this research, which proposes a multi-predictor approach in anticipation of individual predictor's performance being diminished. Therefore, a comprehensive understanding of EMH is important in the context of the stock market. This section will provide a brief description of the EMH concept followed by a critical analysis of the theory.

2.1.1 Efficient Market Hypothesis

EMH states that stock prices in an efficient market are the reflection of all available information about a company and investors' expectations. This implies that it is impossible to profit from trading stocks [59] using information of any kind: private, public or price history information [81]. In 1965, Eugene Fama defined an efficient market as one which satisfies the condition of current information being available to a large number of rational market participants, who are competing to maximise profits through predicting the future market values of securities.

In 1970, Fama developed his efficient markets idea into the Efficient Market Hypothesis [37], of which there are three forms. Firstly, weak-form EMH, which is defined by current and future security prices already including the information contained in historical prices. Secondly, the semi-strong form is where current and future prices already reflect past prices and all available public information. Finally, strong-form, which asserts that securities' prices already include all relevant information, including inside information. More detail is given on these below. However, the important conclusion of all forms of efficient markets is that, since there are no predictable patterns, the prices must be random. This is why in the experiments that follow in the discussion section, evidence of prices following a random walk are sought to demonstrate market efficiency.

2.1.2 Weak Form Efficiency

Under the weak form of EMH, future prices cannot be predicted by historical price movement or volume since prices already account for this information. Therefore, the historical price data has no correlation with present prices [70] [40]. Moreover, the randomness of price movements makes it impossible to discover trends, so one cannot take advantage of such movements to make money in the market. Future earnings or growth does not rely on past prices either. This version of EMH implies that one cannot profit using technical or trend analysis. The reason for this is that if the past information carried reliable signals for predicting the future, everyone would have learned and exploited them. Note that an efficient market assumes the existence of a large number of rational competing agents. Therefore, these signals will ultimately lose their value since the expectation of price increase would be priced into the current price, leaving no room for profit.

In summary, as it is believed to be impossible to achieve a return greater than a weak-form efficient market with technical analysis [40] [55], investors who believe in weak-form efficiency can randomly select securities and should end up with similar returns.

Here is an example of weak-form efficiency [58]. An investor observes that the share price of Company B consistently rises on Monday and declines on Friday. Therefore, one Friday, she buys 100 shares of this stock at 10p per share in the expectation she can sell them for a higher value on Monday. However on Monday, the share price declines to 9p per share. This situation indicates that the market is a weak-form efficiency since the investor was unable to use information about past prices, in this case a historical pattern in the prices, to produce a profit.

2.1.3 Semi-strong Form Efficiency

Semi-strong form efficiency believes that historical, present and future information impacts the prices of stocks to varying degrees. It proposes that all public financial information is already included in security prices [113]. This implies that historical prices and other public information is immediately reflected in the price [40], again with the conclusion that investors are not able to predict future values - and thereby profit - using public information, which can include fundamental data such as the management quality, balance sheets, patents held or earnings forecasts of a company. This is in addition to the weak-form of Market Efficiency, so technical analysis also cannot yield a profit. Notably, semi-strong form efficiency excludes non-public information that could affect the price but is known only to company insiders.

An example of semi-strong form efficiency would be if the investor from the previous example reads a news report saying company B is about to release outstanding results due to the popularity of a new product. The price jumps on the news report as it reflects this expectation, meaning the investor is unable to benefit from this announcement.

Once they own the shares, they may profit only if the actual results posted by the company exceed the expectation reflected in the current price, or they may disappoint and cause the share price to decline, but in either case the public information, the news report, is inconsequential and cannot be used to predict the future value.

2.1.4 Strong-Form Efficiency

Strong-form efficiency proposes that all relevant information, including non-public information only available to insiders, is already reflected in stock prices [37]. In this context, insiders can be company employees, managers or board-members who have the privilege to see some exclusive information that is not available to the public. The strong-form EMH is an extreme situation in which all information available in the world is already incorporated into the price. Therefore, no one can take advantage of any information to profit from the market, even insiders who have exclusive information.

An example of strong-form EMH [54] could be the chief technology officer (CTO) of a public company taking a short position against his company based on non-public information he has seen which predicts a new product feature will cause a loss of customers and therefore revenues. He will profit if the share price of his company declines. However, upon release of the feature the share price doesn't decline as he expected, even though customers are dissatisfied with the new product. This situation is therefore considered an example of strong-form market efficiency because even the insider information was already priced in.

2.1.5 Discussion

EMH has been lengthily debated. What follows are some examples of research which have performed various experiments to test whether the market is efficient. Due to the volume of academic research into the various levels of EMH, we will only focus on the weak-form EMH in this section as this has the greatest implications for our research, which attempts to predict future values using only historical price data. After discussing academic research examples, we will also review examples of EMH as applied in practice in real-world financial markets.

Kendall [60], was a British statistician who investigated the characteristics of time-series data of stocks. The objective of this work was to identify whether correlations exist within and between financial series. Kendall reported little correlation within and between series, therefore concluding that stock markets were unpredictable without outside data, although importantly some scope was left for individual stocks being predictable assuming they do not behave the same as the average of a group of similar stocks.

This work ultimately led to the later random walk model of market prices. In 2003, research by Burton [71] accepted that there are anomalous behaviours in the market but proposed the market had become more efficient and less predictable, concluding that anomalies were not large enough to create trading opportunities. This means investors are still unable to achieve extraordinary returns over the long term.

In research from 2014, Fatih and Yasin [62] examined weak-form efficiency in the FTSE 100 index of shares from the UK market. The period of investigation ranged from 2001 to 2009. The main objective was to investigate whether the index follows a random walk model. Unit root tests were applied to test the stationarity of the time series. The results from the Phillips-Perron (PP test) and Dickey-Fuller test (ADF Test) show that the FTSE 100 was not stationary over the test period. Moreover, the movement of prices showed a pure random walk with no drifts or trends. Additionally, they applied the Generalised Auto-Regressive Conditional Hetroskedasticity, or GARCH(1,1), to investigate volatility in the time series. The results of this also supported market prices following a random walk model in all experiments (ADF, PP and GARCH). In summary, Fatih and Yasin's results support the weak-form hypothesis as applied to the FTSE 100 index.

On the other hand, there are many researchers who have produced results opposing the EMH. For example, in 1996, Sunil [88] argued that while there is evidence of market efficiency in developed markets, the situation is different in emerging markets, such as the Indian market from 1987 to 1994. Sunil performed several experiments testing whether the Indian market during that time followed the weak-form hypothesis and whether the day of the week demonstrated any predictability of stock price performance. The dataset in this research was the Bombay Stock Exchange National Index (BSENI) from 2^{nd} of January 1987 to 31^{st} October 1994. In order to make the results comparable with other works, the daily index values were converted to US Dollars. Since Sunil mainly focused on the weak-form hypothesis, the null hypothesis was that stock prices in the BSENI followed a random walk. In order to test this hypothesis, Sunil attempted to prove that the Indian index has no first-order correlation. In terms of the day-of-the-week effect, this research hypothesised that there would be no day-of-the-week effect up to five consecutive day lags. . Sunil performed several experiments and interpreted the results statistically. Firstly, the distribution of the dataset was evaluated. The result showed that the distribution was positively skewed with a kurtosis value of -0.530. These results proved that the distribution of BSENI was not normal. Therefore, it did not follow the random walk model. Secondly, Sunil ran more experiments to confirm the result. The non-parametric test called the Kolmogorov Smirnov Goodness of Fit test (KS) was used in this experiment. The result clearly showed that BSENI's distribution was neither normal or uniform with a Z value of 0.0000 at 95% level of confidence. Thirdly, the Wald-Wolfowitz Runs test was performed and also indicated similar results that the null hypothesis - the price is random - was rejected. Fourthly, the correlation of the prices was tested. The results showed that the values on the 1st, 4th, 10th, 14th and 15th lags are dependent. Therefore, the hypothesis that there is no serial correlation in the BSENI was rejected. Finally, Sunil found out that the returns on Fridays are significantly higher than other days for BSENI. Therefore, the results rejected the hypothesis that there is no difference between the returns obtained on different days of the week. Overall, Sunil concluded that BSENI does not follow the weak-form efficiency hypothesis.

Sachin and Kantesha [59] also performed multiple tests for the weak-form hypothesis. Twenty-three stocks from various sectors of the Indian stock market were tested to investigate whether the market is efficient. They experimented over a 10-year period starting from the 1st of April 2004 to the 31st of March 2014. To investigate the weak-form EMH, the random walk hypothesis and auto-correlation were used. The objective of this study was to identify whether the Indian stock market follows a random walk model. To produce the results, both non-parametric and parametric statistics were included, for example Dickey Fuller and Auto-correlation coefficient tests. Their study found that the Indian stock market doesn't follow a random walk model, therefore the market is not efficient.

Sachin and Kantesha used a series of experiments to come this conclusion. Here we present a few. Firstly, the experiment to test the efficiency of the market using a non-parametric model on six different stocks from the banking sector showed that the Z values are negative, while the critical value of Z for the level of confidence at 95% is 1.96. Therefore, the null hypothesis that the stocks follow a random walk was rejected. Similar results were also found in all of the sectors in their experiments. Secondly, the experiments on auto-correlation found auto-correlation values fell between 0.941 and 0.999. When the lag time was increased, the impact of historical prices decreased. If the market is weak-form efficient, the correlation should be near zero. Therefore, for the correlation values Sachin and Kantesha came up with, it can be concluded that the prices do not follow the weak-form hypothesis. In conclusion, the Indian market in that period of testing was not efficient in its weak form.

On the basis of their experiments, Sachin and Kantesha summarised that share prices in the Indian stock market are not a reflection of the fair value of the stock, therefore investors could be able to profit in excess of the market return using historical information.

Saqib and Muhammad [81] presented empirical evidence from the major stock markets in South Asia supporting that prices do not follow a random walk model. They summarised that these markets are not efficient. The four stock markets their research focused on are the Karachi Stock Exchange (KSE-100), Bombay Stock Exchange (BES-SENSEX), Dhaka Stock Exchange (DSE-GEN) and Colombo Stock Exchange (CSE-MPI). This research used daily price information, as the majority of research has, but also considered bigger timeframes, such as monthly and weekly information. The test period spanned 14 years from July 1997 to June 2011. The statistical methods they applied are similar to the other research presented in this chapter: serial correlation and non-parametric runs test. However, they also included new tests which have not been used much in the past, such as variance ratio and unit root tests.

Saqib and Muhammad performed four different experiments based on four statistical methods: serial correlation, runs test, unit root tests and variance ratio. For each testing method, the daily, weekly and monthly data were tested separately. Firstly, for the runs test, most of the results were in contradiction to the random walk hypothesis, but the weekly and monthly data of BSE and DSE, as well as the monthly prices of KSE, turned out to follow a random walk according to the higher Z values. Therefore, out of a total 12 experiments, 5 results supported the random walk and 7 results rejected it. However, it can be concluded from these experiments that not all daily prices follow the random walk hypothesis, and daily information is therefore not believed to be random. The second experiment used the Durbin-Waston correlation test. All results consistently show that the prices from all timeframes and all datasets do not follow a random walk model. Results from the rest of experiments, unit root tests and variance ratio, have also turned out to be consistent. All results support the rejection of weak-form EMH. Therefore, Saqib and Muhammad concluded that their findings illustrate that none of the four South Asian markets follow the random walk hypothesis. Hence, they are not the weak-form efficient markets.

We have mentioned multiple research examples testing the efficient market hypothesis in developing markets and have found much evidence contradicting EMH. However, as Sunil [88] hypothesised, these may be different to developed markets, implying results from these markets cannot be generalised to disprove EMH across all markets. In 2010, Chien et al [65] tested EMH with wider datasets. This research used 32 developed and 26 developing countries. The developed countries included Canada Australia, the United States and the United Kingdom. For the developing countries they selected, for example, Brazil, Colombia, India and China. All datasets were tested over the period from 1999 to 2007. The main objective of the research was to investigate whether there were failures of previous EMH studies. The researchers designed experiments to decide whether the stocks of their chosen markets are stationary or do not follow a random walk model.

The methodology to perform the stationary test in this work was proposed earlier in 2005 by Carrion [52]. Carrion proposed a stationary test capable of testing the null hypothesis while allowing multiple structural breaks. This model was tested with monthly stock prices for the designated countries as mentioned above, which were not limited to one continent. The results from developing and developed countries are similar in that they did not reject the stationary hypothesis. Moreover, the results show that in developing markets, stock prices do not instantaneously reflect relevant information. Finally, Chien et al summarised that EMH does not hold for developing or developed markets.

Another research which tested the EMH on both developing and developed markets was published in 2011 by Pabitra [76]. This research examined the efficiency of markets around the world, including emerging markets (such as India and China) and developed markets (such as the UK and USA). The period of testing ranged from January 2007 to December 2010. The GARCH (1,1) and ADF tests were used on the logarithm returns of the daily index from each country in order to investigate the volatility and the random walk model. Firstly, the reported results from the ADF unit root test for all selected markets show that index prices do not follow a random walk, hence are not efficient. Secondly, in order to confirm the results of the ADF test, Pabitra estimated the GARCH (1,1) model for each selected market individually and found evidence of high persistence of volatility during the period of testing for both emerging and developed markets. Such high persistence of volatility indicates inefficiency of the markets.

Finally, Pabitra concluded that markets are moving toward efficiency in the long term, citing how inefficiencies provide incentives and opportunities for novel financial products to emerge. Therefore, there is a possibility in the meantime for investors to outperform the markets; not only the big corporations with inside information, but individual traders with high data-analysis skills can also do consistently well.

EMH has become a long-standing and well-known theory in finance and despite ample literature both in support of and attempting to disprove the hypothesis in its various forms, there is yet to be a definitive answer as to its truth. Any empirical evidence is limited by the period the data is taken from, whereas financial markets are continuous and changing. Evidence that markets follow a random walk over any one period of any length does not mean it always has or always will be random. For example, [62] and [76] tested the efficiency of the same market, the FTSE 100 index, over slightly different time periods. The former selected the period between 2001 to 2009 but the latter worked on the period between 2007 and 2010. Even with the intersection of timeframes, the results turned out to be totally different, with one supporting and the other rejecting the EMH. In this context, it is worth considering evidence from market participants outside of academia, since there are examples even in developed markets of inefficiencies being consistently exploited to profit over the market return by the likes of hedge funds such as Renaissance Technology's Medallion and others as shown in table 2.1. These are well-known funds which have been profitable in the markets for a long time and reflect a mixture of strategies, including statistical arbitrage - looking for statistically significant inefficiencies in price behaviour - and the use of company and socio-economic data. The performance of these funds makes a good case for rejecting EMH.

The problem these present for research is that while their track record contradicts the EMH, great care is taken to prevent the inefficiencies and strategies used becoming widely-available, lest they become over-exploited and lose their predictive value, so they are not readily available for study. However, this seems like an implicit recognition from market participants that markets tend toward efficiency, but not necessarily that they are efficient, otherwise they arguably would not be in business at all. This tendency for markets to render some patterns and indicators obsolete shows the importance in the proposed system of being able to incorporate many predictors, reduce the influence of weaker ones, exploit the strong and add new ones into the system.

There is an additional problem when considering the performance of real world funds, and that is survivorship bias. This is using only a sample of survivors or "winners" - in this case funds with exceptional long-term returns - to support an argument without considering the size of the original sample including the subsequent "losers". The implication in this context would be that while the aforementioned returns of individual funds support the possibility of exploitable inefficiencies existing in developed financial markets, it is possible that the accumulated losses of unsuccessful funds render the collective expected return of market participants zero or negative, which would support the argument that developed markets are mostly efficient.

Investor	Key Fund	Period	Annualised Returns
Jim Simons	Medallion Fund	1988-2018	39.1%
George Soros	Quantum Fund	1968-2000	32%
Steven Cohen	SAC	1992-2003	30%
Perter Lynch	Magellan Fund	1977-1990	29%
Warren Buffett	Berkshire Hathaway	1965-2018	20.5%
Ray Dalio	Pure Alpha	1991-2018	12%

Table 2.1: Returns on leading funds [129]

2.2 Common Trading System

In an ordinary stock trading system, the main objective is to create a profitable portfolio (a collection of stocks). This section explains the main three aspects in finance that could be referred to in this research.

1. **Ordinary Shares** Shares, or common stocks, are the most abundant assets in the market and allow buyers to claim an ownership stake in a company and the right to vote in the corporation's annual shareholder meeting.

2. **Portfolio** A portfolio comprises shares that an investor currently holds. It will change over time after the investor has successfully completed buy and sell orders.

3. **The CAPM Model** *CAPM* or *Capital Asset Pricing Model* is an important mathematical model of portfolio management. It was originally proposed by William T. Sharpe [97]. CAPM describes the relationship between the expected return of assets and systematic risk [114]. The basic formula of CAPM is presented in Equation 2.1 [87].

$$E(r_i) = r_{rf} + \beta_i[E(r_{mkt}) - r_{rf}] \tag{2.1}$$

where

$E(r_i)$ = expected return of stock i.
r_{rf} = return of risk free assets which are guaranteed zero risk in theory
β_i = beta of stock i which compare the volatility of this stock and the market.
$E(r_{mkt})$ = expected return of the market.

From Equation 2.1, the expected return of a specific stock $E(r_i)$ is the difference between the expected return of the market and the return of the risk free rate. An example of a so-called risk-free asset are U.S Treasury securities - debt instrument of the U.S. government. Along with the expected return of the stock, another important aspect in the CAPM formula is β. β is basically the measurement of the stock's sensitivity to market changes and can be calculated as shown in Equation 2.2 [87].

$$\beta_i = \frac{cov(r_{mkt}, r_i)}{var(r_{mkt})} \tag{2.2}$$

From Equation 2.2, the result can be interpreted as high β, high sensitivity and low β, less sensitivity. High sensitivity means that the movement of the stock is more likely to change according to how the market changes. On the other hand, low sensitivity means that the stock is less likely to respond to market movements. If a stock moves in the opposite direction from the market, beta is negative [26]. This means that when the market goes up, a negative-beta investment decreases.

Another important parameter in portfolio management is α, which is the indicator for measuring the specific asset's performance. It indicates the asset's relative under or over-performance when compared to the market. α can be computed as shown in Equation 2.3 [87].

$$\alpha_i = r_i - [r_{rf} + \beta_i(r_{mkt} - r_r f)] \tag{2.3}$$

As can be seen from the Equation 2.3, α is the difference between the actual return of the stock and the expected return obtained from the benchmark or CAPM calculation. If α is positive, it means that the stock has outperformed the market. On the other hand, if α is negative it indicates that the stock underperformed the market. Zero α means the stock return is exactly the same as the result obtained from the CAPM prediction.

CAPM is used in financial research to setup portfolios in order to maintain a balance between risky and riskless assets. It can be assumed that the risk free asset - 3-month

US Treasury Bills are a common example - will not be correlated with other assets [23]. The initial portfolio is built by following the CAPM model. The weight rebalancing process is performed following every change. After rebalancing, the accumulated weights of stocks in the portfolio will be equal to one. This way, the performance of a portfolio can be controlled, ensuring that the portfolio is not overexposed to any kind of asset. In this research we adapt some ideas from CAPM, most notably the ideas of risk-adjusted return and comparison of returns against a benchmark and risk-free asset.

CAPM allows for the comparison of investment alternatives which are to be held simultaneously (a portfolio). In practice, investors use a range of approaches to portfolio construction, including so-called naive approaches such as 1/N: investing equal amounts into each stock with no effort to measure the risk based on past data. Such simple approaches have been shown to be as effective as more complex portfolio management models (OPTIMAL VERSUS NAIVE). This idea of simple approaches matching more complex approaches used in finance is interesting with regards to the research presented in thesis, since we will test whether a combination of simple existing machine learning approaches can produce good results as opposed to developing a sophisticated new machine learning approach.

We include this brief discussion of CAPM and portfolio construction as potential users of our final system will have to decide how our system's recommendations will fit into their portfolios. Note that portfolio here refers to a collection of investments held simultaneously. Investors create a portfolio so as to diversify, that is, to split their capital across investments so as to be exposed to lots of opportunities for profit and also so that smaller amounts are lost in the case of a bad investment (only the portion allocated to the bad investment will be lost). This research focuses on a different means of diversification by having a range of investment strategies - as opposed to assets - that change over time instead of being traded simultaneously. There is no reason, however, that investors using our system cannot combine our proposed approach with portfolio management methods described above.

2.3 Algorithmic Trading System

Algorithmic Trading or *Systematic Trading* refers specifically to the use of mathematical and statistical methods to analyse the historical data of stocks or other financial instruments that investors are interested in. Results from analyses are then used to create mathematical models that provide rational and unbiased data to support investment decisions. People who work in this area are normally called quantitative analysts (or quants). Algorithmic trading ultimately aims to predict stock price performance and automatically execute profitable trades for investors while maintaining a diversified portfolio.

Below are three important aspects that must be understood in order to gain a better understanding of existing algorithmic trading models.

2.3.1 Order and Exchange

There are two well-known types of orders that an algorithmic trading system can produce. The first type is *Market Order or MOs*, and another type is *Limit Order or LOs*. MOs can be considered an aggressive order which means that when the order has been sent, it is expected to be executed immediately. By sending MOs, an investor expects to buy or sell their shares at the best available price in the market at that time. On the other hand, LOs are considered to be passive orders and are not expected to be executed immediately. In this type of order, the investor has to specify their desired buy or sell price, and the order will be executed only if that price is matched in the market. Normally, the limit order price is worse than those in the market, for example it might be higher than the best buy price for sell LOs or lower than the best sell price for the buy LOs, so it will not be immediately executed as with MOs. Examples of MOs and LOs can be seen below:

GOOG, buy, 100, Market

The example above shows a market order saying the investor wants to buy 100 shares of Google stock at market price. So, if there is someone selling this asset, this order will be executed no matter what the price. For a limit order, the order works similarly as shown below:

GOOG, buy, 100, Limit, 200.15

For a limit order, the intended price has to be specified. As in the previous example, the investor wants to buy 100 shares of Google stock at the price at $200.15 or lower only.

Naturally, different order types have their pros and cons. For example, while a limit order can guarantee a share get bought at a price specified by the investor or better, that price might never be reached, resulting in missed opportunities. In addition to the opportunity cost of missed profit, traditionally this would have meant time analysing the share may have been wasted, although a benefit of algorithmic trading is that the investor does not have to spend much time or their own effort analysing securities. Market orders suffer from two main problems: the bid-ask spread and slippage. Market orders can be filled near-instantly in highly liquid shares and are the quickest and easiest orders to make. However the shares are bought at a slightly higher price than that which they can immediately be sold at, effectively meaning the investor makes a small loss as soon as they buy the share. This is the spread, and the share price has to increase to cover this. In highly traded shares, the spread tends to be very small. Slippage is when the market order is filled at a different price what was offered when the order was submitted. It can be better or worse. This is likely to happen in fast-moving and volatile markets. For our proposed system, we will use market orders. Our datasets represent highly liquid shares of large companies, which tend to have small spreads and for which slippage is less of a problem since there are more buyers and sellers in the market and therefore a greater chance of getting the quoted price. Additionally, we see limit orders as adding unnecessary complexity to our system. For example, since limit orders usually attempt to buy at a cheaper price than currently, our system would have to predict how far down the price will go before it goes up again. Should it not go down that far, good opportunities may be missed. Moreover, the fact it has declined some amount and that we have had to wait may change the prediction, so we have selected market orders for this experiment.

2.3.2 Order Book

A stock exchange will publish information about orders - including the parties to the transaction, the number of shares and the price being paid - in what is called an order book. Some traders use information published in the order book to inform their trading decisions, for example if a respected institution or investor recently purchased a large amount of a particular share, a trader may wish to copy them. They might also be able to determine whether trading of a particular share is being driven by retail traders (individuals at home) or institutions. This may be important because retail investors generally represent only a small amount of agents in the market, and they transact in smaller sizes, which means shares with prices driven by retail traders are less liquid - it may be harder to buy or sell the shares at desirable prices. Institutions can make large and frequent transactions and represent a larger portion of market players, so an order book showing a lot of active institutional investors suggests a stock will be highly liquid and better prices can be sought.

Liquidity can be important to consider due to the supply and demand mechanism in the market [77]. If one holds a share whose price is driven by retail traders, and some bad news is announced about the company, demand can dry up very quickly as there are fewer retail traders, and those who want to buy can only purchase small amounts. Therefore, those holding the shares who want to sell them end up stuck with them and have to offer much lower prices - probably incurring a loss - in order to sell them. This is a case where supply exceeds demand and is an example of why stock prices decline [24]. Alternatively, the holder of a stock for which there is little supply (few shares have been issued by the company, or those holding the shares want to keep them as they expect the company to grow) will see the value of their shares increase if large buyers such as institutions become interested in buying them, for example if the company generated a strong profit for the year, causing demand to outstrip supply of the stock.

It should be noted that the order book is only available to traders for a fee, it is not freely published. Perhaps for this reason, it is more widely used by institutions than retail traders, and is generally used by those who trade frequently such as day traders. This makes sense since their shorter time horizon means they trade more frequently to make lots of small profits, meaning buying or selling at bad prices has a larger impact on their results. So calculating their chances of getting a good price is more important.

2.3.3 Technical Indicator

In technical analysis, there are many metrics calculated from the price history of stocks. These metrics are called *indicators* or *technical indicators* and attempt to recognise when a predictive pattern occurs in order to long (buy) or short (sell) a stock. In the following section, popular indicators will be described:

1. Moving Average (MA):
   A widely-used technical indicator is the *Moving Average*. MA smooths price action by removing the noise from fluctuations in the price. MA can be used as a trend following or lagging indicator, since it is calculated from the historical price. There are many different ways to calculate the Moving Average depending on how the investor expects this indicator to react to price changes.

   (a) Simple Moving Average (SMA): SMA is considered the simplest and oldest way to calculate the moving average from a stock's prices [29]. SMA can be easily calculated from recent prices. SMA is suitable for normally distributed data and will react slowly to price changes compared to other moving average approaches. The formula for SMA is represented as follows:

$$M_k = \frac{1}{n} \sum_{i=k-n+1}^{k} C_i \tag{2.4}$$

$$C_i = \frac{(C_{k-n+1} + C_{k-n+2} + C_{k-n+3} + ... + C_k)}{n} \tag{2.5}$$

where:

M_k = simple moving average at period k.
C_i = closing price of period i.
n = total number of periods to be used in the calculation.
k = number of positions of the period being studied within the periods.

(b) Weighted Moving Average (WMA): In WMA, data is observed in proportion to its position in time. The most recent data will be given the highest weight and the oldest data the lowest weight. Then the sum of the daily values multiplied by their weights is divided by the sum of all weights [29].

Example
Assume that we want to calculate a 6-day WMA. The calculation processes can be seen as Figure 2-1

6-day weighted moving average for a hypothetical stock

closing price from	*daily closing weights*	*multiply*	*price*	*equals*	*products*	*sum of products/ sum of weights*
5 days ago	1	×	50	=	50	
4 days ago	2	×	51	=	102	
3 days ago	3	×	53	=	159	
2 days ago	4	×	56	=	224	
1 day ago	5	×	60	=	300	
today	6	×	50	=	300	
sums	21				1135	54

Figure 2-1: Weight Moving Average (WMA)
[29]

From the Figure 2-1, the five steps to calculate WMA are as follows:

Step 1: Assigning the weights 1, 2, 3, 4, 5, 6 to the 6 most recent daily data positions, where 1 is assigned to the oldest day and 6 means the present.

Step 2: Multiplying the closing price of each day with its weight from step 1

Step 3: Calculating the summation of all values in step 2

Step 4: Adding up sum of all weights from step 1

Step 5: Dividing the summation from step 3 by the summation of the weights from step 4

To complete the WMA calculation, these 5 steps will be calculated repeatedly during every period of calculation, as shown in Figure 2-2;

(c) Exponential Moving Average (EMA): EMA is also called *Exponential Smoothing.* EMA is considered to be the best moving average [29] and is increasingly preferred by technical analysts over other moving average approaches. EMA responds quickly to new data in the market. It is considered a good compromise between the insensitive SMA and over-sensitive WMA. Compared to other moving average techniques, EMA follows trends of the data smoothly and minimises jumps very well.

EMA is calculated by using just two values, the data for the current period and the EMA for the previous period. Hence EMA does not need to preserve and handle many historical prices. A key benefit of EMA is that it is not distorted by old data suddenly dropping out of the calculation. Old data is never suddenly dropped because it is not explicitly included in the calculation. The effect of past data gradually fades away due to the ever decreasing weight of yesterday's EMA. Also, the EMA method of calculation avoids erratic current movement due to obsolete and irrelevant data dropping out of the calculation process [29]. EMA is calculated as follows:

$$\mathrm{EMA} = (C - \mathrm{E_p})K + E_p \tag{2.6}$$

where:

row	month-end date	closing price	multiply	weights	equals	products	moving sum of products	sum of products/ sum of weights
6	2/28/74	51.56	×	1	=	51.56		
7	3/29/74	50.21	×	2	=	100.42		
8	4/30/74	47.93	×	3	=	143.79		
9	5/31/74	45.92	×	4	=	183.68		
10	6/28/74	44.90	×	5	=	224.50		
11	7/31/74	41.55	×	6	=	249.30	953.25	45.39
12	8/30/74	37.70	×	1	=	37.70	939.39	42.73
13	9/30/74	33.45	×	2	=	66.90	905.87	39.52
14	10/31/74	38.97	×	3	=	116.91	878.99	38.68
15	11/29/74	37.13	×	4	=	148.52	843.83	37.74
16	12/31/74	36.13	×	5	=	180.65	799.98	36.93
17	1/31/75	40.91	×	6	=	245.46	796.14	37.91
18	2/28/75	43.07	×	1	=	43.07	801.51	39.54
19	3/31/75	44.21	×	2	=	88.42	823.03	41.23
20	4/30/75	46.19	×	3	=	138.57	844.69	42.98
21	5/30/75	48.46	×	4	=	193.84	890.01	45.03
22	6/30/75	50.85	×	5	=	254.25	963.61	47.23
23	7/31/75	47.52	×	6	=	285.12	1003.27	47.77
24	8/29/75	46.29	×	1	=	46.29	1006.49	47.65
25	9/30/75	44.49	×	2	=	88.98	1007.05	46.86
26	10/31/75	47.05	×	3	=	141.15	1009.63	46.79
27	11/28/75	48.24	×	4	=	192.96	1008.75	47.02
28	12/31/75	47.64	×	5	=	238.20	992.70	47.09
29	1/30/76	53.55	×	6	=	321.30	1028.88	48.99
30	2/27/76	53.35	×	1	=	53.35	1035.94	50.56
31	3/31/76	54.80	×	2	=	109.60	1056.56	52.20
32	4/30/76	54.11	×	3	=	162.33	1077.74	53.15
33	5/31/76	53.31	×	4	=	213.24	1098.02	53.54
34	6/30/76	55.71	×	5	=	278.55	1138.37	54.38
35	7/30/76	55.26	×	6	=	331.56	1148.63	54.70

Figure 2-2: Weight Moving Average (WMA)
[29]

EMA = the Exponential Moving Average for the current period.
C = the closing price of the current period.
$\mathrm{E_p}$ = the Exponential Moving Average for the previous period.
K = the Exponential smoothing constant, which equal to $\frac{2}{n+1}$.
n = the total number of periods in SMA to be approximated by EMA.

2. Relative Strength Index (RSI)
   RSI is an indicator that acts as a price-momentum indicator [29]. It is used to identify the general trend of stock prices and evaluate the speed of price movement. RSI normally ranges from 0 to 100. One way to use RSI as a decision-making indicator is monitoring when the RSI value goes over the upper bound threshold and selling that stock or buying when RSI is beneath the lower boundary. RSI can be calculated as Equations 2.7 - 2.8.

$$\mathrm{RSI} = 100 - \frac{100}{1 + \mathrm{RS}} \tag{2.7}$$

$$\mathrm{RS} = \frac{\mathrm{AvgGain}}{\mathrm{AvgLoss}} \tag{2.8}$$

   Where RS represents Relative Strength. There are two steps to calculate average gain and average loss;

   First calculation

$$\mathrm{AvgGain} = \frac{\sum_{i=1}^{n} \mathrm{Gain_i}}{n} \tag{2.9}$$

$$\mathrm{AvgLoss} = \frac{\sum_{i=1}^{n} \mathrm{Loss_i}}{n} \tag{2.10}$$

   Where $Gain_i$ and $Loss_i$ are gain and lose at the day i and n is the period of the RSI calculation. The default number of n is 14 days.

   Second and subsequent calculations

$$\mathrm{AvgGain_t} = \frac{(\mathrm{AvgGain_{t-1}} * (n-1)) + \mathrm{Gain_t}}{n} \tag{2.11}$$

$$\mathrm{AvgGain_t} = \frac{(\mathrm{AvgLoss_{t-1}} * (n-1)) + \mathrm{Loss_t}}{n} \quad (2.12)$$

Where t represent the present day

When RSI goes over the upper bound, it can be interpreted as meaning the price at that time is *overbought* or *overpriced*. It can be assumed that the price will fall in the near future. On the other hand, if RSI drops under the lower bound, it can be considered *oversold* or *underpriced* and the price of that stock is expected to rise [29]. The bounds are normally determined by the standard deviation of the price's moving average.

3. Bollinger Band or *BB* is an indicator based on Standard Deviation of prices changes [128] which can be used to compare relative prices and volatility over a specific period of analysis. BB are composed of 3 main values or three bands, called Upper Band, Middle Band and Lower Band, which are calculated as follows [19]:

$$Upper\_band = \bar{X} + 2\sigma \quad (2.13)$$

$$Middle\_band = \bar{X} \quad (2.14)$$

$$Lower\_band = \bar{X} - 2\sigma \quad (2.15)$$

$$\sigma = \sqrt{\frac{\sum_{j=1}^{N}(X_j - \bar{X})^2}{N}} \quad (2.16)$$

$$\bar{X} = \frac{\sum_{j=1}^{N}(X_j)}{N} \quad (2.17)$$

where

X = stock price
N = period of time (days).
σ = standard deviation.

Bollinger bands are plotted above and below the simple moving average of the stock prices. The standard distance is two standard deviation away which can be seen in figure 2-3.

Figure 2-3: Higher and lower bollinger bands

Figure 2-3 shows the Bollinger bands composed of the rolling mean, upper band and lower band. The rolling mean is the SMA calculated from the stock prices (SPY). The lower and upper bands are computed by expanding the rolling mean by adding and subtracting its product with the standard deviation. An example of applying these bands to the stock signal generation process is if the price increases until exceed the upper band, a sell signal is generated. On the other hand, if the price drops below the lower band, the buy signal is sent.

4. Average True Range (ATR) *Average True Range* or *ATR* is a financial indicator for measuring the historic volatility of a market [119]. ATR is typically calculated using a 14-day moving average. Although developed for use in commodity markets, ATR is now widely used across many different types of investments including indices and stocks. ATR can be interpreted easily as a high ATR indicates high volatility and a low ATR indicates low volatility. As such, this indicator does not predict the price's direction. It is commonly used as an exit signal together with other indicators.

   ATR is a subjective measurement. A single ATR value cannot inform of any particular trend. To use ATR, traders must compare ATR with an earlier value or use it together with other indicators, e.g. ADX, which is explained in the following section.

ATR can be computed as follows:

$$TR = max[(high - low), abs(high - close_{previous}), abs(low - close_{previous})] \quad (2.18)$$

$$ATR = \frac{1}{n}\sum_{i=1}^{n} TR_i \quad (2.19)$$

where:

TR_i = a particular true range
n = time period employed (default 14 days)

5. Average Directional Index (ADX) *Average Directional Index* or *ADX* is a financial indicator used to evaluate the strength of a trend. Trends in stock prices can be either up or down, and this can be shown by two additional indicators, the Positive Directional Indicator, DI^+ and the Negative Directional Indicator (-DI). The three indicators are generally used together to generate a signal.

   The direction of a stock price trend can be interpreted as follows: an uptrend is expected when +DI is more than -DI. On the other hand, a downtrend is expected when -DI is greater than +DI. To make a decision on going short or long, traders wait for crossover moments when one of the indicators becomes greater than the other, which they will take as a signal. When a crossover or trend happens, traders will use ADX to evaluate its strength. If ADX is below 20, the trend can be classified as *Weak*. Meanwhile, any value above 25 can be considered a *strong* signal. [119]. A weak or absent trend does not necessarily mean the price is stable, but may instead mean that the price is too volatile to identify a clear direction. ADX, +DI and -DI can be calculated as follows:

$$DI^+ = (\frac{SmoothedDM^+}{ATR}) * 100 \quad (2.20)$$

$$DI^- = (\frac{SmoothedDM^-}{ATR}) * 100 \quad (2.21)$$

$$DX = (\frac{|DI^{+} - DI^{-}|}{|DI^{+} + DI^{-}|}) * 100 \tag{2.22}$$

$$ADX = (\frac{(PriorADX * 13) + CurrentADX}{14}) * 100 \tag{2.23}$$

where:

$+DM$	= Current High - Previous High
$-DM$	= Previous Low - Current Low.
$SmoothDM^{(+)/DM^{(-)}}$	$= \sum_{t=1}^{14} DM - \frac{\sum_{t=1}^{14} DM}{14} + CurrentDM$
ATR	= Average True Range

The above are some traditional technical analysis methods used by traders in financial markets. As can be seen, they suffer to a degree from elements that are arbitrary, such as how many days over which to calculate the moving average or the upper and lower bounds of RSI. Each technical analyst may use different figures. In the past, it may have been difficult to figure out the best values for each stock to which an indicator was applied, and traders might select one set of values to apply to all stocks. Nowadays, it might be possible to figure out the best values for each stock based on its past, and this research builds on this idea that each stock's historical price data has it's own unique characteristics. However, these may be more subtle and complex than can be identified by the models above. This may because as such indicators have become widely known the market prices in their expectations, and as predicted by the EMH the predictive value of the indicators has been diminished. As will be seen, this research borrows one traditional technical analysis technique, EMA, but applies this to monitor the scores of different predictors rather than predict future stock prices.

2.3.4 Backtesting and System Evaluation

Backtesting is the process of evaluating a trading system by testing it with an unseen dataset. The purpose of this is to see whether a trading strategy can work out of sample, on data other than that which it was trained. The backtesting dataset covers a certain period from the available past price data of a stock, but the algorithm is not allowed to observe this data and use it to train. Backtesting can be performed by running the system on various sets of historical data and measuring the performance.

1. Annual Return: The *annual return* or *yearly return* is a basic metric for measuirng trading systems. A common method to calculate the return is the percentage increase or decrease in capital, also known as the “Simple Return” or “Arithmetic Return”. While the simple return provides an intuitive and meaningful evaluation of an investment, there are some quirks that make it difficult to use. For example, the continuously compounded arithmetic return is not symmetric. A loss of 50% requires a subsequent 100% to return to break-even. Such asymmetries are not always obvious when simple return values are presented. To avoid this peculiarity, the logarithmic return or continuously compounded return is often used in academic and valuation settings. This presents the above returns, -50% and 100% as -69.31% and 69.31% respectively. In this section, we will include both the simple return and logarithmic return. However, in our experiment, the logarithmic return is favoured as we take into account the compounding value.

   - Simple Return

   $$\begin{aligned} simR_t &= \frac{P_t - P_{t-1}}{P_{t-1}} \\ &= \frac{P_t}{P_{t-1}} - 1 \end{aligned} \tag{2.24}$$

     Where

     $simR_t$ = Simple return at time t
     P_t = Price at time t
     P_{t-1} = Price at time t-1

   - Logarithm Return or continuously compounded Return

   $$\begin{aligned} logR_t &= \ln\left(\frac{P_t}{P_{t-1}}\right) \\ &= \ln(P_t) - \ln(P_{t-1}) \end{aligned} \tag{2.25}$$

     Where

     $logR_t$ = logarithm return at time t
     P_t = Price at time t
     P_{t-1} = Price at time t-1

2. Sharpe Ratio (SR): The *Sharpe Ratio* is also a common measure used to evaluate investments that includes the mean and variance of returns. It was introduced by Sharpe [98] in order to measure the performance of mutual funds. This measurement is also known as the *reward-to-variability*, Sharpe Index [90] and Sharpe Measure [1].

   Beyond evaluation, many common financial measurements are also calculated from historical data to make decisions about future investments, under the assumption that historic results have some predictive value [99]. For example, if the historical performance of security X is higher than security Y, it is assumed that X will have higher performance than Y in the future too. More specifically, it may be higher or lower by some multiple of the historic measure. In acknowledgement of these two uses, Sharpe [99] defined two versions of his ratio, ex ante for theoretical discussion, and ex post after an investment has been made. Both are defined below.

   - Ex Ante Sharpe Ratio: the differential return can be computed as equation 2.26. In this equation, the tildes over variables mean that the exact values may not be available in advance.

$$\tilde{d} \equiv \tilde{R}_F - \tilde{R}_B \tag{2.26}$$

     Where

     $\tilde{d}$ = differential return
     R_F = Return of fund F
     R_B = Return on a benchmark portfolio (or a security)

     In this case, the Sharpe Ratio which indicates the expected differential return per unit of risk can be calculated as equation 2.27

$$S \equiv \frac{\bar{d}}{\sigma_d} \tag{2.27}$$

     Where

     S = Sharpe Ratio
     $\bar{d}$ = the expected values of d
     σ_d = the predicted standard deviation of d

- Ex Post Sharpe Ratio: the differential return at time period of t can be computed as equation 2.28.

$$D_t \equiv R_{Ft} - R_{Bt} \tag{2.28}$$

Where

D_t = differential return in time period t
R_{Ft} = the return on fund F in time period t
R_{Bt} = the return on a benchmark portfolio (or a security) in period t

The average of D_t between time period $t = 1$ and $t = T$ is calculated as equation 2.29.

$$\bar{D} \equiv \frac{1}{T}\sum_{t=1}^{T} D_t \tag{2.29}$$

Where

$\bar{D}$ = the average value of D_t over time period $t = 1$ to T

The equation to calculate standard deviation of D is shown in equation 2.30.

$$\sigma_D \equiv \sqrt{\frac{\sum_{t=1}^{T}(D_t - \bar{D})^2}{T-1}} \tag{2.30}$$

Where

σ_D = standard deviation over over time period $t = 1$ to T
D_t = differential return in time period t
$\bar{D}$ = the average value of D_t over time period $t = 1$ to T

Finally, the Sharpe Ratio or historic Sharpe Ratio, which indicates the historical average of the differential return per unit of the historic variability, is shown in equation 2.31.

$$S_h \equiv \frac{\bar{D}}{\sigma_D} \tag{2.31}$$

Where

S_h = historical Sharpe Ratio
$\bar{D}$ = the average value of D_t over time period $t = 1$ to T
σ_D = standard deviation over over time period $t = 1$ to T

Both the ex ante and ex post Sharpe ratios are time dependent and relate to the period over which they are measured. For example, the Sharpe ratio when the return is calculated between adjacent days will be different from when it is calculated between adjacent weeks. $\bar{d_1}$ and σ_{d_1} are the one-period mean and standard deviation of the differential return. Assuming that the differential return over time period T is calculated by accumulating the one-period differential returns, and the returns have zero serial correlation, the mean and standard deviation of the T-period returns can be calculated as equations 2.32 - 2.34.

$$\bar{d_T} = Td_1 \tag{2.32}$$

$$\sigma^2_{d_T} = T\sigma^2_{d_1} \tag{2.33}$$

$\therefore$

$$\sigma_{d_T} = \sqrt{T}\sigma_{d_1} \tag{2.34}$$

Where

$\bar{d_1}$ = one-period mean of the differential return
σ_{d_1} = one-period standard deviation of the differential return
$\bar{d_T}$ = mean of the T period return
σ_{d_T} = standard deviation of the T period return

Under the assumed conditions above, the Sharpe Ratio for the one-period and T-period can be computed as equation 2.35.

$$S_T = \sqrt{T}S_1 \tag{2.35}$$

Where

S_1 = Sharpe Ratio for one-period
S_T = Sharpe Ratio for T-period

Practically, the Sharpe Ratio will be shown as an annualised value in order to provide reasonable meaningful comparisons to other strategies. In this research, we use the ex-post Sharpe Ratio because this value takes into account both profit and risk, which are considered to be important in our system. However, we will also provide other measurements to give more details of the system's performance.

Clearly, the Sharpe Ratio for a trading strategy is not stable and changes over the period tested, as the volatility and achievable profit will be different. In practice this opens the measure up to being abused by backtesting over the period that gives the best Sharpe Ratio. We have avoided this in our research by selecting a neutral test period and using the same period for all stocks tested. However, this means that when comparing strategies based on Sharpe Ratios, the method used to select the sample is an important consideration also.

In order to make the calculations in this research closer to what is practised in the real world and comparable to other research, a risk-free return rate is taken into consideration. While the Sharpe ratio can indicate whether the return achieved is good for a given level of risk, an investment may still not be a good idea if a similar return could be achieved for theoretically no risk. The risk-free rate of return is included by subtracting it from the total return of a strategy: the return on a strategy is the amount that outperforms the risk-free rate [68] [89] [17].

It is common to use the yield of short-dated domestic government bonds as a proxy for the theoretical risk-free rate as a government is unlikely to default. Given the USA's dominance in finance, the yield on US Treasury bills or bonds are often used as a risk-free rate in investment, or a historical average of them in order to smooth out periods of exceptionally high or low interest rates. This is because they are viewed not only as the safest investment but also used as a proxy for many important financial matters, for example mortgage rates [56]. The yield of a 10-Year treasury bond from 2010 to 2019 ranged from 1.80% (in 2012) to 3.22% (in 2011), giving an average of 2.405% [83]. Other research has used 3% as the risk-free rate, which is close to the returns of the bonds described previously, so to allow for comparison 3% will be used as the risk-free rate.

3. Drawdown: *Drawdown* is a decline in equity between a successive peak and trough [11]. Therefore, the maximum drawdown is the largest peak-to-trough decline

over a given period of time. When presented on a graph, we see the drawdown as the distance between the highest point and the successive lowest point [107].

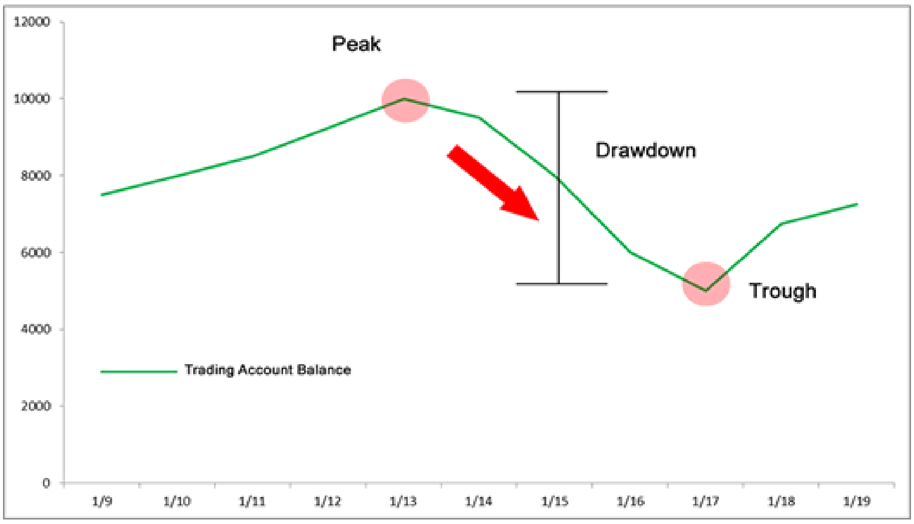


Figure 2-4: Drawdown and Maximum Drawdown

Figure 2-4 shows that the drawdown period is that from when the stock price starts to fall until the price changes direction and surpasses the initial price. Drawdown is calculated from the difference between the initial price and the lowest price during the drawdown period. There can be many drawdown periods but only one maximum drawdown for a given period of time.

2.4 Discussion

This chapter described some basic financial concepts that influenced the development of our new trading system. Firstly, we considered the EMH, which if conclusively correct would render efforts to create a trading system futile. However, while developed markets may be largely efficient and becoming more so, there is evidence that they are not completely random - that they do not follow a random walk - and inefficiencies that can be exploited for profit still exist which we can see from their exploitation by hedge funds and other market participants. However, we agree with the EMH that widely-followed technical indicators, like those described in this chapter and meant to identify repeat patterns and inefficiencies, become invalid by their being well-known. The inefficiencies that still exist must have become more complex and subtle, and may even be unique to particular stocks and particular times. This is the justification for out multiple machine learning approach, as we believe this gives the strongest chance of spotting tradeable patterns. Moreover, machine learning offers the opportunity to predict future prices rather than just the direction of the share price movement. Traditional indicators tend to focus on the direction, although traders may interpret particular indicators as being strong - the stock price will move far - or weak, but this is largely subjective. In this dissertation, we focus on machine learning techniques for predictors and will use several of them together to create a dynamic stock forecasting system. Technical indicators are only represented by the inclusion of EMA to track trends in our proposed scoring system.

Chapter 3

Machine Learning and Ensemble models in Stock Prediction

Due to enormous interest from many major companies, stock market forecasting is a popular and active area. A lot of researchers are investigating prediction models for future prices or trends of stocks with high accuracy. Despite the research activity in this area, successful models are still limited due to the non-stationary nature of data. In this section, we consider the applications of several Machine Learning techniques to this problem and the various degrees of success reported.

3.1 Machine Learning Algorithms

In this section, machine learning algorithms and the applications of machine learning to financial market prediction are explained.

3.1.1 Regression

One classic technique for continuous value prediction is regression. This takes input variables and tries to map output onto the continuous result function, finally giving the predicted values based on the function. There are two common approaches, linear and polynomial regression, explained in this section.

3.1.1.1 Linear Regression

The basic idea of this technique is to find the line that best fits the training dataset by minimising the cost function. The prediction will be made by simply computing the weighted sum of all input features, plus a constant term, the so-called *Bias*. The linear regression equation is shown in 3.1;

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{3.1}$$

where

$\hat{y}$ = predicted value.
n = number of features.
x_i = the i^{th} feature value .
θ_j = the j^{th} model parameter.
θ_0 = bias term

As mentioned above, training the model means trying to find its parameters so that the model fits the training set. To achieve this, we need a measurement value to report how well or poorly the model and the specific set of parameters fit the training data. The most common metric is *Mean Square Error (MSE)*. To train a linear regression model, we need to find the values of θ that minimise MSE. MSE of a linear regression hypothesis h_θ on a training set X is shown in equation 3.2

$$MSE(X, h_\theta) = \frac{1}{m} \sum_{i=1}^{m} (\theta^T x^{(i)} - y^{(i)})^2 \tag{3.2}$$

where

m = number of instances.
$x^{(i)}$ = a vector of all feature vectors.
X = a matrix containing all feature values .
h = system's prediction function, called a hypothesis.
$MSE(X, h_\theta)$ = the cost function measured using hypothesis h

The aim of this regression problem is to minimise the cost function $MSE(X, h_\theta)$. The process repeats until the minimum value is found. Then the line according to that cost

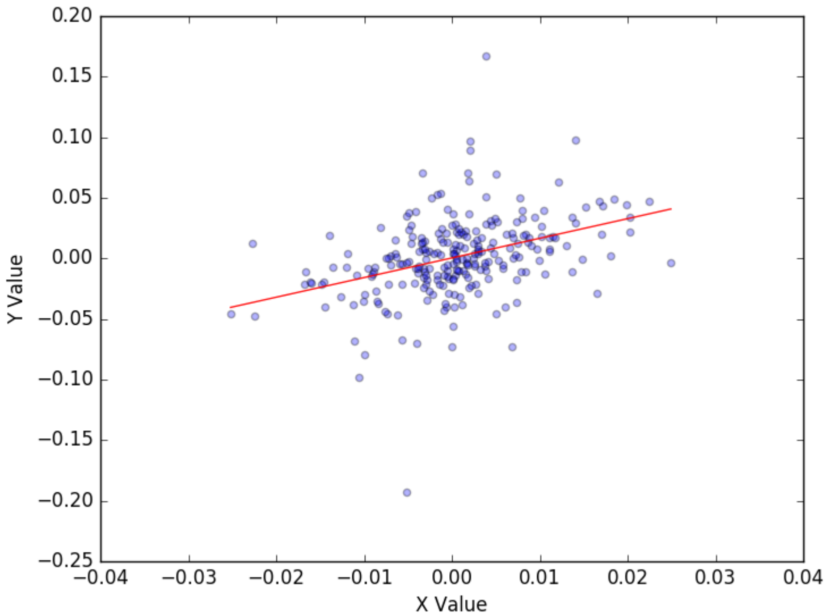


Figure 3-1: Linear Regression: TSLA and SPY 01-01-2012 to 01-01-2013

function is fitted to the data as shown in Figure 3-1, where the x-axis is the price of SPY, the index used in the prediction process, and the y-axis is the price of TSLA, the stock that we want to predict.

From Figure 3-1, the upward trend of the line of best fit indicates that the prices of the SPY index and our TSLA stock are related. If the price of SPY goes up, the TSLA price will go up even more so.

Linear regression is simple to implement and easy to understand. However, this algorithm is not good with outliers or noise [109]. There is not a lot of research in finance that uses linear regression in isolation. However, research by Bhuriya et al [16] used linear regression as a single model to predict a stock price. The accuracy from this model is outstandingly high at 0.9774. However, as they tested only a single stock - TCS, Tata Consultancy Service - over a single time period, the results might be due to luck. We found from our own experiment that stock prices are not stationary. Therefore, we focused on predicting the return instead of the stock price. Also, the results from an experiment are much more valid if they are performed on a variety of stocks or over different time frames in order to confirm the performance of the linear regression model. Having run linear regression on multiple stocks during the course of

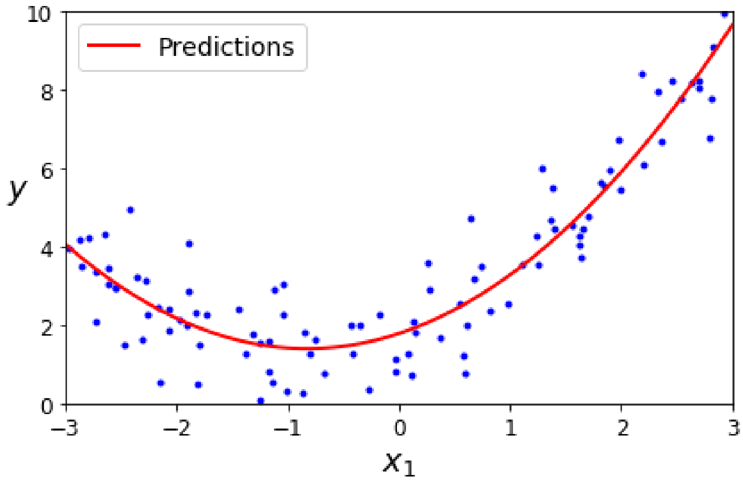


Figure 3-2: Polynomial Regression

this research, we found that testing on only a single stock is insufficient as the results could be completely different for another dataset.

A possible reason linear regression is rarely used in isolation in finance is because of the fluctuation of stock prices, which can be driven by various factors and to which linear regression is sensitive. Other models can be included to reduce fluctuations before using linear regression to make a prediction about a stock. This is the approach of our research, which clusters our data into multiple groups representing different levels of volatility, then trains multiple machine learning approaches on specific datasets. Clustering reduces the effect of fluctuations on predictors such as linear regression, which we have included as a predictor in this research.

3.1.1.2 Polynomial Regression

If the dataset is too complex to fit a single straight line, a simple way to solve this problem is to add new features to the original dataset. The new features are the powers of the original features. Then, a linear model can be trained on these added features. This is called *polynomial regression.* An example of a linear model fitting on a simple quadratic equation is shown in Figure 3-2

In Figure 3-2, polynomial feature degree 2 is used. Therefore, before trying to fit a linear model to the data, an additional feature was added to the original dataset. With multiple features, polynomial regression is able to find non-linear relationships between features.

The basic idea behind polynomial regression is similar to linear regression, which is to find the line of best fit to our data. While linear regression cannot fit non-linear data, polynomial regression is able to manage non-linear data well. An important hyperparameter for polynomial regression is the degrees of freedom. Higher degrees of freedom fit non-linear data better. However, too many degrees of freedom lead to overfitting [47]. Overfitting is when the model tries too hard to fit the training data by increasing the number of parameters. This makes the model fit the training data well, but when tested on a different dataset it works badly. So while polynomial regression can provide a better fit than linear regression, it should be used carefully by paying attention to the degrees of freedom to avoid overfitting.

In [82], researchers compared the performance of linear regression, polynomial regression and support vector machine on stocks from the US market. The methods used in the research are not complicated, starting with loading and normalising the dataset. The regression models are then trained on the normalised data. Random sampling of the stocks is used to select data to evaluate each regressor.

In order to evaluate the performance of each algorithm, two groups of data are randomly selected. The first is the stock while the second is the period of time to test. From the experimental results, this research concludes that polynomial regression is more sensitive to the normalisation method than linear regression. However, the accuracy of linear regression is lower than polynomial regression. Also, this research confirms that linear regression has less chance of overfitting the training data. Linear regression provided less accuracy at the beginning, but went on to perform better than polynomial regression, which suffered from overfitting.

For these reasons, this research suggests swapping between linear and polynomial regressions as needed to achieve a balance. This is what we would like to do with our system. Therefore, we decided to include both linear and polynomial regression into our model to benefit from the lack of overfitting of linear regression and the fitness to non-linear data of the polynomial regression.

3.1.2 Support Vector Machine (SVM)

Support Vector Machine or *SVM* is a machine learning approach suitable for small or medium-sized complex datasets [74]. SVM is capable of handling both linear and non-linear classification or regression problems.

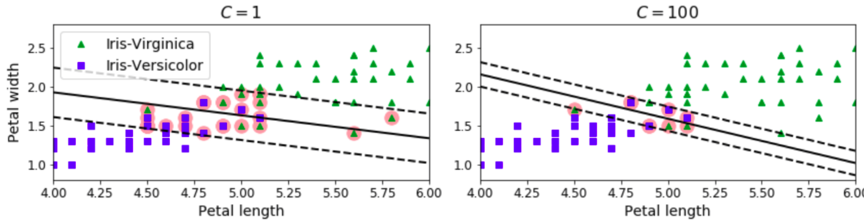


Figure 3-3: Support Vector Machine with parameter c = 1 and c = 100 [39]

3.1.2.1 SVM Classification

The fundamental idea of SVM for classification is to find the best decision boundary that can separate two classes of data. These two classes can be either linear-separable or non linear-separable as follows:

3.1.2.2 Linear Classification

Figure 3-3 represents the SVM boundary which separates two classes of the Iris dataset [39]. The boundary is represented by the space between two dashed lines in Figure 3-3. This boundary is sometimes called *street* as it looks like a street separating the two classes. If this street is too wide, there will be too many misclassified data points (data points within red circles on the street). On the other hand, if this street is too narrow, even though there are fewer misclassified data points, this boundary will be overfit to the training set, leading to poor classification accuracy on the test set. Therefore, the ideal street should be the best *trade-off* between the misclassified data points on the training set and the classification accuracy, concerning both the training and testing sets. The parameter that controls the width of this street is c. If the value of c is small (Figure 3-3 on the left), the street will be wide. On the other hand, if c is large (Figure 3-3 on the right), the street will be narrow.

3.1.2.3 Non-linear Classification

When the dataset is non-linear separable, one way to approach this problem is to add more features, for example the polynomial features as discussed in Section 3.1.1.2 on page 53. Figure 3-4 shows an example of a non-linear separable dataset. As you can see, for Figure 3-4 on the left, the two classes represented by blue squares

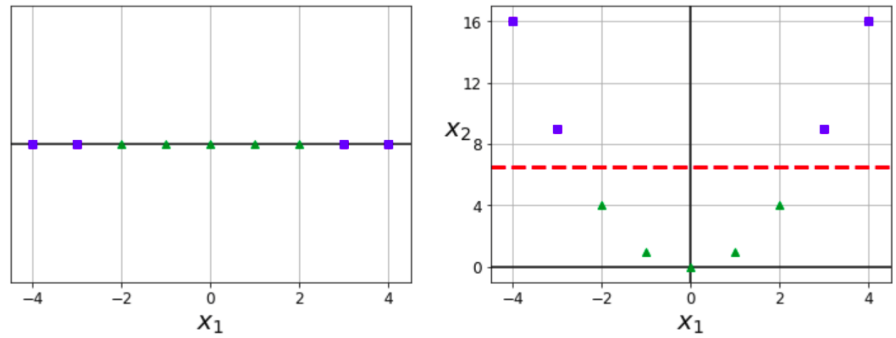


Figure 3-4: Example of a non-linear separable dataset [39]

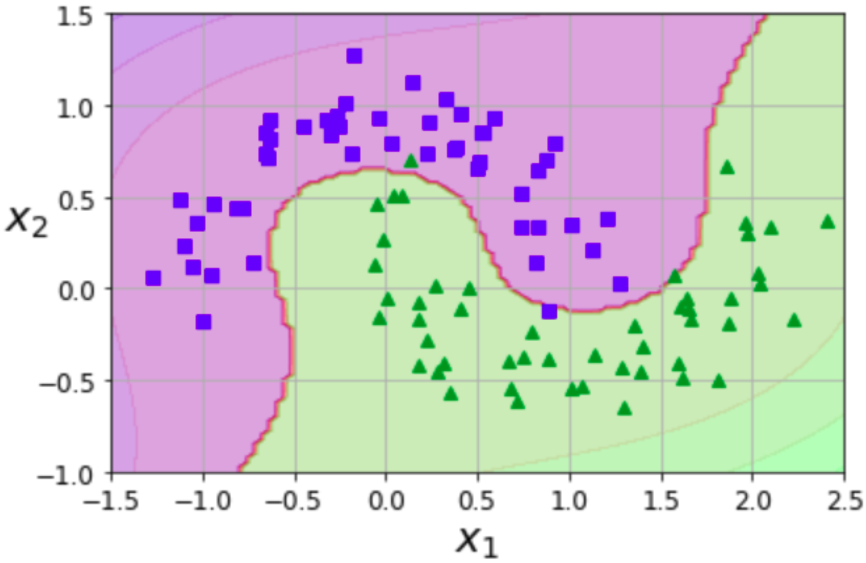


Figure 3-5: Example of using polynomial SVM degree = 3 with a non-linear separable dataset [39]

and green triangles cannot be separated by a straight line. However, when adding a second feature (polynomial feature degree = 2), the dataset becomes linear separable as shown in Figure 3-4 on the right. Another example of separating a non-linear separable dataset with polynomial SVM classification degree 3 is shown in Figure 3-5.

The benefits of SVM are that it is considered to be a robust and accurate technique which requires only a relatively small sample for training, but a trade-off for this is that it is computationally inefficient [120]. Specifically, it is robust to distributions which do not behave as expected, and also insensitive to the number of dimensions [12].

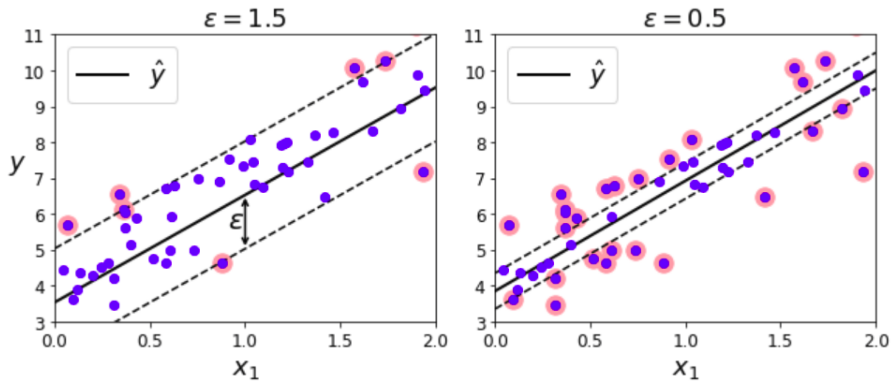


Figure 3-6: SVM Regression [39]

3.1.2.4 Supper Vector Regression (SVR)

SVM is a versatile method that can be used for both classification and regression problems [39]. The general idea of using SVM for a regression problem is the opposite of using SVM for classification. SVM regression will try to fit as many data points as possible on the street while limiting the number of misclassified points outside the street. The width of the street is controlled by a parameter ϵ. Figure 3-6 shows that a smaller ϵ leads to a narrower street and a larger number of ϵ results in a wider street.

[122], an adaptive version of SVR is proposed to cope with high fluctuation and noise. Instead of using the standard SVR, they proposed SVR with an adjustable margin. The typical loss function of SVR contains a fixed and symmetrical margin called FASM. When the margin is very small, the model is prone to overfitting. On the other hand, if the margin is too large, the model is more generalised and tends to incur more errors. The disadvantage of having a fixed margin is the model cannot adapt well to the data [108]. Their research focuses on margin adaptation in two ways. The first approach is FAAM (fixed and asymmetrical margins) and the second method is NASM (non-fixed and symmetrical margins).

In FAAM, the upper and lower margins can be asymmetric but are fixed numbers. However, in NASM, the margin can be adapted according to inputs. This adaptation is performed by a shift window. These approaches are tested on stocks from Hong Kong's Hang Seng Index (HSI) from January 2001 to June 2001. Experimental results show that the fixed margin is not helpful for prediction, even with an asymmetrical margin. However, the dynamic or non-fixed margins work well, resulting in significantly decreased errors (from above 130 to 116). In the experiment, the margin for NASM is calculated from the standard deviation of the input. Therefore, it keeps changing when a new input is obtained.

From the results, neither the upper or lower margin matter when they are fixed. Increasing the upper margin makes the error of positive class higher. However, at the same time, the error of negative class decreases. Therefore, varying these values does not help. The key point here is to deal with fluctuated data, and for this the margins need to be able to automatically adapt. This research could be improved by incorporating normalisation into this adaptive margin system, such as using two times standard deviation or another value. Then, we can see the effect of this dynamic system from a different perspective.

3.1.3 Auto-Regressive Integrated Moving Average model (ARIMA)

ARIMA is the acronym for Auto-Regressive Integrated Moving Average model. It is a mathematical model introduced for short-term predictions [3], [117]. Even though ARIMA is good for short term forecasting, it requires long-term training data to make accurate predictions [67]. The ARIMA model is composed of three main parts: Autoregressive model or AR(p), Integrated procedure or I(d) and Moving Average Model or MA(q). The Autoregressive model estimates the value of Y at time t by observing the values of time series X within the time period p. Moving average is the process that creates a q-day lead-lag dataset to observe and decrease error from the prediction process. Finally, the integrated process is performed to achieve stationarity [67]. Therefore, it is represented as ARIMA(p,q,d).

ARIMA has been the focus of much research since its introduction, some of which is described in the following paragraphs beginning with Wang and Leu's work [115] on the application of a recurrent neural network trained using features extracted from an ARIMA model. The model was trained on TSEWSI (Taiwan Stock Exchange Weighted Stock Index) data between 1991 and 1994, then tested on the 1995 data. Based on these stocks, they found that the best period for the AR and MA models to give effective predictions is one day, but they could not determine a best value for the Integrated process. An ARIMA model was also used to perform short-term predictions of the U.S. dollar [116]. In this work, an ARIMA model provides the short-term trend, rather than price, to describe the volatility of the currency. The model comprised three steps: smoothing input data, identifying parameters (p, d, f) and testing them by the addition of noise. Moreover, they investigated the model further by studying the relation between U.S. dollar and Shanghai index. A few conclusions emerged:

1. ARIMA does not cope well with fluctuating situations. When the test dataset shows a downward trend overall, but has some small segments of fluctuation, ARIMA does not work.

2. ARIMA is much more suitable for short-term prediction. In this particular case, it did not work well for periods longer than a month.

3. Other economic factors should be taken into account, such as the correlation between datasets, in order to improve the performance of ARIMA model.

Another interesting point of this work is that the authors used weekly closing prices to make predictions. It would be interesting to see how this model would work with daily closing prices, since the daily closing price is the most popular indicator and is used in most stock prediction models.

3.1.4 K-Nearest Neighbors

K-NN or K-Nearest Neighbors works by selecting similar time-series factors in the past from before the observation point which is to be forecast. Unlike other approaches, K-NN selects the nearest neighbour by focusing on relevant previous observations based on their geometry, trajectories, and levels, not their location in time. Most K-NN classifiers use Euclidean distance to measure the similarity between objects or neighbours [105]. However, it can be calculated with other distance functions too, such as Manhattan distance.

K-NN can be used for both classification and regression problems [53]. K-NN can be used to classify objects, for example it has been applied to image classification problems [42] [124]. This method classifies objects based on the distance between objects in the feature space. K-NN is an example of instance-based learning as it learns by measuring the similarity between objects. As for K-NN in regression, K-NN simply predicts the value of an object by averaging the same values from other objects, K nearest neighbours. K-NN works well for the regression problem when the contribution of data points is weighted, so the closest neighbours have more impact on the data [75].

The performance of K-NN is determined by the number of K [105] [53]. Since K is a hyperparameter, much research aims to solve the problem of the optimal number of K. In [105], the number of K is not fixed to a single value. In traditional K-NN, the number of neighbours is fixed for every data point. However, that research proposes a new approach to finding the most suitable K for each individual sample.

The proposed algorithm is able to define the optimal K, which is the minimum number of neighbours that are able to provide the correct class label for each new incoming data point. This means that K will change with different inputs during the test period. The proposed algorithm, called the Adaptive K-Nearest Neighbors Algorithm or AdaNN, starts from the training process. In the training step, instead of trying to run multiple experiments with different K (grid search) to find the most suitable K for the whole training data, this approach looks at each sample in the training set individually. Then, it finds the minimum number of neighbours that could provide the correct class from that sample. The number of neighbours in this training step starts from one to nine. If nine neighbours still can't give the correct class, then nine will immediately become the optimal K for that sample. After training, each sample in the training set is assigned the K number which suits them best. Then, when it comes to testing, AdaNN will calculate the Euclidean distance for each input to find the closest neighbours in the training set and adopt this number to define neighbours in order to get its class. In order to evaluate their model, 15 datasets from the UCI repository are used. The number of clusters of these datasets ranges from 2 to 8. AdaNN results are compared with the results from other K-NN models which use fixed K numbers (from 1 to 9). The results show that there are 3 datasets out of 15 where AdaNN overcomes the other fixed K-NN models. There are six datasets for which AdaNN comes up as the second best algorithm. Finally, there are two datasets in which AdaNN performs badly (comes 6th and 10th). The study concludes that for most of the data, AdaNN has a high possibility to overcome traditional K-NN. Even though it cannot win over every model, it often comes second place. Another conclusion is that AdaNN works better with small scale datasets

This research changes the traditional method of K-NN in classification. Even though it was not the best on every dataset, it gave good results with clearly less work making the decision on the number of K. However, one should consider whether decreasing the amount of work infantilising K at the beginning is worth this much effort to calculate the distance and find the closest neighbour for the new test data every time. This could increase the processing time substantially. And the time will depend on the size of both training and testing sets, instead of only the training set like in traditional K-NN. Therefore, it is necessary to consider further the trade-off between increasing accuracy and time consumption.

K-NN is also applied in the stock market to predict the future returns [?] [?]. [?] developed an ensemble model of K-NN and SVM. SVM is used in the first step to classify stock data while K-NN is used to predict the future return of two indices, the

Bombay Stock Exchange (BSE Sensex) and CNX Nifty. This research is composed of three main stages, which are data preparation, training and testing. The final output of the system is the future return. In the preparation step, the system calculates additional features related to the price movement, such as the simple moving average, relative strength index and average true range. Then these features will be normalised. Step 2 obtains the normalised data from the previous step and divides this data into two sets, training and testing. The training set will then be used to train the SVM regressor. Finally, step three takes the classifications obtained from SVM and applies K-NN in order to predict the future return by averaging from the K nearest neighbours. To evaluate their system, a comparison between the proposed method and other systems is shown for the two indices mentioned above. The system performance is represented by MAPE. The comparison between the proposed model and the other 2 models shows that the proposed model performs much better, resulting in much lower MAPE. For example, for the BSE Sensex index, the proposed model obtained a MAPE of 0.0650, while another model had a much bigger error at 0.23 for the 1-day prediction. After confirming that the proposed model works significantly better than the others, the researchers performed more comparisons on different periods of prediction: 1 day, 1 week and 1 month. As might be expected, the results show that the minimum error is obtained from the 1-day prediction, followed by 1 week and 1 month, respectively.

The only issue we noticed here is in the data preparation process; the additional features are calculated altogether for the whole dataset at the very fist step, then the training and testing sets are separated later. This could cause accidental look-ahead bias in the calculation, especially at the beginning of the test set. Therefore, it would be better to separate the data first and prepare them separately.

3.1.5 K-means clustering

K-means clustering is an unsupervised learning technique. It is normally used to identify similarities or dissimilarities between data points in order to compose meaningful clusters within the data [63]. K is a hyper-parameter which is set at the initialisation step before starting other calculations. It refers to the number of clusters. In the initialisation stage, K can either be randomly picked from data points or can be any random point within the range of the dataset.

X is a set of data points in r-dimensional space. Besides being the number of clusters, K represents the centroid of each cluster. To make a decision about which group a data point should belong to, the K-means algorithm will decide from the distance between it and the centroids. The data point will then be sent to the nearest centroid.

The K-means algorithm comprises three steps, as follows:

1. Initialising K-cluster centroids: This can be done by randomly picking some data points, or any points within the range of data. These K-points are taken as the initial centroids.

2. Assigning clusters to data points: This is done by calculating the distance, such as the Euclidean distance between each data point to every centroid. Each data point will then be assigned to the nearest centroid.

3. Update new centroids: In this step, the system will recalculate to find new centroids. New centroids for each group can be computed by averaging all the data points in that group.

Steps two and three are repeated until the centroids do not move any more, or when other criteria, such as a time limit, have been met.

K-means clustering has been applied to many areas, such as network anomaly detection [63], enhancing prediction performance [112][27] and data mining in agriculture [106]. In [63], Kumari et al applied K-means clustering to prevent cyber attacks. In network attacks, intrusions play very important roles as they will be the first step before damage happens. This research proved the K-means algorithm is able to identify intrusions. To run the experiments, the researchers obtained network intrusion datasets from KDD cup 1999 data which was used in a competition to build a network intrusion detector in the fifth international conference on knowledge discovery and data mining. KDD cup 1999 is the 38-dimensional data of good and bad connections. Good connections are normal traffic while bad connections refer to intrusions or attacks. Researchers found that the optimal number of clusters for their data was 150 by performing grid search. After obtaining the best number of clusters, the thresholds for each cluster were set to be used in real time testing with new datasets, such as Flume, Kafka and HDFS. Any data points beyond the threshold are considered attackers.

A paper by [112] focuses on another problem of K-means clustering: finding the best value for K [100]. As K is the number of clusters, this problem is basically how to find the optimal number of clusters. Normally, the number of K needs to be defined at the beginning of a K-means operation. K-means clustering normally uses Euclidian distance to identify clusters. This work focuses on improving the Euclidian formula to enhance the cluster quality. The new enhancement methodology is based on an ensemble of normalisation and majority voting techniques. The performance of this system is evaluated on a well-known dataset, Iris. The results from this research show better accuracy and less processing time. Therefore, the researchers confirmed that their technique can be used to improve the quality of K-means clustering in different problem spaces.

K-means clustering has several benefits: it is simple, doesn't require supervision, and versatile, with stable - if only moderate - performance across different applications [126]. Although, it has been pointed out that it is unable to handle very large datasets well, and it is also sensitive to both the initial conditions and the presence of outliers [120]. As with all machine learning approaches described in this chapter, there are advantages and disadvantages, which validates our ensemble approach as one machine learning technique can make up for the pitfalls of another.

3.1.6 Hidden Markov Model

The Hidden Markov Model (HMM) was initially developed to solve problems with speech recognition and is also used for DNA sequencing and ECG analysis. HMM is a statistical model composed of two types of state: hidden and observed. The main idea of HMM is to classify or predict the hidden state by extracting the possibility from the observed states. This is similar to a GP testing patients for a fever but being unable to check on the patients directly. The only thing the GP can do is ask the patients if they feel normal, cold or dizzy. In this case, the hidden states are *healthy or fever*, and the observed states are *normal, cold or dizzy.* The GP collects this information and performs a calculation based on statistics which will tell him the probability of whether the patient is healthy or has a fever [25].

In order to improve the performance of stock prediction, Hassan [45] introduced HMM to this area in 2009 with a model that combined HMM and fuzzy logic together to make a prediction about a stock's future. The main idea of this combination is using a HMM model to identify patterns, before creating fuzzy rules using fuzzy logic. With this technique, multivariate financial data can be predicted with more precision than previous approaches. The results from this paper improved upon those of previous models based on ARIMA and combination models of HMM with neural networks and genetic algorithms when tested on six stocks, including Apple, IBM and Dell.

HMM has also been used to predict the closing prices of four stocks: TATA Steel, Apple Inc, IBM Corporation and Dell [43]. There were two main adaptive aspects in this paper. Firstly, HMM probability is remodelled using Gaussian Mixture Models

(GMMs), and secondly, the observation states were changed from the actual open, high, low and close prices to fractions of them. Therefore, the observation states can be calculated as shown in equation 3.3:

$$\mathrm{O_t} = (\frac{\mathrm{close} - \mathrm{open}}{\mathrm{open}}, \frac{\mathrm{high} - \mathrm{open}}{\mathrm{open}}, \frac{\mathrm{open} - \mathrm{low}}{\mathrm{open}}) := (\mathrm{fracChange}, \mathrm{fracHigh}, \mathrm{fracLow}) \tag{3.3}$$

After testing this model on the four different stocks, the results from two stocks, Apple and IBM, outperformed three other models: HMM Fuzzy, ARIMA and ANN, while for the other stocks this model gave comparable results. A future improvement that might be added to this model is correlation analysis. This model assumes that all stocks are independent, yet in the real world where there may be some relations between equities, which means correlation analysis should make this model provide better results.

Another work that supports the strong performance of HMM in the field of finance is the literature survey of [102]. This paper compares HMM performance with other state-of-the-art approaches. The results of this paper show support vector machine and multi-kernel modes give 60% and 64.35% accuracy respectively, while two HMM-based models give significantly better results: the first HMM work, from Hassan et al. [45], has more than 90% accuracy, and the second paper from Gupta and Dhingra [43] gives 84.40%, 84.90%, 93.89% and 91.76% accuracy for the four stocks.

According to the previous results from many HMM models, the performance of HMM is better than other models. However, the paper does not confirm if all models were investigated on the same stocks, over the same periods, or otherwise under any of the same conditions. Hence it cannot be concluded that HMM outperforms the others because in the real word, financial data acts differently over different time periods and across stocks.

We have discussed some existing applications of HMM to financial markets, now we will discuss some general advantages and disadvantages of the concept of HMM. Its primary strengths are its strong statistical foundations, computational efficiency and ability to handle new data [46]. At the same time, it is sensitive to the initial parameters chosen to build the HMM structure, which if poor will result in bad performance. We also discussed how HMM assumes independence between stocks, which may not hold in reality.

3.1.7 Artificial Neural Networks

Because market behaviour is dynamic, chaotic and complicated [9], it is difficult to analyse and forecast with analytical models. The nature of a Neural Network makes this method effective for predicting financial time-series data. Artificial Neural Networks (ANN) have also been successfully applied to stock market prediction, however some disadvantages have been identified, for example performance is limited in some market situations [45].

A Neural Network (NN) is a data-driven algorithm. It can discover non-linear relationships within datasets without prior knowledge of the input-output relationship. An NN is structured in layers and is normally composed of two or more, typically input, output and hidden layers. The input will be fed into the neural network at the input layer, then these inputs and its weights will be sent to the hidden layer(s). At the hidden layer(s), weights will be recalculated and adapted. Then the output from the hidden layer might be recurrently fed back into the input layer until the result is obtained later at the output. However, this depends on the type of NN. Some NN have only two layers, input and output, for example the Self-Organising Map Neural Network (SOM). SOM will take input data at the input layer and the calculation will be (re)performed at the output later until the result is obtained or the groups of data are found [118].

Figure 3-7 shows the structure of a multi-layer Neural Network. This network comprises an input layer, an output layer and two hidden layers. The characteristics and patterns of input are captured in these hidden layers.

There are many publications with different levels of success on financial forecasting based on Neural Network models. Zhou and Hu [128] proposed a novel time-series pattern-finding program using a Neural Network model. The model has two main steps, feature extraction using a sliding window method and pattern recognition using an Artificial Neural Network. Moreover, in order to decrease the computational running time, the PIP (Perceptual Important Point) was incorporated within the feature extraction process. This model was evaluated with Hong Kong, USA and China indices' data over five years. The results showed that the proposed method can recognise some price patterns well.

[103] is very positive about ANN as an approach, although not specifically for its application to stock market prediction. Some benefits they highlight are ANNs' abilities to self-organise, recognise patterns and handle difficult and complex data. However, ANNs can suffer from overfitting problems as they can become too dependent on the

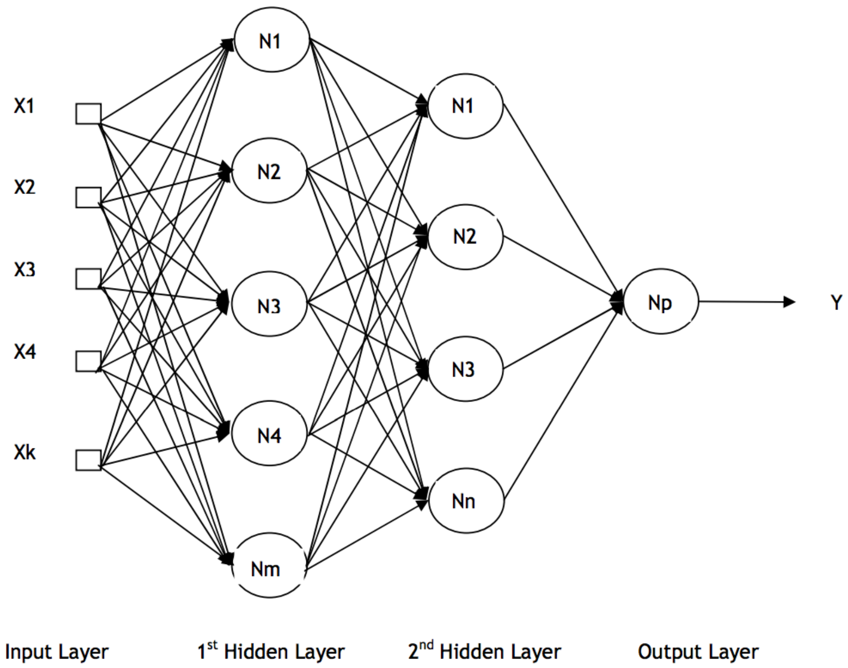


Figure 3-7: Muti-layer Neural Network structure

samples observed. Moreover, they require relatively more data and processing time [125].

3.1.8 Random Forest

Random forest is an ensemble of decision trees. Trees in the random forest are trained using different pieces of data [48]. Two popular techniques to train random forest are bagging and pasting. These two techniques are widely used to train the ensemble models by dividing the training data into smaller parts and deciding which parts are to be used to train which tree. Basically, each tree will be trained on the same data but on a different random subset. With bagging, the training data will be sampling with replacement [21], while pasting is sampling without replacement [22].

Random Forest is one way to avoid overfitting by a decision tree as it builds multiple decision trees instead of one and lets them get involved in the prediction or classification by voting. Random Forest has been widely applied in many areas, such as feature selection [64], solar components forecasting [13], short-term load forecasting [33], and the stock market.

Research by [61] proposed a new method for reducing prediction errors in the stock market. The key idea is to manage the prediction problem as a classification problem. This method will minimise risk in the stock market with an ensemble-based model of decision trees, a random forest. Firstly, the historical data is smoothed by taking the exponential moving average in order to remove noise or random variance, then multiple technical indicators are created as extra forecasting features, such as stochastic oscillator and moving average convergence divergence. These features will be taken to train the random forest classifier. With this method, the model is able to produce signals to buy or sell a stock with high accuracy for multiple stocks, such as Samsung, Apple and GE. The accuracy this system provides is outstanding, more than 90%. The accuracy is also compared with other machine learning algorithms, such as support vector machine and logistic regression. However, there is no report of other financially relevant factors, for example the Sharpe Ratio, profit or any risk values. This is very unfortunate as in finance it is necessary to report a strategy's performance in terms of return and risk.

Further research into the stock market is done by [96]. This research, like the previous example, aims to be used for stock prediction. However, this research predicts the value of a stock, not the direction only. The premise for this research is that classification and regression trees (CART) are mainly built to learn extremely irregular patterns, and this makes the trees are very prone to overfitting. Even a little noise or an outlier can affect the structure of the trees greatly. This problem can be overcome by using several CARTs which are trained differently. This research proposed a new method to improve performance by combining answers from multiple CARTs. The proposed model is Least Square Boost-based Random Forest or LS-RF. The model improves the performance of the ensemble CARTs by reducing the variance and the overfitting of the trees or CART. This model was tested on two market indices, CNX Nifty and S&P BSE Sensex. There are multiple evaluation values, such as MAPE and MSE. The results are shown as predictions 1-10, 15, 30 and 40 days in advance, which is quite a long time. The results show that at every period of prediction, their model performs much better than the support vector machine. However, there is no clear reason why the model was compared with support vector machine only. Comparison with multiple models would have been better to give an idea of what kind of models this system can overcome.

A drawback of Random Forests is that there is a threshold number of trees which, when crossed, offers only marginal improvements if any, but increases the computational cost [84]. However, an important problem in this area is working out how many trees to include in the Random Forest, similar to the need to find the optimal value for K in other machine learning techniques. We are including Random Forest as it has proven high accuracy over a range of dataset types [123], it is robust against outliers, errors and random variations in a dataset, and it is fast to construct so processing speed is good [111].

3.1.9 Genetic Algorithm

Genetic Algorithm or GA is a very useful method in many areas of research, including financial forecasting. It provides many possible solutions across the problem space. The common processes of GA can be described as follows:

1. Initialisation: in this step the initial population chromosome is created.
2. Evaluation: calculate the fitness of every offspring in the new population.
3. Selection Process: parent chromosomes are selected based on probabilistic according to the chromosomes's fitness.
4. Modification: apply cross-over and mutation to the parent chromosome to produce a new generation.
5. Assigning: set the new populations to be the new parents by considering the maximum fitness.

Steps 2-5 are repeated until the desirable fitness value or maximum fitness of the new population does not change any more. GA is a robust algorithm [**?**] There is possibility to obtain the results close to the global optimum with GA if it is allowed to run long enough. However, this is not guaranteed. GA can be effective but it is hard to explain as its processes are a black box once the input has been fed in.

What follows are some GA applications to financial forecasting, starting with GA applied by itself and followed by combinations of GA and other state-of-art machine learning techniques. Mallick et al. [72] proposed an automatic trading-rule generation system based on GA. The goal of this work is to create trading signals - buy, sell or hold - automatically. The model was tested on the DJIA (Dow Jones Industrial Average) and its performance evaluated by comparison with a start-of-art rule creation method called MACD (Moving Average Convergence/Divergence). The results show that GA performed better than MACD on more than 75% of the stocks that this research tested.

Another financial prediction model based on GA was published by Matsui and Sato [72]. This paper focused on identifying the best genotype coding method which needs to be done at the beginning of GA. They indicated that simple binary chromosome coding was not the best way to encode GA chromosomes. Therefore, two new chromosome encoding methods, called Locus-based and Allele-based representations, were experimented on. The experiments were performed on four types of technical indicators,: SMA (Simple Moving Average), EMA (Exponential Moving Average), BB (Bollinger Band) and PCB (Price Channel Breakout). The datasets were 10-year historical stock prices taken from 20 different stocks in Tokyo. The experimental results show that the best genotype representation method was the Allele-based method because it provided the largest profit and lowest computational cost.

The main advantages of GA are its effectiveness at dealing with non-stationary data (like financial market data) [32]. Moreover, it does not require much information to search large, complex search-spaces effectively [18]. However, GA can be time-consuming and computationally costly.

3.1.10 Other models

In this section, further methods and models that combine more than one Machine Learning techniques will be described in order to investigate the performance of combination models

Another model that incorporated GA in financial prediction was introduced by [50]. They proposed a model that combines GA with reinforcement learning technique in order to create the most suitable algorithm for the Foreign Exchange market. The main idea of the model was selecting the best financial indicators for predicting sell and buy signals in the Foreign Exchange market. It starts by setting a base indicator and applying more indicators. It continued performing based on GA and Reinforcement learning until it could no longer increase the performance of the system. This model was evaluated on the 5-month USD dataset and provide 49% profit. Even though the researchers claimed that the model was effective, it was only tested on one dataset at one period of time. Therefore, it would be better to confirm the model with other different datasets and time periods.

Leu and Chiu introduced a novel combination model of GA and Fuzzy logic in order to construct the maximum return portfolio [66]. The main task of this model is selecting stocks to add to the portfolio. In this model, GA was used to build the optimum portfolio and Fuzzy logic takes part in the return prediction. This model also has a stop-loss system to control risk for the portfolio. This executes an automatic sale of stocks when a predetermined lower price limit is reached. To evaluate the model, 7-years of data covering 50 Taiwanese index stocks (TAIEX) was used. The period of portfolio adaptation for the test is a month. The result was compared to the common buy-hold policy and the market benchmarks and shows that the proposed model performed very well with significantly improved return about 28%, while the other models gave below 10%.

Another Combination of GA model is the application of corporate Support Vector Machine and GA which was introduced by Cheng and Shiu [28]. The aim of this research was to create a method for selecting the best financial indicators to forecast stock trends. There are two main processes in this model. The first is the indicator selection process. In this process, financial indicators will be selected by using multivariate adaptive regression splines and stepwise regression method. The second step is trend prediction using SVM. In this process, GA is used in order to optimise SVM' s parameters. The model was tested on the Chunghwa telecom stock price during the years 2003 to 2012. The experimental results indicated that this model only works well for short-term prediction and also is not suitable for fluctuating datasets.

There are stock prediction systems using multiagent strategy. Davis et al [30] introduce a stock prediction system that incorporated many agents which are collecting information from different sources, both fundamental and technical information, for example, news, trading data and comment from analysts. This framework can be used to manage a portfolio. However, it is still need to be improved in the knowledge representation process. Another model with agent-based learning is the combination of Multi-Agent and GA model which was proposed by Schoreels et al. [94]. The objective of this model is different from Davis's work in that this model aims to generate the signals that make the best profit, instead of constructing a portfolio. The system comprises many agents which deal with different trades. The decisions are made by considering six financial indicators. GA takes part in the task of keeping the good agents and removing the unprofitable agents. The model was tested on three groups of German index (DAX) data and the results showed outstanding returns with more than 175% during the year 1998 and 2003.

Recent research has introduced a new inverse approach in sentimental analysis in order to generate profit in stock market. Birbeck and Cliff [17] applied a new sentimental labelling method to Twitter posts. Instead of labelling the data by the true sentiment of the posts, for example positive or negative, they reversed the idea by labelling them on the ground-truths or the price of the related stock at the time. Therefore, the collected Twitter posts were labelled as buy or sell based on the movement of the stock prices. In order to make predictions, three machine learning models - Support Vector Machine, Naive Bayes and Logistic Regression - were applied. To evaluate their system, they selected four well-known stocks, which were Apple (AAPL), Tesla (TSLA), Twitter (TWTR) and Facebook (FB). The datasets were originally collected over two years as it was expected that data further in the past would be less useful for the prediction, then they later decreased the time frame to three months based on their experiments. The evaluation process was executed from January 2017 every trading day from 10am to 3pm. To show their system's performance, the monthly return rates after estimated trading fees for AAPL, TSLA, TWTR and FB were 4.48%, 8.27%, 0.04% and -4.08%, respectively. These results will be compared with our model in chapter 6, section 6.4.

The results are promising, however, the number of trade orders is relatively high: can be up to 120 trades per month for every stock. If trading with this maximum number, this system buys and sells 6 times everyday on average, which is not convenient for non-professional traders with low initial funds due to brokerage fees and the speed at which they can place orders through brokers. Also, even though these results are promising, they've only been tested on technology stocks, which normally have a high Sharpe Ratio, high volatility and small drawdown in general. It would be better to see the performance of this model different sectors of the market which might exhibit different characteristics.

3.1.11 Discussion

Section 3.1 demonstrated many machine learning techniques that have been applied to financial prediction. Due to their ability to understand and learn from large quantities of data, machine learning approaches have been applied not only to predict prices, but also to select indicators and predict trends. Two well-known regression techniques, ARIMA and KNN, have been successful in their predictions of short-term future prices, as well as HMM techniques, as demonstrated in the CARIMA model [117], which provided less than 3% prediction error for the majority of tested stocks and [43] more than 90% prediction accuracy using HMM techniques. ANN has not only been used for price prediction, but also price pattern recognition as in [128], where the model recognised diamond and bump & run patterns. GA was applied to predict the future prices as well. Even though the GA-based model did not provide high accuracy, it still outperformed using financial indicators alone, as in [72]. Finally, other combination models, including multi-agent models, were presented. In the multi-agent system, agents make predictions by collecting data from different sources, as can be seen in [30] and [94].

Machine learning techniques have often been applied in combinations, rather than individually, in order to achieve better results. For example in [45], HMM was used for identifying patterns, and fuzzy logic for creating the trading rules. However, the results from most of the ML models are still attributable to specific times, locations and situations; they are not representative of all financial data and may not perform well under different circumstances, as will be discussed further in the Section 3.2. For that reason, this research aims to create a system that can dynamically select different models depending on the situation. The proposed system will be able to deal with all the main market situations, bull, bear and ranging, by applying multiple machine learning models. The model selected as the main indicator will vary appropriately to different situations.

The table 3.1 below summarises the advantages and disadvantages of several machine learning approaches which have been discussed in this section.

Table 3.1: Performances summary of FERG comparing between our model and the buy & hold strategy for the test period of year 2014 after optimisation

Algorithms	Advantages	Disadvantages
Linear Regression	Easy to implement and understand [109] Less chance of overfitting [82]	Not work well with non-linear data Not good with outlier or noise [109]
Polynomial Regression	Work well with non-linear data [47]	Easily to overfi[47] Sensitive to normalisation [82] Have to deal with hyperparameter
Support Vector Machine	Robust and accurate [120] Requires small sample for training [120] Insensitive to the number of dimensions [12].	Computationally inefficient [120]
ARIMA	Work well for short term forecasting [117]	Sensitive to stationarity of data [67]
K-NN	Simple and easy to implement [105]	Have to deal with hyperparameter [53] [105]
K-means	Simple, versatile and stable [126]	Have to deal with hyperparameter [63] Sensitive to outliers [120]
HMM	Strong statistical foundations [46] computational efficiency [46] ability to handle new data [46]	sensitive to the initial parameters [46]
ANN	Handle difficult and complex data [9] Abilities to self-organise, recognise patterns [103]	Require relatively more data and processing time [125] Suffer from overfitting [125]
Random Forest	Not easy to overift [13] Has proven high accuracy [111].	Have to deal with hyperparameter [84]. computational cost [84].
GA	Can provide result close to optimum [32] Robust and versatile [32] Dealing with non-stationary data [32]	Black box algorithm [18] Time-consuming [18] Computationally costly [18]

3.2 Ensemble model

Financial time-series prediction is considered to be one of the most challenging fields in time-series forecasting. Many researchers have investigated this topic with hopes of making and improving profits from short-term investments [72], [9], [69]. Researchers and traders have also tried to use Technical Analysis as a tool for analysing and predicting stock behaviour. Machine learning has applied in order to forecast more consistently and accurately the future prices of stocks, and subsequently stock market movement prediction become an area of research that receives a lot of attention. Common algorithms, for example artificial neural networks [9], [4] and Support Vector Machines [7], [51], have been widely applied in this area but there is difficulty in selecting the best machine learning techniques to suit different market situations, such as extended uptrend (bull), extended downtrend (bear) or when there is not clearly defined trend (ranging). To overcome this problem and improve accuracy over state-of-the-art machine learning algorithms, it has become popular to combine them, forming so-called ensemble systems [31].

A simple ensemble model was proposed in 1974 for handwritten-digit recognition [104] in which the method counts the votes or answers of many artificial models and takes the majority as the final result. Ensemble models can be divided into two categories, non-hybrid and hybrid models [57]. The non-hybrid ensemble model comprises versions of the same model with different parameters. On the other hand, the hybrid model comprises different models. According to Dietterich [31], the ensemble model is an improvement over individual classifiers because the model can receive a good approximation of an answer, especially when the question cannot be satisfied with only one hypothesis.

3.2.1 Basic Ensemble Framework

Common ensemble models in classification tasks contain the following components;

1. Training set: A labeled dataset constructed for training an ensemble model. The training set usually represents attribute-value vectors. A represents an input set composed of n attributes: $A = a_1, ..., a_i, ..., a_n$, and y represents the class variable or target attribute.

2. Base Inducer: An algorithm that receives a training set and models a classifier (predictor) which generalises the relationship between an input and a target at-

tribute. The notation can be $M = I(S)$ for representing a classifier M which is induced by an inducer I on a training set S.

3. Diversity Generator: This component will control the diversity of classifiers.

4. Combiner: This component combines the results of classifiers.

The following sections describes some common ensemble techniques.

3.2.2 Majority Voting

With this technique, the class that obtains the highest number of votes is the answer [69]. This method is also called Plurality Vote (PV) or the Basic Ensemble Model (BEM). The mathematical specification of this method can be written as [93];

$$\text{class(x)} = \arg\max(\sum_{K=1}^{n} g(y_k(x), c_i) \tag{3.4}$$

where $y_k(x)$ represents the classification of classifier K and $g(y, c)$ is 1 if $y = c$ or zero if $y \neq c$

Hansen and Salamon [44] use Majority Voting in conjunction with a neural network ensemble model for improving performance of the classification. In this model, several different neural networks were trained separately and their results combined by using simple regression and majority voting. To optimise the network parameters and architectures, this model applies cross-validation to select the best design. Errors in prediction are reduced by applying the ensemble of the different networks. The model is tested on the simple linear clustering problem with different neural network and noise levels. The results indicate the neural network with six hidden layers provides the best performance for this problem space.

3.2.3 Stacking

Stacking tries to obtain an accuracy by using a meta-learner by trying to identify whether classifiers are reliable [93]. In this approach, an ensemble model works in layers. The results of individual classifiers from the first layer [69], called base classifiers, will be taken and used as input for different classifiers at the second layer. The stacked approach is different from bagging and boosting because stacking does not combine the results of the same type of base classifiers, but combines those of different algorithms [57].

Merz [73] combines stacking and nearest neighbour in order to classify the unseen data. Dzeroski and Zenko [34] proposed an enhancement of the stacking approach. The model employs a multi-response technique tree to learn at the meta-level. The performance of this model is tested using cross-validation and shows that the new model is better than existing stacking approaches in terms of enhancing forecasting accuracy by about 5%.

3.2.4 Bagging

Bagging stands for bootstrap aggregating. This method aims to create different subsets of the training data which are selected randomly [69] and trained with different predictors. Then results from every predictor are selected by simple majority vote [57]. The bagging strategy is commonly used to construct an ensemble model of decision tree, neural network and linear regression [127].

Bagging is commonly incorporated with decision trees or random forest, as can be seen in Pradeep and Vahida's research [69]. The objective of their model is to construct an ensemble model to predict future prices of gold and silver. Their ensemble model includes many state-of-the-art classifiers, for example Support Vector Machine (SVM), Logistic, Decision tree, NaiveBayes, and Multilayer Perceptron. They then compared the performance of different techniques, bagging and majority voting, with the historical gold and silver prices. The experimental results show that majority voting was the least effective technique with prediction accuracy of only 50%, while bagging and stacking performed significantly better with accuracy between 70% and 90%.

Bagging has also been used as a benchmark for other ensemble algorithms. In 2010, [121] Wei et al. introduced a new constraint bagging method, which can reduce errors in the prediction process. In comparison to the traditional bagging model, constraint bagging is slightly better in terms of Mean Absolute (MAE) Error which decreases about 1.22%.

3.2.5 Upper Confidence Bound and Multi-armed bandits

Upper Confidence Bound or UCB is a popular method of online recommendation system. There are two main approaches in the recommendation system, offline and online.

With the offline approach, the system performs a classical machine learning method, either classification or regression, on the historical data, then exploits the knowledge or policies it found in order to predict or recommend the next item for the user in the future. On the other hand, the online approach can be considered a reinforcement learning problem or multi-armed bandit problem [78]. The system makes suggestions based on the user's current situation, which means that to make a suggestion about a new item, the system will take into account the current preferences that a user has.

THe multi-armed bandit problem allows machine learning to learn from data that was gathered while it was running (or during the testing period). The name, multi-armed bandit, is obtained from the casino situation [101]. In the casino, there are a multiple slot machines with different payouts. The gamblers intend to maximise their outcome (money). However, they never know in advance the probability of each machine. The problem is how they can select the correct machine that is going to make them rich.

One of the easiest methods to resolve this is to spend time trying every machine and collecting statistical data until they are able to identify the best machine (if possible). However, this could take a lot of time and, more importantly, money. There is a high possibility that the gambler runs out of money before finding the best machine. This is the same problem as the online recommendation system. Imagine a campaign for a new product about to be released to the market. The advertising team has come up with 10 different posters to be used to promote their new product. How can they identify which is the best poster and attract the highest number of customers. If they have to run A/B tests for all of them, it could take a lot of time and cost a lot of money, especially in opportunity cost. With this type of problem, UCB can be applied to speed up the process and reduce the cost since it is able to manage the balance between exploitation and exploration [8]. The exploitation is the process of applying knowledge or policies that the system already knows (making use of the poster that the system already knows will attract customers), while exploration is the process of expanding or randomising for new options (occasionally using a different poster) with the hope of obtaining new policies which lead to better performance. This makes sense, as an algorithm intends to gain maximum profit, so it repeatedly maximises the reward (exploitation). However, by repeatedly trying to maximise the reward, the algorithm's overall knowledge is limited. Therefore, exploration is needed occasionally in order to improve knowledge.

UCB focuses not only on exploiting the best arms and exploring other arms randomly, but on tracking on the arms' reliability. Therefore, it takes into account the arms' confidence. UCB estimates the arms' reward and confidence based on the size of the past experience of each arm [79]. Throughout the testing, the UCB method will select the arm which has highest upper confidence bound, which is defined by equation 3.5.

$$U_{K,t} = \mu_i + \sqrt{\frac{\alpha \log t}{N_i}} \tag{3.5}$$

where

μ_i = mean of arm i.
N_i = number of times that arm i has selected up to time t
α = coefficient

The upper confidence bound in 3.5 is calculated from an estimation of the mean reward and the confidence of the arms. Therefore, UCB is composed of two steps. Firstly, UCB will select each arm once. At this stage, the selection is an exploration to ensure that every arm is selected at least once. Then, the confidence bounds for all arms are initialised. Secondly, the confidence bound of each will be adjusted.

In [78], researchers applied UCB in order to enhance product recommendation in an online advertisement system. The original idea of this research comes from the multi-armed bandit problem. Instead of using the standard procedure, such as ϵ greedy, which only focuses on exploitation of the best arm and randomly exploring other arms, UCB takes each arm's confidence into account. This research presents a new recommendation system based on UCB, called UCB-RS. The main objective of this new method is to better deal with non-stationary and large state multi-armed bandit problems. The idea of this method is to use the recommendation system to estimate the arms' rewards. To estimate the rewards, the recommendation system takes the information correlated to the arm such as, when a user clicks on an item, automatically referring the user to another item in a similar category by considering other customers' historical behaviours. With this inference technique, the system estimates higher rewards for those products too. This method is able to enhance the performance of UCB in the online recommendation system by increasing the click through rate from 20% when using ordinary UCB, to 40%.

3.2.6 Ensemble models in financial prediction

Ensemble models have commonly been used in pattern recognition, for example handwriting, image and signal recognition. However, recent research has applied ensemble models to financial prediction in the hope it will to improve the accuracy of predictions despite the dynamic and complicated nature of financial data. There are two types of ensemble model application: hybrid and non-hybrid. There are several machine learning approaches that have been applied to create ensemble models. The neural network is one of the common techniques that used to create both hybrid and non-hybrid ensemble models [2], [4]. Other machine learning techniques, such as decision tree or random forest, have also been applied to create ensemble prediction models [41].

Abdullah [2] and Xu [121] proposed neural network ensemble models to predict financial time-series data. Their two models are non-hybrid with only Neural Network based predictors. Abdullah's model [2] employs multilayer neural networks to predict the next-day's stock prices. The performance of this model was evaluated with the Kuala Lumpur stock exchange and compared to the performance of a single neural network. The result from the ensemble model was slightly more accurate than the single network, about 0.5% per trade. In my opinion, since the accuracy is not much improved, in order to confirm the performance of this model time should also be taken into account since the architecture of the ensemble model is more complicated, but it is not mentioned in this publication.

Xu's algorithm [121] aims to reduce errors in the prediction process. This model is a non-hybrid neural network ensemble. All classifiers are neural network model with different architectures. They have their own, optimised architecture and parameters obtained from the bagging method. The method was improved by adding the constrained condition to leave out weak classifiers during the process. This model was tested on the Dow Jones Index (DJI) and compared with traditional classification and bagging approaches. The results show that the proposed model reduces the MAPE (Mean Absolute Percentage Error) from 14.8 % to 3.95%, comparing to a traditional neural network model.

There is not much research on financial prediction using an ensemble model. Now, we present the few ensemble systems that combine different machine learning techniques to preform financial forecasting. A recent hybrid Neural Network-based ensemble model was presented by Al-Hnaity and Abbod [4]. This ensemble model, named EEMD, worked by combining back propagation neural network with Empirical Model Decomposition (EMD). The aim of this work was to predict the closing price of the FTSE100 index. The model of this proposed ensemble system can be seen in Figure 3-8;

As shown in Figure 3-8 , the model uses EEMD to decompose stock index data and send it separately as input to the next stage, called Intrinsic Mode Functions (IMFs).

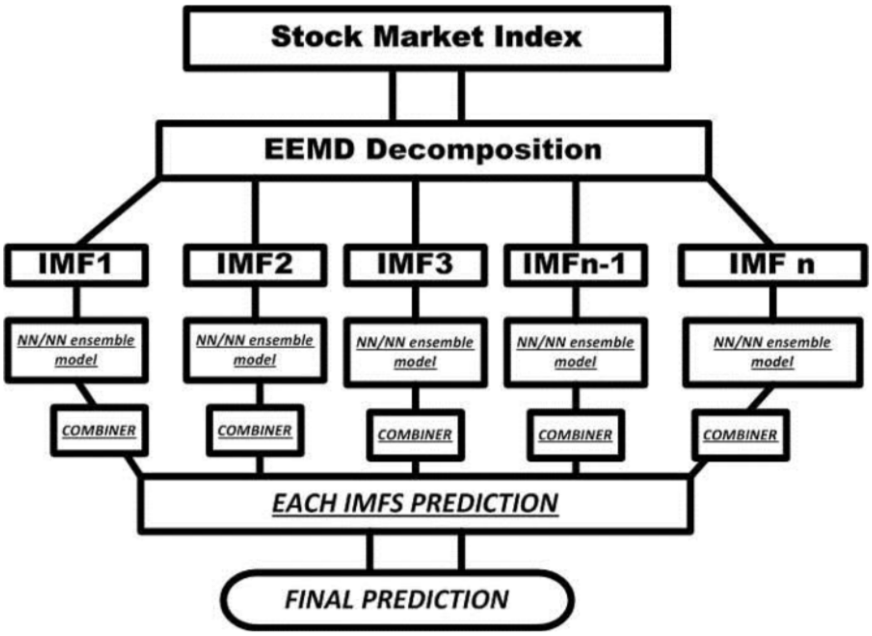


Figure 3-8: The NN-based EEMD model
[4]

Each IMF comprises one or more neural network. This step also combines the ensemble results before being sent to the final prediction stage, which works by using a weighted average combination function. A GA (Genetic Algorithm) is used to optimise the combiner's weights in this step. The model's performance is tested against the FTSE100 index. The results show the model produces less prediction errors, while a neural network has the worst performance with an almost 50% error rate.

Similar to Al-Hnaity and Abbod's reseach [4], in terms of using neural networks to construct a hybrid ensemble model, Asad [7] and Anish [6] also applied neural network techniques to create their model. Asad [7] proposed an ensemble model based on three different machine learning techniques: support vector machine (SVM), random forest and neural network. The objective of this model is to optimise portfolio allocation. The inputs of this model rely on technical indicators instead of historical prices. The result shows that after 67 iterations, this model can construct the high correlated portfolio which including a stock from Istanbul market (ISE), an index of Istanbul market (ISE-500) and an index of the German market.

Genetic Algorithm (GA) is another technique used to construct ensemble models to predict stock prices. Gonzalez et al. [41] proposed a novel ensemble model of GA, named GAENSEMBLE, to predict the weekly direction of stock prices. This model uses SVM as the main classifier and GA as an optimisation tool. Figure 3-9 illustrates the architecture of this model.

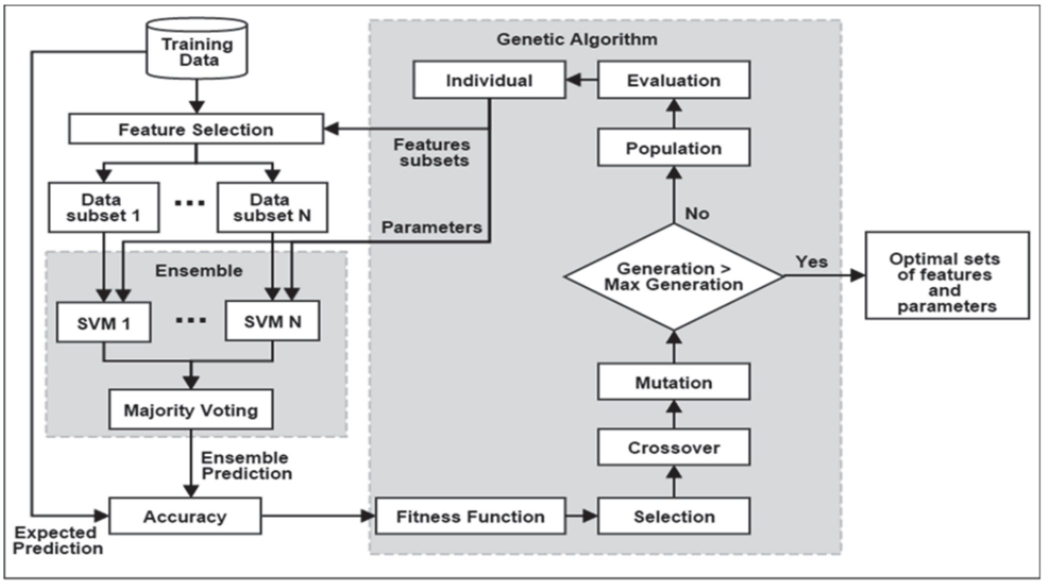


Figure 3-9: Architecture of GAENSEMBLE system
[41]

The performance of this model was evaluated with the stock indices of many countries, for example; S&P Index (U.S.), FTSE100 (UK) and Nikei225 (Japan), and some international currencies, for example; USD, EUR and CNY. The results were compared with other models, bagging and AdaBoost, as well as random forest and support vector machine and show that two of the ensemble models, GAENSEMBLE and AdaBoost, were the best in terms of the accuracy. The model provide more that 75%, while other models only give between 65% to 69% accuracy.

Recent research has applied machine learning to financial prediction with the aim of improving accuracy. Qin [89] proposes a prediction system based on gradient-boosted random forest. The model uses a scoring system for the indicators that are the input for the prediction. Indicators with better performance get higher scores and more weight when fed into the predictor. The model is evaluated on 9 stocks and 1 index from the Singapore Exchange (SGX) and compared with the buy-and-hold strategy. Although Qin's research also uses a scoring system, its use differs in that ours assigns scores to the predictors, whereas Qin assigns them to the financial indicators which are the features of the predictors. Also, our model is ensemble based, while in Qin's there is

Table 3.2: Result Comparison with Related Work [89]

Algorithm	Yearly Profit (%)	Correction Prediction (%)	Sharpe Ratio	Drawdown (Max)
Buy & Hold	-1.94	-	0.01	-47.34
Gradient Boosted Random Forest	25.14	30.25	0.03	-10.06

no coordination of predictors. Qin's results are listed in Table 3.2 and compared with our model in chapter 6, section 6.4

3.2.7 Discussion

Many machine learning techniques have been explored to create hybrid and non-hybrid ensemble models. For non-hybrid models, neural networks seem to be more popular, while hybrid ensembles exhibit a greater diversity of prediction models, such as Support Vector Machines (SVM) or Genetic Algorithm (GA). The results from the ensemble models are normally compared to prediction models based on a single Machine Learning technique. However, when designing new ensemble models in the future, comparing performance between different ensemble models could be analysed.

The objective of our research is to create a new model for stock prediction, based on machine learning, which can offer effective predictions of stock prices under any market conditions. As can be seen in section 2.2, it is difficult to achieve such a system with only one predictor that works well with different situations. The ensemble model can be a suitable way to create a dynamic prediction model based on many types of machine learning algorithm. Such a model can enhance the accuracy of the prediction and remain accurate under a variety of market conditions.

3.3 Prediction Market

Prediction Market was introduced in 1988 at Iowa Electronic Marketplace (IEM). The first purpose for releasing this model was to bet on the political elections. A prediction market is commonly used to predict the future outcome by aggregating the results from the environment or the crowd. So it can be said that the Prediction Market forecasts by analysing and summarising the wisdom of the crowd [91]. After its introduction, Prediction Market has been applied to many prediction problems, for example political voting and sport outcomes [80]. Research suggested the outputs from prediction market even more accurate than the predictions of experts [14]. The main concept of wisdom of the crowd is to let populations with different backgrounds and information make predictions on future events on the basis that people with diverse knowledge can provide high-performance solutions to a huge array of problems [49], [15].

The common process of Prediction Market starts with each trader or participant betting their money on the outcome of future events, based on their background knowledge. Every event's outcome has a separate security associated with it. So there are costs and gains in every bet. If a trader made the right prediction, they will gain money. On the other hand, if they made an incorrect prediction, they will lose the money they bet. In the next round of the prediction, a trader with more money can bet more money. Eventually, the weak traders will no longer be able to participate in the system.

3.3.1 Applications in Prediction Market

Even though Prediction Market is widely used in many areas of forecasting, applications to stock price prediction remains limited. In this section, the relevant prediction market applied to financial area will be analysed.

The Hollywood Stock Exchange (HSX) [1] and Iowa Electronic Market (IEM) [2] are popular online prediction market websites. HSX utilises Prediction Market to forecast the future profits of movies, while IEM has an online website where people can buy or sell contracts based on their belief about the election or other outcome. As for in business, a prediction market is used by some companies in order to make valuable decisions to keep information classified [20]. Hewlett-Packard or HP [3] is an example of a company that utilised Prediction Market to make important company decisions.

In terms of financial systems, Velic et al. [110] propose a new model for stock rating prediction using wisdom of the crowds, which assumes that a group has better knowledge than an individual alone, while an individual can help improve group performance by providing information. The researchers present the main algorithmic implementation but do not disclose the mathematical model. This model works as a game. In the game, there are many players that have to give ratings for stocks. During the game, there is a function, called *keep it fair* to control the game by limiting time span after start affecting player's or minimal number of active predictions for a stock to have a rating. The performance of this model was evaluated on 130 artificial stocks with 47 players. The result shows top 10 stock as can be seen in Figure 3-10;

[1]http://www.hsx.com/
[2]https://tippie.biz.uiowa.edu/iem/
[3]http://www.hp.com

Ticker	Rank	1 Y (%)	1 M (%)
VDKT-R-A	95.83	238.83	37.20
TISK-R-A	94.44	66.47	16.04
PBZ-R-A	93.06	10.22	3.00
LPLH-R-A	91.67	-51.68	-3.90
KODT-R-A	90.28	2.89	16.92
ISTT-R-A	88.89	-20.84	-6.58
HUPZ-R-A	80.50	16.83	11.88
CROS-P-A	86.11	66.24	3.17
VART-R-1	80.56	-32.08	20.62
SLRS-R-A	79.17	78.53	17.37

Figure 3-10: Prediction market in finance [110]

From Figure 3-10, it can be seen that there are 3 times out of 10 then the rankings are totally wrong when then unprofitable stocks were selected. There are only about 4 or 5 times that the model seems pick the right stocks. The authors analysed that the problem occurred when the inexperience players would copy the high ranked players and make prediction in the same way.

3.3.2 Discussion

The prediction market works well in many areas. Although it is still rarely applied to the problem of stock price prediction, its performance shows that it can be suitable for our research, since we want to create a dynamic stock prediction based machine learning model. In our system, there will be many different types of machine learning predictors. To make a good prediction, we will apply the prediction market method in order to create the momentum of the predictors. The predictors that work well will have more weight than the weak predictors which are the predictors that did not provide correct answers recently. This weight will be used for evaluating the level of confidence of each predictors before making generating the trading signals.

3.4 Machine Learning in a Trading System

The term *trading system* usually covers the following aspects of investment: portfolio, orders and data [10]. In this research however, we are omitting portfolio construction since our focus is on developing a mechanism to effectively manage machine learning predictors. Figure 3-11 illustrates how a basic trading system is organised, excluding the portfolio.

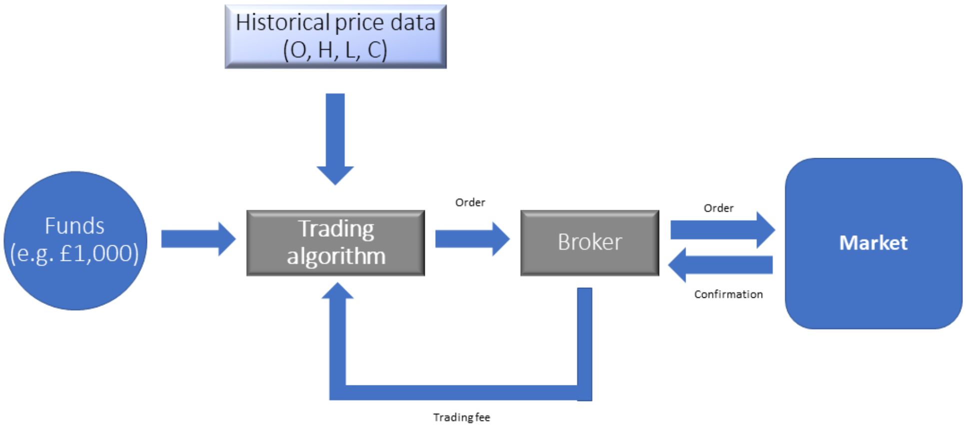


Figure 3-11: Basic Framework of Machine Learning in a Trading System [10]

Figure 3-11 shows that the trading algorithm is the central component which connects other parts of the system, and altogether the structure takes historical price data and eventually turns this into an order signal which is entered into the market through a broker. If the algorithm predicts a positive return for the stock, then an order to go long (buy the stock) will be passed onto the broker who facilitates the purchase of the shares from the market, charging a fee for doing so. If a negative return is expected, then the order will state to sell any shares the investor holds to the market, again incurring a fee from the broker. Brokerage fees will be included later in our system evaluation.

The last process of this framework is when the broker purchases or sells shares from or to the market. During this process, a typical trading system does not necessarily send the order all at once: very large orders may have to be separated. However, given our decision to focus on retail traders, we do not expect this to be necessary for our trading system as the funds retail traders have to trade with are significantly smaller.

3.5 Discussion

This chapter reviewed multiple areas in machine learning which could be useful for developing our trading system. The chapter started with an overview of well-known machine learning algorithms and applications of machine learning to the stock market. What follows are things we considered when selecting machine learning algorithms for our research. Firstly, the type of algorithm. In order to build our trading system, we focused on regression algorithms as we would like to predict the future returns of stocks. However, there are many regression algorithms to choose from, and one consideration when we narrowed them down was their relative complexity. While there are a lot of computer scientists continuously trying to come up with new complex algorithms, such as deep learning neural networks, we believe that this is not the only way to improve the quality of the predictors, and they would still suffer from needing to be applied at the right time. We believe existing machine learning could be enough if we are able to consistently select predictors at the right times. Additionally, in a complex area subject to regime shifts such as the stock market, we consider a redundancy of machine learning algorithms to be an advantage in that our model should be robust to abrupt changes in the characteristics of the data and able to adapt and consistently profit. We discussed advantages and disadvantages of the approaches we have included, which demonstrates how they can make up for each other's weaknesses. For example, if polynomial regression gives poor results due to overfitting, its predictions will not be used for the final signal to buy or sell and another approach less overfit and making good predictions will be chosen instead. For any given historical period there may be an individual algorithm that could beat our model, but this does not invalidate the principle of the ensemble model. Finding the very best algorithm out of all machine learning for a time in the future is nearly impossible and with a single algorithm the strategy is fragile if the nature of the market changes. Moreover, we hope to show that it is unnecessary, since making a handful of existing algorithms work together effectively as an ensemble should consistently profit above the benchmark strategy. These may even be ordinary, less complex algorithms, none of which are the optimum out of the universe of machine learning. This is the idea from which we started our work.

Our proposed model is able to select the best algorithms and predictors for a given stock at a given time. Therefore, it is able to make many different predictors work together effectively, resulting in higher performance than a benchmark strategy or any individual model within the system. Finally, the last concern for our machine learning choice is the time processing. Algorithms with short processing times are desirable because one algorithm will be used to create multiple predictors according to the number of clusters. Outside of this research, the processing time decision depends on the needs of users. However, as discussed previously, this research would like to show that even basic machine learning can work well if used correctly. For these reasons, we excluded time-consuming algorithms, such as neural networks and deep learning. However, the choice of machine learning to include need not be limited as our model is designed to be able to add or remove predictors easily. We restricted ourselves as our aim was to prove that our model can effectively put different predictors to work together.

The machine learning algorithms we selected are common algorithms which most engineers and scientists will know [4] [5] We started with the most basic ones, polynomial and linear regressions. Once we found out that this initial selection of algorithms could not deal with some characteristics of the data, we started to add more algorithms. SVM was the next choice as it a little more complicated and works well with outliers. For SVM, we used three different kernels, which are rbf, linear and polynomial. Since changing kernels leads to different results, these SVMs are considered to each be different predictors. Finally, we added one more popular algorithm called random forest. Even though it takes a little longer to run than the other algorithms, random forest is added as it is supposed to handle imbalance and missing data well, and these issues affect some stocks. As mentioned earlier, the model is not limited to these choices. This is just the set of algorithms that we started with in order to prove that our system works. These choices can be changed if needed in the future.

In addition to machine learning, this section reviewed multiple possible methods for the proposed ensemble model. Then, we discussed predictive markets, which build on a similar idea called the wisdom of the crowd. Finally, this section reviewed common trading systems with machine learning frameworks in order to form a better understanding of where and how machine learning predictors could be incorporated in the system.

[4] `https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article` retrieved 2019-09-15

[5] `https://medium.com/analytics-steps/top-10-machine-learning-algorithms-77704f259638` retrieved 2019-09-15

Chapter 4

Ensemble Machine Learning for Individual Stock Investment System

This chapter aims to provide the overview of our ensemble machine learning system in order to provide the board understanding. Then in the next chapter, the experimental details during the process of designing this system will be explained. The system comprises six main subsystems as shown in Figure 4-1 on page 89.

The first step is data preparation, where the dataset will be loaded, cleaned and divided. After that, all the necessary features will be created in order to be used by the following step. The second step is the clustering process. In this step, the training set will be clustered into different classes. The third step is the process where all the predictors will be created and trained with a specific cluster of training data. After this the training, predictors are ready to work together in the next step which is called system development. This step is the most important process in this research as the system have to design how all the predictors will work together with the highest efficiency. Therefore, we create a scoring system in order to measure the accuracy of each predictor. These scores will be used as the criteria for selecting the appropriate predictors at the right time.

The final two steps are signal creation & evaluation and testing. The signal creation & evaluation step takes the predicted result from the predictors selected in step 4 and produces the trading signals. The signals will then be evaluated and optimised. After

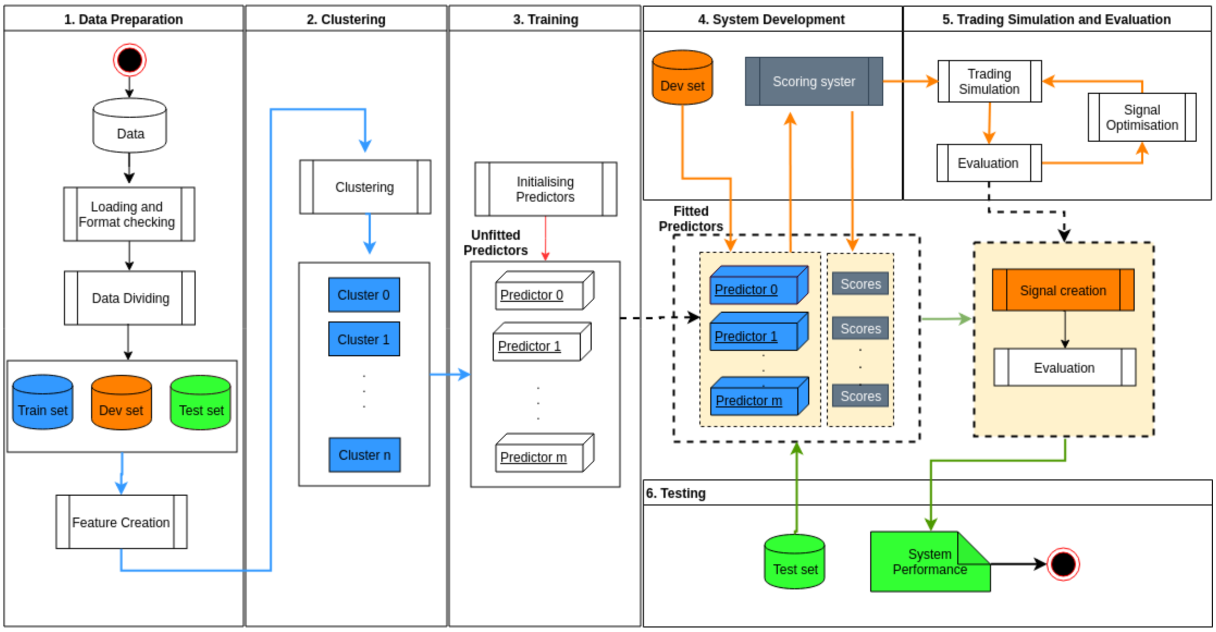


Figure 4-1: System Diagram

optimising the signal, we will perform the last step, testing the model with unseen data and measuring the performance of the whole system.

4.1 Step 1: Data Preparation

The first step, data preparation, composes of three tasks which are loading & checking dataset, separating dataset and creating features. The diagram of this step shows in Figure 4-2.

4.1.1 Data Loading and Format Checking

The first task is loading a stock dataset into the system, following which we will check the validity of the data format to make sure that we have got all the necessary information. The correct format of the dataset comprises the date, Open - High - Low - Close prices and Volume, as shown in Figure 4-3.

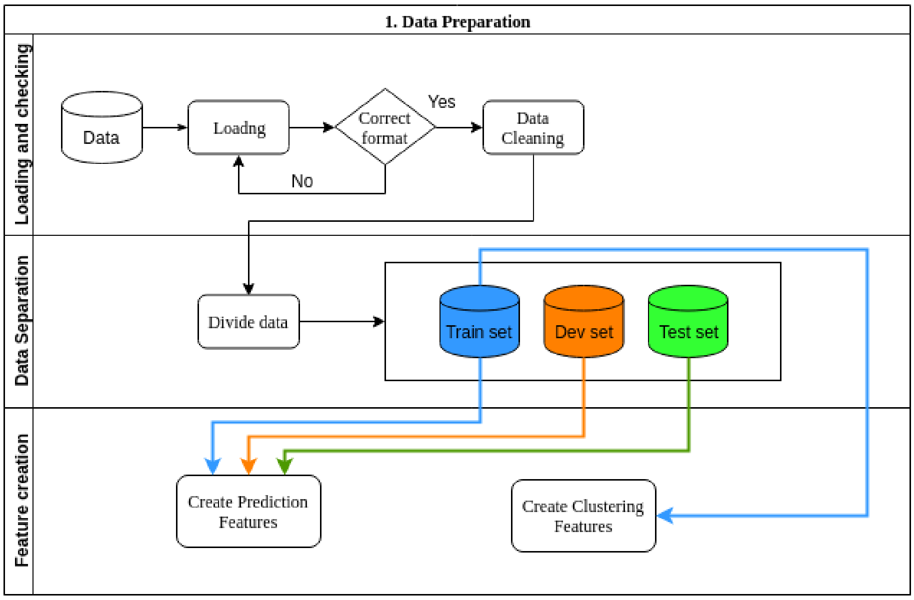


Figure 4-2: Data Preparation Diagram

A	B	C	D	E	F
Date	**Open**	**High**	**Low**	**Close**	**Volume**
02/01/2003	50.65	51.61	50.52	51.6	7545500
03/01/2003	51.61	51.61	49.85	50	8389300
06/01/2003	50.2	50.55	49.67	50.19	7438400
07/01/2003	50.32	50.76	50.1	50.46	6669000
08/01/2003	50.4	51.36	49.86	49.99	7796900

Figure 4-3: Correct format of dataset

4.1.2 Data Separation

In this step, the dataset is divided into three sets: training, development (or validation) and test sets. These are shown in Figure 4-4 on page 91. The training set is used for creating the predictors. The development or validation set will be used to create the ensemble trading system and to optimise the trading parameters. Finally, the test set will be used for the purpose of system evaluation.

Figures 4-4 shows training, validating and testing set in blue, orange and green colour graphs, respectively. Training set starts from the beginning of 2000 until the end of 2014. Then, the next two years (2015-2016) is used as the validation set. Finally, the last two years (2017-2018) is the test set.

Figure 4-4: Train, Development and Test sets

4.1.3 Feature Creation

In this step, additional features will be created for two reason, firstly, to be used to cluster our training data and secondly, to be used by machine learning predictors to predict the future returns of a stock. Therefore, there are two type of features in this step.

- **Clustering Features**: These features are to be used in order to cluster the training data into smaller groups. We decide to use two features which related to the trend & the volatility of stock's price. These clustering features are the mean and standard deviation of returns.

- **Prediction Features**: The prediction features aim to be used by machine learning predictors to predictor the future return. As the price is normally not stationary, in this research, we will predict the future returns instead of the prices. As for the return, we will use the return in logarithm scale to show the percentage change or multiplicative factors and to avoid the problem of skewness towards large values when a few trading days having much larger returns than the other days. The equations for the logarithmic calculation can be seen in Chapter 2 section 2.3.4 on page 42. The prediction features created in this step are the logarithm returns of the previous week. For example, in order to predict the tomorrow return, a series of five day returns in the past will be taken into the prediction. In this research, the look-back period is a control parameter setting to 5 days but this can be changed. However, this research will not focus on this issue.

4.2 Step 2: Clustering

In this step, we apply clustering features from section 4.1.3 to cluster the training set into smaller groups. The framework of this step is shown in Figure 4-5. The framework shows that the clustering features from the training set (items in dashed lines are the values obtained from the previous step) are fed into the K-means clustering module in order to cluster the training data into small groups.

The purpose of this step is to cluster the training set into groups based on their characteristics. Since we want to create an ensemble model which incorporates many machine learning predictors, training each predictor to be the best at predicting a specific pattern is the key idea. Therefore, we cluster the training set to clusters and train each predictor only on a specific cluster in order to make them act like an *expert* on that pattern. After training, a predictor should provide accurate predictions when facing the pattern they have been trained for. On the other hand, they might not be very good at other patterns. We will then put all of the predictors to work together with our control system which aims to select suitable predictors for the present. Therefore, our prediction system is able to adapt itself to cope with different patterns of input data over different periods of time.

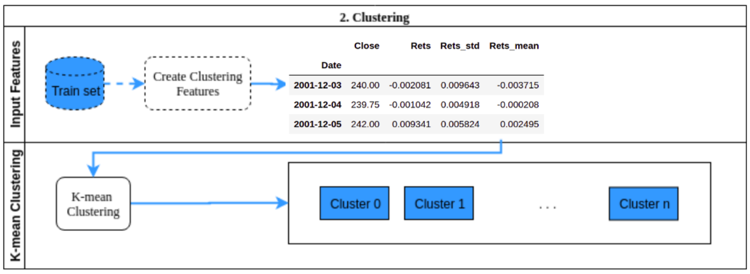


Figure 4-5: Clustering process diagram

We have conducted a numbers experiments to confirm that having multiple groups of training sets increase the performance of the system. The detail of these experiments can be seen in chapter 5, section 5.1.1, on page 118. The clustering technique used in this step is k-means cluster as it is one of the well-known and effective machine learning algorithm. Beside that, dataset we would like to cluster has only two dimensions (mean and standard deviation) therefore, k-means clustering would work effectively.

4.3 Step 3: Training

In this step, we initialised and trained machine learning predictors with our clustered training data. Figure 4-6 shows the framework of this step. There are m different machine learning algorithms to be used as predictors and n groups of training data. Therefore, the number of predictors in total is $m * n$, since each machine learning algorithm will be trained for each clusters.

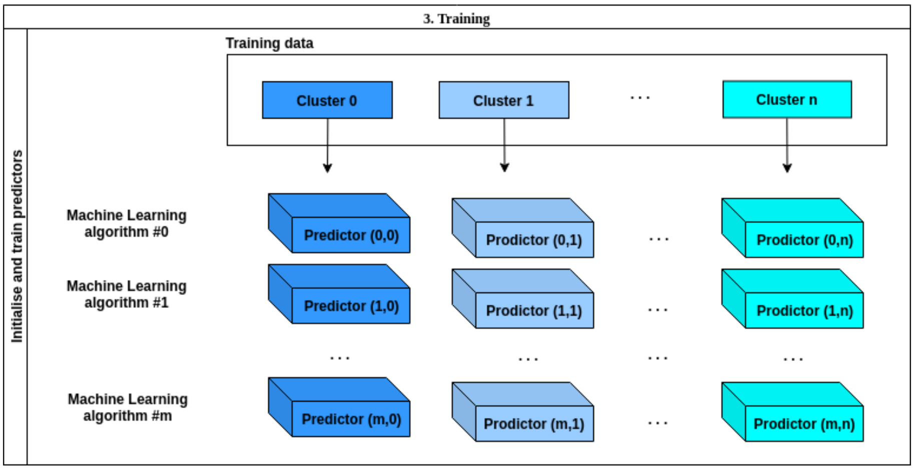


Figure 4-6: Training process diagram

From Figure 4-6, it can be seen that the training data in each cluster is used to train only predictors that were created for that specific cluster. As for the choice of machine learning model, our model is designed with the flexibility to add, remove or change the type of predictor. To begin with, we will work with the four most well-known and widely-used machine learning algorithms, which are linear & Logistic regression, support vector machine and random forest. The same algorithms with different kernels will be accounted for as different models as they provide different results. The list of our starting predictors is as follows:

- Linear Regression
- Polynomial regression
- Support Vector Regression (rbf)
- Support Vector Regression (linear)
- Support Vector Regression (Polynomial)

- Random Forest

4.4 Step 4: Scoring System

After fitting the predictors from the previous section, in this step we will bring all the trained predictors to work together as an ensemble predictive model. In order to build this ensemble model, we created a control system, which we refer to as the *scoring system*. The scoring system manages the results of individual predictors, then makes a decision on which predictor or set of predictors will be selected for the next prediction. With this scoring system, the different predictors will be activated at the different times based on their scores, which indicate their performance. The scoring system is composed of the two layers, the predictor layer and ensemble model layer, which means that there will be two main scoring computations in this model. The framework of this step is shown in Figure 4-7. on page 94.

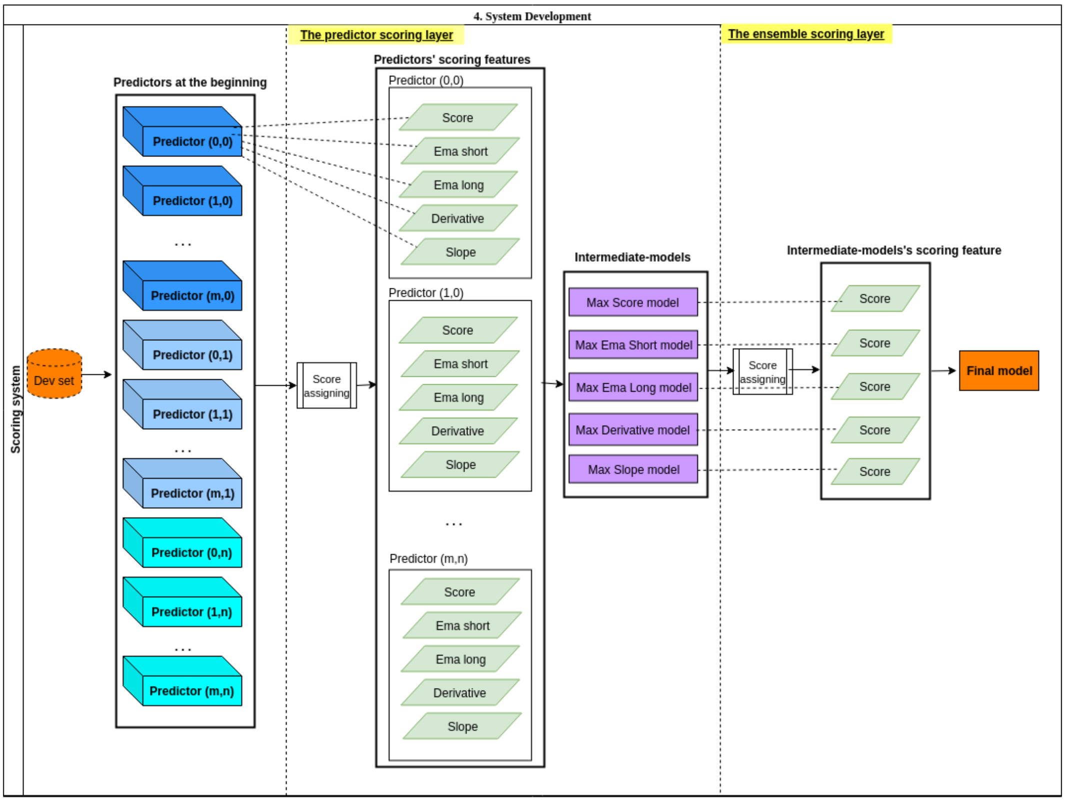


Figure 4-7: System Development diagram

The framework in Figure 4-7 shows that the system development step starts from loading the dataset (dev set) into the system, then every predictor making its own prediction individually. Then, predictors will be rewarded or punished according to the prediction they have made. Rewards and punishments are turned into scores which will be sent to the two following scoring layers. These two scoring layers comprise the intermediate models and the final model.

Firstly, in order to create the intermediate models, the scores obtained from the predictions will be used to calculate five scoring features for each predictor, and the predictors with the highest values in each feature will be selected in the intermediate models. Secondly, the intermediate models will provide their own predictions. These intermediate models' predictions will be assessed and a score again calculated to create the intermediate-models' scoring features. Finally, the intermediate models with highest score will be selected as the final model. The predictors that are in the final model will be nominated to predict for the next round. More details of this scoring system are explained below.

To make many predictors work together effectively, a two-layer scoring system has been invented. The first layer is *predictor scoring layer* and the second layer is the *ensemble scoring layer*. The predictor scoring layer provides intermediate models which will be used in the next layer, *ensemble scoring layer* in order to create the final model or to select the nominated predictor(s) for the next prediction.

4.4.1 The predictor scoring layer

The predictor scoring layer is the first stage in the scoring system. The purpose of this stage is to obtain the intermediate models. Figure 4-8 on page 96 shows the framework of this step. All predictors start predicting from day t (after the look-back period), and every predictor obtains new information of day t and uses that information together with the previous information they have, then gives the prediction for the next day $(t + 1)$. In this step, each predictor will give one result, which is the predicted future return for the next day. The results from the predictions are shown in the grey circles. Predicted values are the real numbers which will be changed to -1,0 or 1. A result of -1 means that the predictor has made the decision that the future price will be lower the next day. On the other hand, a result of 1 means the future price is expected to increase. If a predictor predicts no change is to be expected, the predicted value will be 0.

After the predicted return on day $t+1$ had been made, at the end of the day $t+1$, the

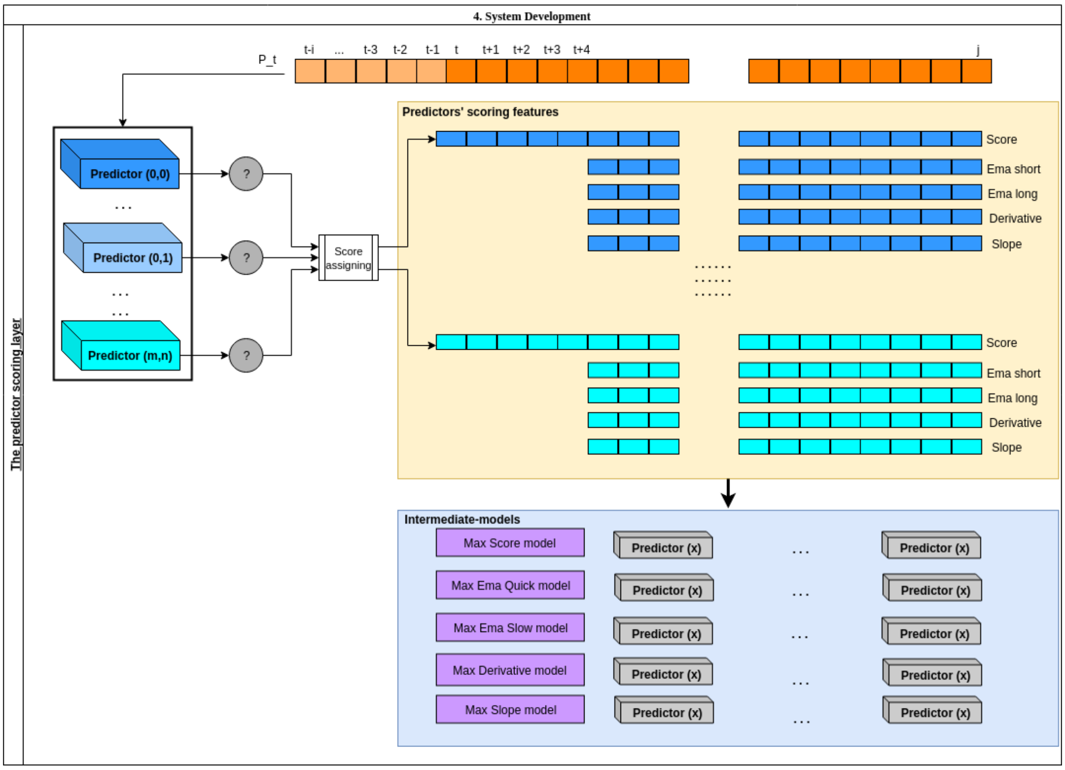


Figure 4-8: The predictor scoring layer

scoring assigning will start working. It will compare the actual return and predicted return then predictors will be rewarded or punished based on their accuracy. Rewards and punishments are the scores (points) that each predictor will be given. A positive score will be given when the prediction is accurate. On the other hand, a negative score will be assigned to a predictor as a punishment if it has given the wrong answer. The values of the rewards and punishments vary based on how well the predictor did. A better prediction provides the correct answer and leads to more profit, which means that predictors which are accurate on high-return days are very important. On the other hand, if the predictions lead to massive loss, the predictors that gave the answer will lose more points.

There are 4 different scores (points) that will be given to our predictors. The best predictors (those giving answers leading to high profit) are given a full score of 1. The accurate predictors (which give the correct answer but on the days that have a low return) will be given 0.5. The predictors that work very poorly (their answers lead to a big loss) will be punished with -1, and the inaccurate predictions that do not effect much of the overall system (their answers are incorrect but not on very important days) will be punished with only -0.5. To set the thresholds of this scoring system, we calculated the different quartiles from the distribution of the return from our training set.

Here is an example of how we get these thresholds. Figure 4-9 illustrates the return distribution of Marks and Spencer Group Plc shares, or MKS. The figure comprises two sub graphs. The top graph represents the logarithm returns for each trading day. The logarithm returns vary between -0.3 and 0.18. The are a few peak points on the positive side of the graph which indicate the extraordinary high return days of that stock in the training set. On the other hand, the spiky points on the negative side indicate the days on which the price dropped enormously. The bottom sub graph shows the distribution of logarithm return. There are three vertical lines on the graph which represent quartile 1 (Q1), quartile 2 (Q2) and quartile 3 (Q3), respectively.

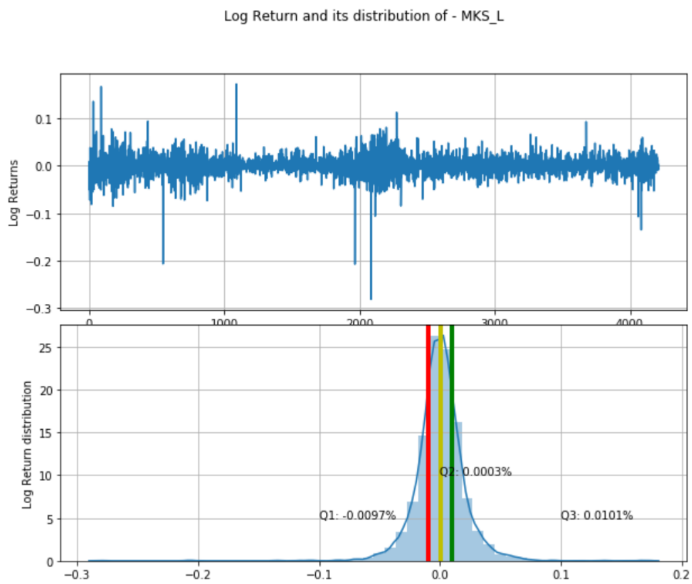


Figure 4-9: Return Distribution

From Figure 4-9 on page 97, the best predictors are the predictors which provide answers resulting in a return higher than Q3 (0.0101), and the worst predictors are the ones that gave answers leading to a loss more than Q1 (-0.0097). These 2 types of predictors will be assigned scores of 1 and -1, respectively. As for the accurate and inaccurate predictors that lead to returns between Q1 and Q3, these will be given scores of 0.5 and -0.5, respectively.

After each predictor is assigned a new score, their accumulated scores are updated and will be used for creating the additional 4 predictor scoring features, which are the Exponential Moving Average (calculated over both shorter periods and longer so as to respond more quickly or slowly to information, respectively), derivative and slope. These features are stored in the memory part of each predictor. Every predictor has a set of these features as shown in the yellow rectangle in the framework (Figure 4-8 on page 96).

At the beginning of the process, only the accumulated score is updated until day t. Another 4 additional features will be created from day $t + 1$. A number, i, is the pre-set value indicating the look-back period (how many days) that we want our predictor to take information in order to make prediction. The details of predictors' scoring features are as follows:

- **Score**: storing accumulated score for each predictor
- **Exponential Moving Average (Quick)**: storing values of exponential moving average of the accumulated score with beta = 0.9. This feature will keep track of the changes in the score over a short period (about 10 days). Therefore, this feature will monitor the rapid changes in each predictor's performance.
- **Exponential Moving Average(Slow)**: storing values of exponential moving average of the accumulated score with beta = 0.98. This feature will keep track of the changes in the score for a longer period (about 50 days). Therefore, it is used to identify the gradual changes in predictors' performances.
- **Derivative**: storing the derivative of the accumulated score for the period of i days. This feature is only interested in the changes that happen between the beginning and the end of a period, not the other days in between.
- **Slope**: storing the slope of the accumulated score within i days period. This feature is similar to the derivative in that it is trying to monitor the changes from the beginning to the end of the period. However, the difference is it will consider the values of other days in between this period as well.

The predictors' scoring features as listed above are created for different purposes in order to capture the changes of a predictor's performance. After each prediction, these features will be calculated. Then, we will create intermediate models which can be seen in the blue rectangle in Figure 4-8. There are five different intermediate models according to predictors' scoring features as illustrated in the purple rectangles. In each intermediate models, the predictors with the maximum values for each feature will be selected to be members of each model. For example, the predictors with maximum accumulated score will be added as members of the max score model. While, the predictors with maximum derivative will be added into the max derivative model, etc. Each intermediate model can have one or multiple predictors as members.

In the case of having multiple predictors in an intermediate model, the results from the member predictors will be averaged to provide the result. With this strategy, the performances of each predictor will be identified with different perspectives as follows:

- The predictors which perform well in general will become members of the max score model.
- The predictors which are not performing very well presently but have done well recently will be added into the max ema quick model
- The predictors which are not performing very well presently but have done well for a long time in the past will be added into the max ema slow model.
- The predictors with suddenly improving performance will be assigned to the max derivative model
- The predictors with continuous change of performance will be counted as members of the max slope model

Once the intermediate models are created, we will be ready to move to the next layer of the scoring system, the ensemble scoring layer.

4.4.2 The ensemble scoring layer

The ensemble scoring layer is the final process of the scoring system. The framework of this process can be seen in Figure 4-10 on page 100. After all the intermediate models have been created from the previous process, as seen in the dashed-rectangle, each intermediate model will predict a result for day t again but this time the result from each model will come from every member within that intermediate model. There are two types of results from an intermediate model: return and signal. Return is averaged from all of the members' results in that intermediate model, while signals are obtained from the majority voting of all members.

Every intermediate models' result will be rewarded or punished by the score-assigning module in the same way as the predictors' scoring system. Each of the models will be rewarded or punished with a score of 1, 0.5, -1 or -0.5. To decide the score each intermediate model, we will again take the returns' distribution into account as seen in Figure 4-9 on page 97. According to the distribution, the best intermediate models which gave answers leading to profit more than Q3 will be awarded 1. Correct intermediate models which made profit but less than Q3 will be given only 0.5. The worst intermediate models which provide incorrect answers on the day that lead to a loss more than Q1 will be punished with -1. The intermediate models which also give incorrect answers but lost only a small amount, less than Q1, will be punished less with -0.5.

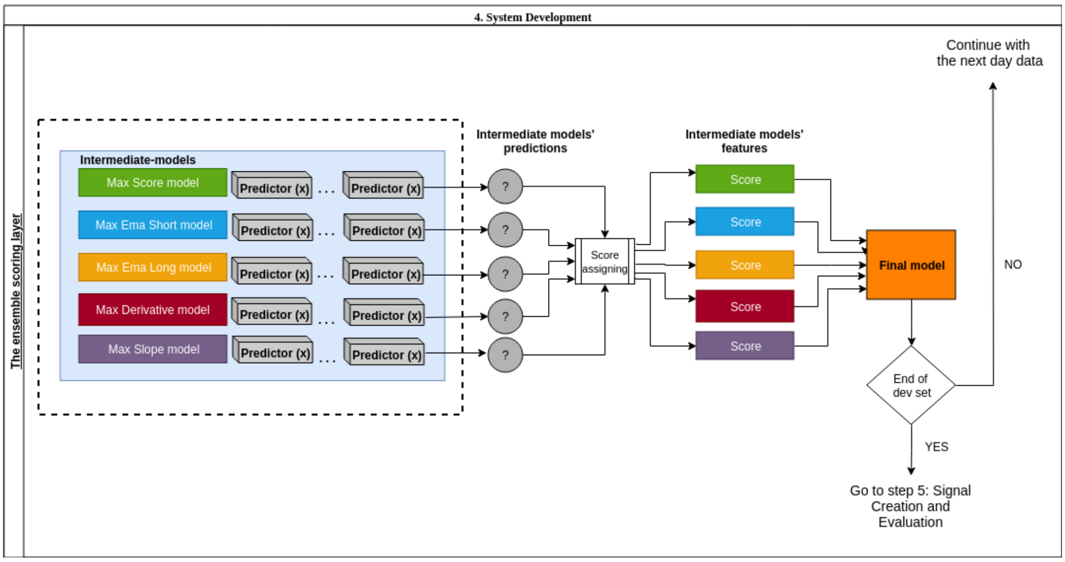


Figure 4-10: The ensemble scoring layer

After having been assigned a new score, each will update their accumulated score individually. These accumulated scores become the features of intermediate models which are to be used in order to select the intermediate model that will be a member of the final model. In the final model, the predictors which have been selected to be members will act as representatives of all the predictors to provide the prediction for the next day (day $t + 1$). And the scoring process will continue from the beginning with the next day's data until the end of the development period.

4.4.3 Example of how the predictors' scoring layer works

In this example, we use an artificial stock as shown in Figure 4-11 on page 101. The predictors' scoring layer starts working from the beginning of this data. The dash rectangle shows the area where the scores will be illustrated. The predefined look-back period is a week or 5 days ($i = 5$).

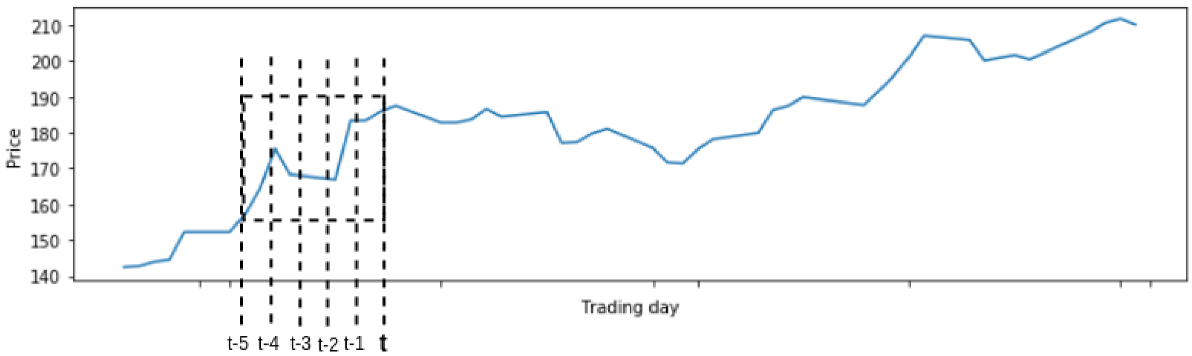


Figure 4-11: Focus period of an example

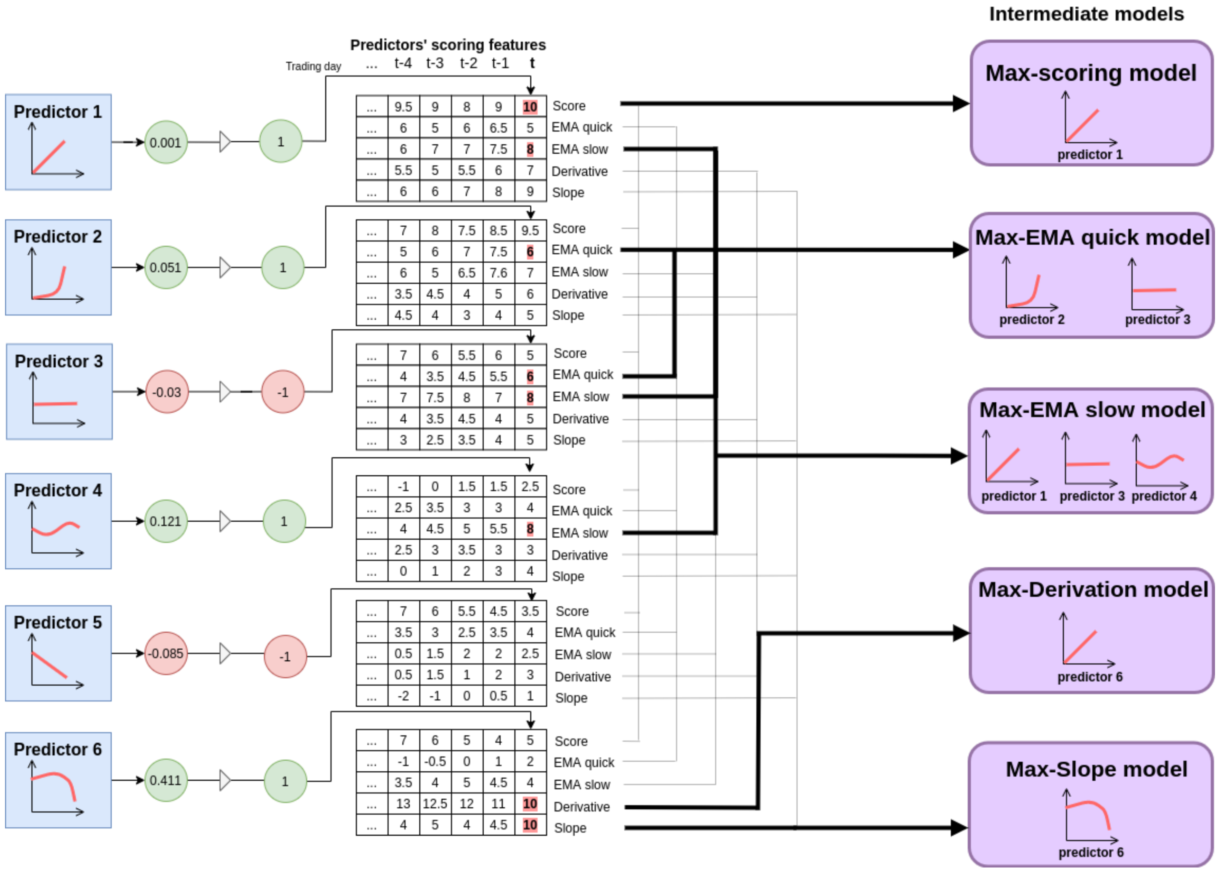


Figure 4-12: Example of how scoring system works on the predictor scoring layer

Figure 4-12 on page 101 shows the mechanism inside the *predictors' scoring layer*. At the end of day $t-1$, the closing price of day $t-1$ and those of the previous i days are sent to every predictor (blue square). Each predictor uses these inputs to predict the return of the next day (day t). Then, the predicted returns will be transformed into signals which can be either 1, -1 or 0. If the signal is 1, it means that the predictor predicts an increase in the value of that stock. On the other hand, if the signal is -1, it means that a decrease in value is expected. Then every signal from the predictor will be evaluated at the end of the next day (day t) when the actual price is reported.

The predicted direction will then be compared with the actual direction of the stock price. A predictor that provided the correct answer will be awarded either 1 or 0.5 points based on how much profit that predictor made on that day. On the other hand, a predictor which gave an incorrect answer is punished with either -1 or -0.5 depending on how much of a loss that predictor caused. The criteria for score assigning can be seen in section 4.4.1 on page 95. After getting a new score, the accumulated score on day t of each predictor will be updated, followed by other features, which are EMA quick, EMA slow, derivative and slope.

For example, on day $t-1$, predictor 1 predicts that the price will go up (the signal is 1). Then, the actual price increases the next day (day t). Therefore, in the predictor's scoring features section, the accumulated score (shows as score) of predictor 1 increases from 9 to 10. Predictors 2,4 and 6 also increase their accumulated score by 1 as they have all given the correct prediction. On the other hand, predictors 3 and 5 which gave wrong predictions get -1, and their accumulated scores decrease.

After all of the predictor's scoring features are calculated for every predictor, five intermediate models are created by selecting the predictors with the maximum value from each feature to be their members. The maximum values of each feature at day t are highlighted in red in the tables. For example, in order to create a max-scoring model, the predictor(s) with the highest score (highest score is 10), which is predictor 1, will be selected. As there is only one predictor with the maximum score, the max-score model has only one member as shown in the purple rectangle on the top right of the framework. On the same basis, the max EMA quick model is created by selecting the predictor(s) with the highest ema quick value (highest ema quick is 6), which are predictors 2 and 3. As predictors 2 and 3 both have the same ema quick value and it is the maximum, the max-EMA quick model has 2 members. Other intermediate models are also crated in the same way. At the end of this process, each intermediate model should have at least one predictor as a member, except when the maximum feature values are negative. In this case, it will not be possible to select the predictor as no predictor is doing well enough. Therefore, that specific intermediate model does not have any members on that round and will provide the signal 0 for the next day.

Figure 4-13 on page 103 shows the mechanism of the second part of the scoring system called *ensemble scoring layer*. In this example, the max-scoring model, max-derivative model and max-slope model have only one predictor, while the max-ema quick model and max-ema slow model have two and three members respectively.

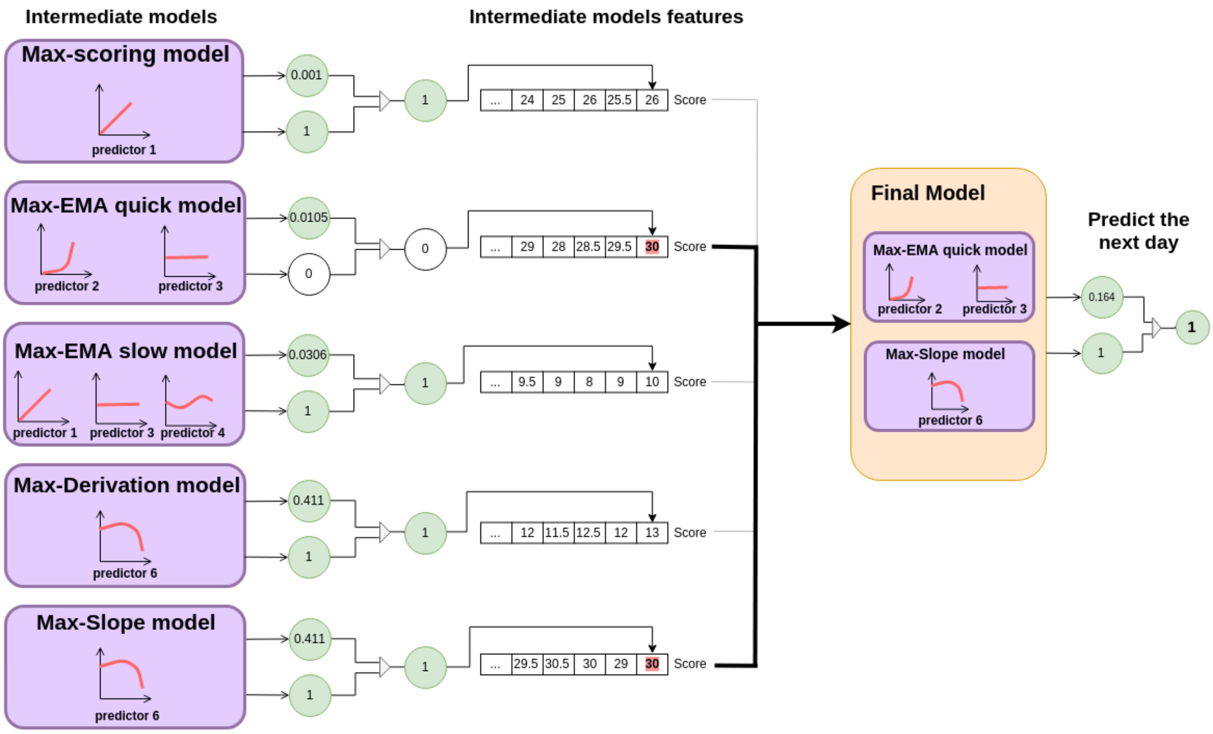


Figure 4-13: Example of how scoring system works on the ensemble scoring layer

In Figure 4-13, the intermediate models are represented as purple rectangles with the predictor function inside. Some of them have only one member, which means that the result will come from that specific predictor. Therefore, the result will stay the same as in the previous layer. For example, the max-scoring model has only one predictor which is predictor 1. Therefore, the result of this intermediate model is the result of predictor 1, 0.001 and 1 for the predicted value and signal, respectively. These results are the same as the result of predictor 1 in the previous step as shown in Figure 4-12 on page 101. This works the same for the max-derivative and max-slope models as all have only one member.

For the intermediate models with multiple members, their results come from the results of every member inside. For example, the max-EMA quick model has two members which are predictor 2 and predictor 3. As predictors 2 and 3 provided predicted returns of 0.051 and -0.03 (as can be seen in Figure 4-12), the return of the max-EMA quick model is the average value of 0.051 and -0.03, which is 0.0105. The predicted signal of this model comes from the majority vote of predictors 2 and 3, which are 1 and -1 as shown in Figure 4-12. Therefore, the majority vote result is 0.

After getting the return and signal from the member, these two results will be compared with each other in order to create the intermediate model's answer. If the return and signal are both a positive number greater than 0, the intermediate model's result will be 1. On the other hand, if the return and signal are negative, the intermediate model's signal is -1. Otherwise, that intermediate model's signal will be set to 0. For example, in the max-ema quick model, the predicted return is 0.0105, while the predicted signal is 0. Therefore, the answer of this model will be 0.

As for the max-ema slow model, it has three members, which are predictors 1,3 and 4. Therefore, the predicted return of this model comes from the average of results from every member. As predictors 1, 3 and 4 provide predicted values as 0.001, -0.03 and 0.121 respectively, the average predicted value is 0.0306. As for the predicted signals, predictors 1, 3 and 4 predicted signals of 1, -1 and 1, and the average signal is calculated from the majority vote of these results. Therefore, the signal of the max-ema slow model is 1.

As explained above, each intermediate model sends out two values, return and signal, as can be seen in the two circles sent out by two arrows from every purple rectangle in Figure 4-13. To obtain a final signal of each model, these two values will be compared with each other. If they are both positive values greater than 1, the resulting signal is 1. If these two values are negative, the resulting signal will be -1. Otherwise, the resulting signal will be 0.

For example, the predicted return and signal of max-ema slow are 0.0306 and 1, which are both positive numbers. Therefore, the answer of this intermediate model is 1. This means that the max-ema slow model predicts that the price will increase and suggests buying this share at the beginning of the next day. On the other hand, the return and signal of the max-ema quick model are 0.0105 and 0. The value 0.0105 is positive which means to buy and 0 means to hold or do nothing. Therefore, the resulting signal from the max-ema quick model will be 0, which means that this model suggests holding the position or not doing anything, neither buying nor selling, because there is no clear signal to buy or sell.

After getting a signal from every intermediate model, the intermediate models' features will be updated. The intermediate model feature here is the score. As same in the first layer scoring, we always start with score merely. If the score does not work, we then investigate further on the additional features. It turns out as the score in this ensemble scoring layer works really well as can be seen later in chapter 5, section 5.3.2.4, on page 181. We then conclude that the only score is enough for this second layer scoring system. Therefore, the intermediate predictors with the maximum cumulative score will be selected in the final model and are able to predict in the next round. This process will continue throughout the development set (dev set). When finishing, we will move onto the process of simulation and signal optimisation in order evaluate and improve the quality of trading.

4.5 Step 5: Trading Simulation and Optimisation

Results for step 4 in section 4.4 are obtained in this section through simulating a realistic trading environment which accounts for brokerage fees and initial funds. There are four sub modules in this step, which are signal creation, trading simulation, evaluation and signal optimisation, as shown in Figure 4-14.

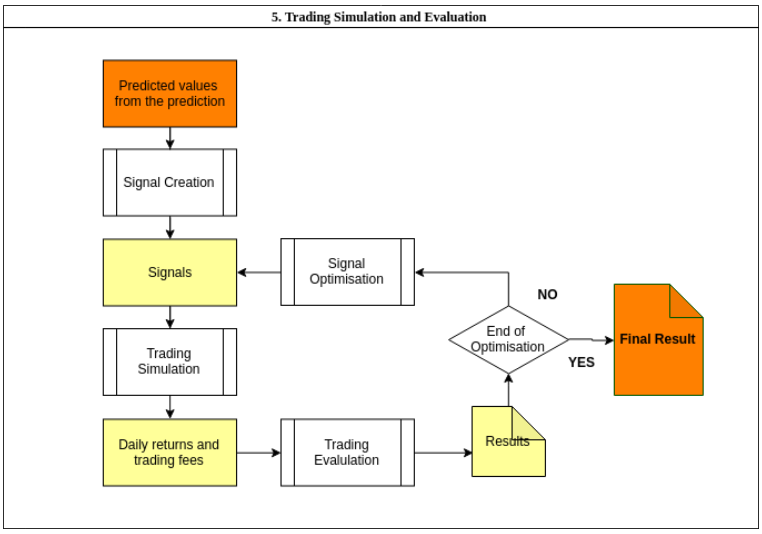


Figure 4-14: Trading simulation and optimisation diagram

4.5.1 Signal Creation

This step feeds in the values obtained from the scoring system in step 4 section 4.4, which are the predicted return and the predicted direction (either -1, a decrease in price, 1, an increase, or 0, no clear prediction). The predicted return and direction can be different as they are calculated in different ways. The predicted return comes from averaging the return forecast by each individual predictor in the final model, while the direction is obtained by a majority vote on the direction forecast by each of the predictors. Mostly, if the predicted return is positive, the predicted direction is also positive, and the same applies when the predicted return is negative. However, there are a few cases where the predicted return and direction are opposites. Therefore, in order to make decision which calculation is the best for creating the final signal, we performed a number of experiments which the detail can be found in chapter 5, section 5.5.2, on page 212.

From those experiments, we conclude that the combined return & signal is the best option for creating our final signal. Here we will illustrate how this method works by providing an example. An example of the final signals created by the combined direction & return method can be seen in Figure 4-15 on page 106.

Predicted Results			Signal Creation		
Date	**Predicted return**	**Predicted signal**	**Predicted return**	**Predicted signal**	**Combined return & signal**
2015-02-24	0.002254	1	1	1	1
2015-02-24	-0.000478	-1	-1	-1	-1
2015-02-24	0.01547	1	1	1	1
2015-02-24	0.000047	0	1	0	0
2015-02-24	-0.00087	0	-1	0	0
2015-02-24	-0.0000114	1	-1	1	0
2015-02-24	-0.00000178	0	-1	0	0
2015-02-24	0.00526	-1	1	-1	0
2015-02-24	0	0	0	0	0

Figure 4-15: Signal creation method

From Figure 4-15, it can be seen that in order to create the final signal in the last column (grey), we will combine the predicted return and direction. If they are both positive, the final signal will be 1, which means "buy". On the other hand, if the return and signal are both negative, the final signal is -1, which instructs investors to "sell". Most of the time, the predicted return and direction are either both positive or both negative. However, there are some cases where these values are in counter directions. In such cases, the final signal will be 0, which implies "to hold" or not do anything as the signals are not strong enough to recommend any actions. However, as described earlier, the interpretation of a "hold" signal depends largely on the individual investor and their appetite for risk and does not necessarily mean they must hold on to shares they currently own or stay in cash if the previous signal instructed them to sell.

4.5.2 Trading Simulation

This section takes the final signals from the previous step, 4.5.1, and applies them by simulating a realistic trading environment, by which we mean brokerage fees and initial funds are taken into account.

This research aims to be used by individual investors followed our objective to support the investor to trade with the limited funds. We will only use a flat-rate fee as opposed to a percentage of the order size, as this is a more common fee structure for private and retail investors. Brokerage fees are subject to the broker companies. In this research, we take the fee information from the 16 popular brokers including Fidelity, ChoiceTrade, SpeedTrader, TradeStation and Interactive Brokers (`https://www.stockbrokers.com/guides/features-fees` retrieved 2019-09-15). The trading fees of these companies range from £0.00 - £5.61, and we have taken an average of these values. The average fee of the 16 companies is £3.79. Therefore, for every trade, £3.79 is deducted to represent the brokerage fee throughout the simulation. In reality, inflation might lead to fees being increased over time, however as our testing dataset is quite recent, we have not simulated this. Additionally, as the average of fees was calculated recently, the effect of any inflation would mean that the brokerage fees are larger in our simulation than they might have been in reality.

Figure 4-16 shows an example of the result from the trading simulation process. There are four columns in this example, comprising the date, close price, signal, profit and trading fee. Please note that the close prices are shown in normalised form. In this case, we start with initial funds of £10,000. The first signal that this model sends out is 1, which means "buy". This signal comes in at the end of 2000-02-08. So, at the

Index	Close	Signal	Profit	Fee
2000-02-01 ...	0.00881	0	10000	0
2000-02-02 ...	0.008723	0	10000	0
2000-02-03 ...	0.008679	0	10000	0
2000-02-04 ...	0.008635	0	10000	0
2000-02-07 ...	0.008723	0	10000	0
2000-02-08 ...	0.008766	0	10000	0
2000-02-09 ...	0.009029	1	10296.1	-3.79
2000-02-10 ...	0.009292	1	10596	0
2000-02-11 ...	0.009336	1	10646.2	0
2000-02-14 ...	0.009424	1	10746.6	0
2000-02-15 ...	0.009336	1	10646.2	0
2000-02-16 ...	0.009424	1	10746.6	0
2000-02-17 ...	0.009249	-1	10742.8	-3.79
2000-02-18 ...	0.009205	-1	10742.8	0
2000-02-21 ...	0.008986	1	10483.5	-3.79
2000-02-22 ...	0.008942	-1	10479.7	-3.79
2000-02-23 ...	0.008898	-1	10479.7	0

Figure 4-16: Example of the result for signal creation step

beginning of the next day (2000-02-09), this share was bought to the value of £9,996.21 (£10,000 less the brokerage fee of £3.79). This research uses the closing price of the purchase day as the purchase price. The signal would have been generated the prior day. The reasoning behind this is that executing an order can sometimes take a long time, so we considered it an appropriate and simple approach to use the last possible price for the execution. The opening price or an average price would also have been viable alternatives, as would using the high price to run a more conservative simulation, but in all cases the simulated prices will be slightly inaccurate owing to the slippage that occurs in reality. Additionally, as ours is not an intraday trading system and holds shares for multiple days, we do not expect the results to vary much among the different possible purchase prices. The first trade is shown in the green rectangle. After this share was purchased, the price continued to increase. Then, at the end of 2000-02-16, the model sent out a -1 signal, instructing to sell. Therefore, the next day (2000-02-17), all of the shares were sold, closing the first trade. At the end of this trade, our money increased to £10,742.80 after paying an additional brokerage fee of £3.79 to close the trade.

To summarise, in this example of the first trade's performance, we started with 10,000 and ended up with 10,742.80, which means that we made a profit of 742.80. As this trade started on 2000-02-09 and ended on 2000-02-16, we held this trade for a period of 8 days in total (including the weekend when the market is closed). Therefore, we made 7.4% during the first 8 days of our trading.

The second buy signal appears on 2000-02-18, which is only two days after the previous sell signal. The period of the second trade is shown in the red rectangle, starting with the buy signal on 2000-02-18. A buy position is opened the next trading day after the weekend (2000-02-21). However, the buy signal is not as the previous time. This signal happens only for one day before the sell signal. Therefore, we only held on to this trade for a day. The price dropped on that day so we ended up losing money. After exiting this position our money declined from 10,742.80 to 10,479.70 or -2.45%.

To summarise the second trade, we held on to the trade for only a day and ended up with a 2.45% loss or around 263. However, this loss isn't only due to the decrease of the price, it is made larger by the brokerage fee, which had to be paid twice in quick succession. Although the signal produced by our system was wrong this time, it did not last long. The model realised the fault in its decision very quickly and brought us out of the wrong position straight away to prevent a bigger loss, which could have happened as the price continued going down after we exited the position.

4.5.3 Trading Evaluation

After finishing the simulation in the previous step, the results shown in Figure 4-16 on page 108 have their performance calculated and evaluated using the following process. Performance cannot be measured by profit only. Higher profit does not always indicate a better model since it is not going to be a good model if a trader can make a lot of profit but has to take massive risks at the same time. Therefore, the most important value that we use to evaluate our trading performance is the *Sharpe Ratio*, also known as the *Risk Adjusted Return*. The Sharpe Ratio adjusts the return by the risk that an investor had to take to achieve it. The Sharpe Ratio can be calculated as shown in Chapter 2 section 2.3.4 on page 42.

Other metrics will also be calculated alongside the Sharpe Ratio. There are two types of other measurements we will feature, relating to money (both profit and loss) and risk. In terms of money-related metrics, we will calculate the annual profit as mentioned in Chapter 2 section 2.3.4 on page 42 and the brokerage fee that the trader has to pay.

As for the risk-related metrics, we calculate the annual volatility and maximum drawdown, which is the biggest percentage decline over a given period of time. A bigger drawdown indicates more difficulty to return to a profit for a trading strategy. More details about drawdown can be seen in Chapter 2 section 2.3.4 on page 42

An example of results from the trading evaluation process can be seen in Figure 4-17, page 111. This result comes from a stock in the UK market called CARR, representing CARR's Group plc. The results are shown in four sub-graphs. The first graph on the top shows the closing price. The X-axis represents trading days and the Y-axis shows the closing price. The areas in green show the periods when we have positions open (holding shares that we have bought). These periods correspond with the second graph which illustrates the signals obtained from the model. When the signal in the second graph is 1 (on the Y-axis), this period of time will be coloured green in the top graph as well, representing when we have bought shares.

From these two graphs at the top, it can be seen that in the period of trading from the beginning of 2017 to then end of 2018, we opened positions (bought shares) 199 times and paid total brokerage fees of £754.21. There are few positions open for long before shares are sold again.

The third graph shows the balance in our portfolio when we invest with initial funds of 10,000 and we have bought and sold shares following every signal from the system. This balance has already taken the brokerage fees into account. At the end of the trading period, we obtained a profit of 79.17% with a Sharpe Ratio 1.12.

The bottom graph shows the logarithm return (Y-axis), comparing our model's result (green) and the buy & hold strategy (red). The brokerage fee is not included in these graphs. The result shows that even though our model does not work very well from the beginning until about August 2017, it is still be able to track the buy & hold. Then, after August 2017, it shows an extraordinary result which is much better than the buy & hold until the end of the test period.

The result's summary of Figure 4-17 is shown in table 4.1.

It can be seen from table 4.1 that our model performs better that the buy & hold strategy, as it has a higher Sharpe Ratio and profit. Moreover, our model shows a better result in terms of risk as it provides lower volatility and a much smaller drawdown. The buy & hold strategy has negligible brokerage fees, while for our model the total fee is £754.21 as it placed 199 orders.

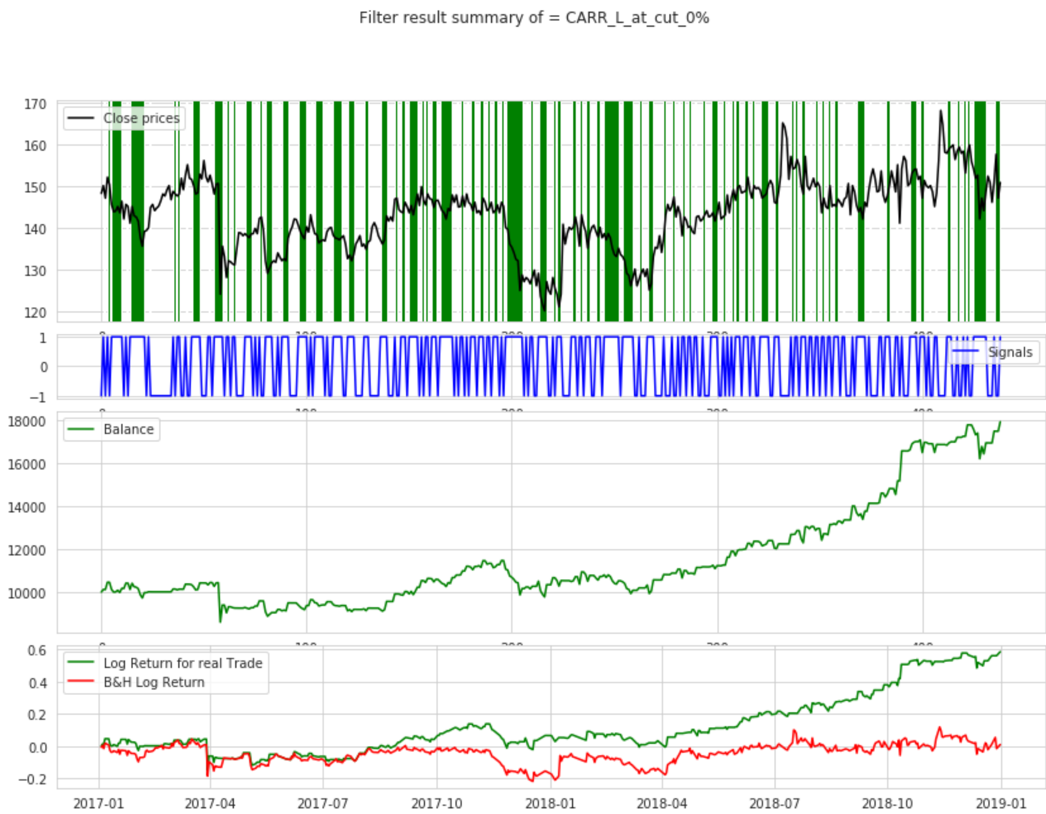


Figure 4-17: Result from signal creation of CARR when the initial fund is 10,000

Performance	Buy & Hold strategy	Our model
Sharpe Ratio	0.01	1.12
Profit (%)	0.92	79.17
Volatility	0.386	0.3
Maximum Drawdown (%)	-23.08	-17.77
Number of Trades	2	199
Brokerage fee	£7.59	£754.21

Table 4.1: Performances summary of CARR comparing between our model and the buy & hold strategy

4.5.4 Signal Optimisation

This is the process of adjusting the number of trades by filter out some weak trades. Weak trades here means the trades that related to the expected small returns. In this research, we consider a trade with higher profit as stronger trade or stronger signal. The signal optimisation process will improve the quality of trading by filtering out weak signals in order to decrease the effect of brokerage fees since fewer trades will be placed. During signal optimisation, initial funds are taken in to account. With lower amounts of initial funds, more signals will be filtered out to decrease the impact of brokerage fees and reduce the risk of following unclear signals. On the other hand, if traders start with more money, the system may retain more signals even if they are not the strongest ones as they are worthy to trade even when considering the brokerage fee. If taking this module out of this step, an investor will try to follow every signal the system had given. This could be problematic, especially, when the investor has limited amount of fund. The experimental results to support this step can be found in chapter 5, section 5.5.2, on page 212.

This optimisation process takes the development set and run grid search on it to find the optimum point to filter signal for the test set. The signals are filtered by considering the level of expected returns. Basically, signals with small expected returns will be filtered out as they indicate a smaller amount of profit.

To make this easy to understand, we will provide example as seen in figure 4-18.

Date	Close	Log Rets	Predict Value	Predict Signal	Strategy Return (before filtering)	Strategy Return (after filtering)
2013-01-02	105	NaN	NaN	NaN	NaN	
2013-01-03	104.8	-0.00190655	-0.00000012	-1	0.00190655	0
2013-01-04	106	0.0113853	0.05241254	1	0.0113853	0.0113853
2013-01-07	104.75	-0.0118625	-0.00154876	-1	0.0118625	0.0118625
2013-01-09	102.5	-0.0217138	0.000000004	1	-0.0217138	0
2013-01-10	103.5	0.00970881	0.045896000	1	0.00970881	0.00970881

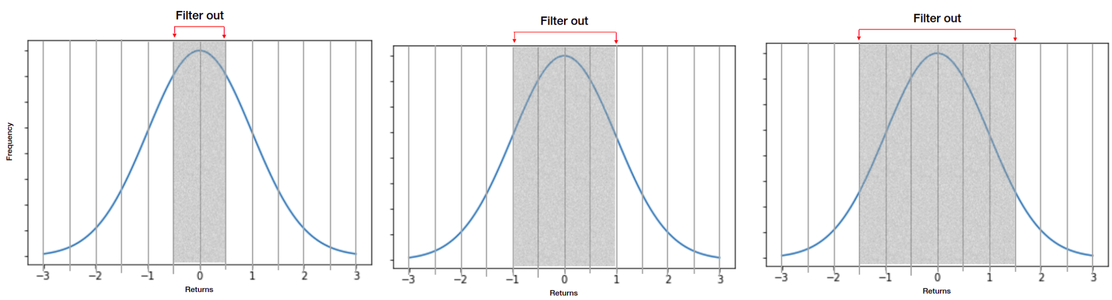
Figure 4-18: Example of the signal filtering

Figure 4-18 shows example of the result from our system performing on the development set. There are four different parts, showing in different colours. The first two columns (Close and Log Rets), shown in grey, indicate the closing price and the logarithm returns. The next two columns (Predict Value and Predict Signal), shown in green, are the result from our predictor(s). The predicted value is the forecast return that our predictor had made, while the predicted signal is the signals which our predictors provided. The last two columns (Strategy Return before and after filtering), shown in yellow and blue, are the return our system will obtain before and after performing this signal optimisation (signal filtering) step.

Without signal optimisation, our predicted value for the day 2013-01-03 is negative (column predict value), -0.00000012. Since the actual return on that day is also negative (-0.00190655), the strategy return in the yellow column is calculated as positive (0.00190655) as the system got the direction correctly. On the other hand, on the day 2013-01-09, the actual return is negative, however, our system predicted positive result. Therefore, the strategy return in the yellow column is negative as the system got the wrong answer.

Now, we will look at the same table but the result will be after performing signal optimisation. Let's say, the optimisation module provide the information that best signal cutting area is between -0.0005 and 0.0005. Therefore, the predict values that fall in this area will be filtered out. This makes the system filter our the signal on the day 2013-01-03 and 2013-01-09. This makes the strategy return on these two day equal two zero. This is how the signal filtering works. It runs grid search on the predicted return (or predicted value) to find the best cut off area. The best cut off area means if the signals in this area got filtering out, the cumulative profit in the blue column will be higher. In fact, the key value we used to optimise the signal is Sharpe Ratio, therefore, the optimum cutting area is actually the area leading to the maximum Sharpe Ratio. Once obtaining the best cutting area from the development set, this parameters will be taken to filter out the signals in the testing set.

To perform this optimisation in order to get the cutting area, we take the distribution of predict values (predicted returns) and run grid search as shown figure 4-19 and 4-20.



(a) (b) (c)

Figure 4-19: Signal filtering using the returns' symmetric distribution

In order to filter the signals, signals are firstly filtered out symmetrically as can be seen from the example in Figure 4-19. This graph shows return distribution of the training set. The grey areas in the graphs show areas where signals are filtered out. From sub Figure A, we start filtering out signals in the grey area in the middle, which are signals

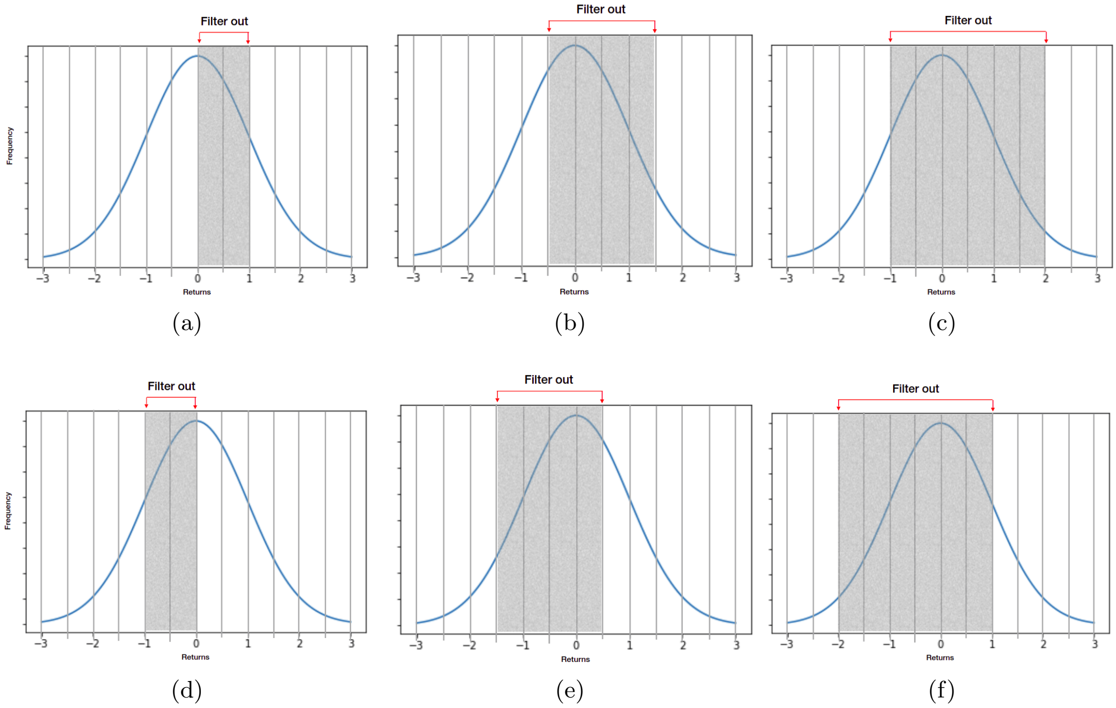


Figure 4-20: Signal filtering using the return distribution

with low expected returns (near 0). Then, the size of the grey area is increased which means that more signals will be filtered out, as can be seen in sub Figures (b) and (c). In this system, between 10% and 90% of signals will be filtered out and the number that provides the maximum Sharpe Ratio will be selected.

Signals will be optimised using asymmetric filtering as well. An example of asymmetric filtering can be seen in Figure 4-20 on page 114. In sub Figures (a) to (c), more buy signals are filtered out than sell signals, which might be a good way to deal with stocks with a left-skewed distribution when starting with less money. This means an investor does not have to open as many positions. On the other hand, in sub Figures (d) to (f), the sell signals are filtered out more than the buy ones. This might be used for stocks with a right-skewed distribution as they tend to have too many sell signals which will make the investors have to sell their positions too often. Selling too often will affect the performance of the trade greatly, especially when starting with less capital.

After finishing the optimisation process including both symmetric and asymmetric filtering, the best parameters or the regions that provide the maximum Sharpe Ratio will be used to filter the signals from the test set.

4.6 Step 6: Testing

This is the last step of the model. After every predictor was trained, the scoring system was created and all the parameters for signal optimisation were obtained, we put them all together in order to work with the last part of our data, the testing set. The testing set is shown in the green database sign in figure 4-1 on page 89. This data is completely unseen by our model and we will use it to evaluate our model's performance against other benchmarks and other methods. The framework for this step is shown in Figure 4-21.

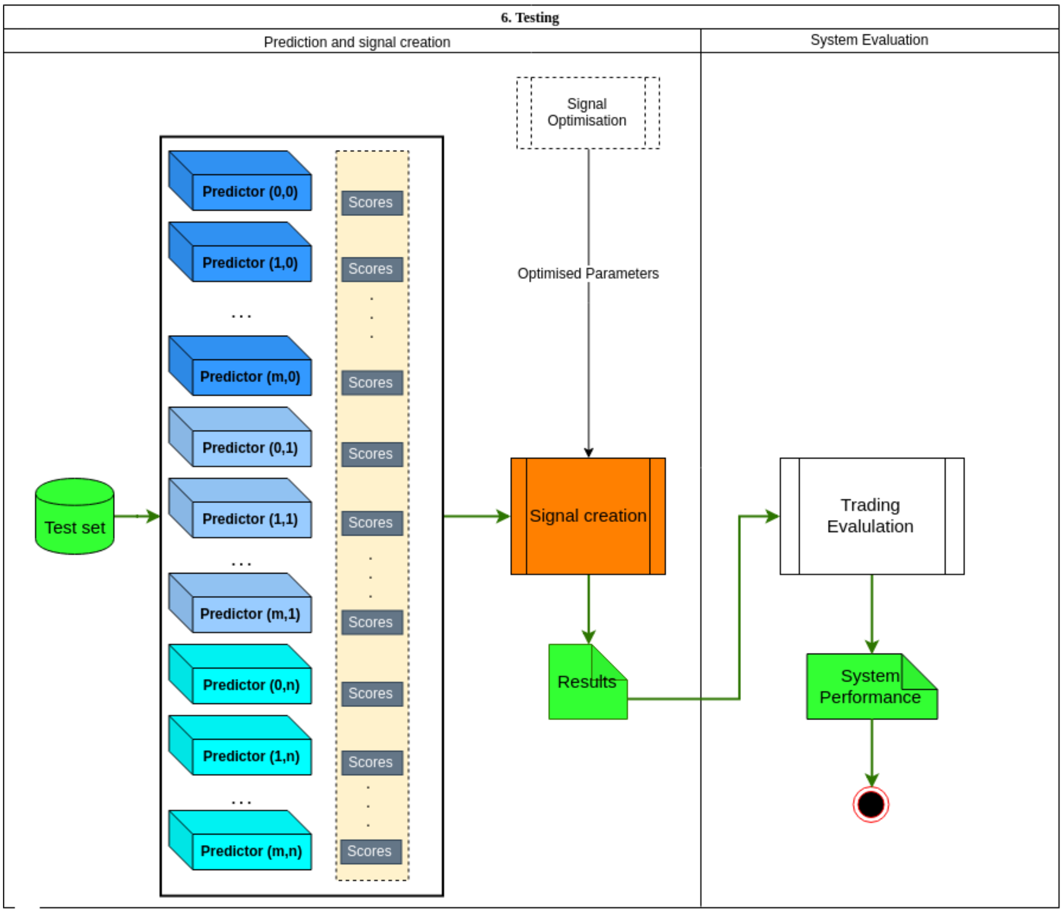


Figure 4-21: Testing process diagram

Figure 4-21 shows that the testing process is composed of two sub tasks, *prediction and signal creation* and *system evaluation*. Firstly, the testing set will be fed into the completed set of predictors which have been trained and have their scores ready to be employed. Test data will be fed into these predictors daily and the predicted result will be put into the signal creation process, for which the parameters were obtained from the optimisation process in the previous step. Each day, if there is a signal sent out from the system, the result will be stored in the result file.

After finishing the process of prediction and creation for all of the testing data, the result, which contains all signals for the days that the system recommended to buy or sell, is sent to the trading evaluation process. The evaluation process will calculate the profit of each trade including the fee, followed by calculating other metrics mentioned in section 4.5.3 on page 109. An example of the final result can be seen in the following table:

Metric	Result
Sharpe Ratio	1.13
Profit (%)	53.36
Daily Volatility	0.01
Yearly Volatility	0.166
Maximum Drawdown	-16.49
Average Drawdown	-3.7
Number of trades (days)	28
Brokerage fee (£)	106.12

Table 4.2: Example of system evaluation on MCD during 2017-2018

Table 4.2 shows the final result when testing on MCD with initial funds of £10000. At the end of the process, the eight values provided indicate the performance of our system on the selected dataset. For MCD, testing over two years from 2017-2018, the system provides a 53.36% profit with daily and yearly risk (volatility) of 0.01 and 0.166, respectively. The maximum drawdown is -16.49% which indicates the maximum loss that an investor has to accept throughout this period. The average drawdown is 0.01%. During these two years, the system trades 28 times in total (including both buying and selling), which makes the brokerage fees equal to £106.12, this is only just over 1% of the initial funds.

Chapter 5

System Design

This chapter explains the process of designing our system step-by-step. In order to understand this chapter clearly, we refer to the previous chapter at some points. There are four main experiments in this chapter which will lead the reader to the final design of our system. The first three experiments test the main design of our research and are presented in order, while the fourth is an additional experiment. It provides small adjustments to the system which are not critical, having only a minor effect on the main design of our trading system. The following briefly explains the structure of the chapter.

The first experiment aims to investigate our original idea about the benefit of having multiple predictors and training them differently, so they each act as experts on specific characteristics of the data. In order to perform this experiment, we have to control the predictor selection process to make sure that the correct predictor is selected for the given time. In order to do that, look-ahead bias is included in this experiment. At the end of this experiment, we show the results that support our idea that having multiple predictors is better than a single predictor. The second experiment is designed to remove the bias and make sure that the conclusion from the first experiment remains correct. This experiment is set up the same way as experiment 1. The only difference is the removal of the bias from the predictor selection method. Experiment 3 investigates different predictor selection methods. In this experiment, four different predictor selection techniques will be discussed. After finishing these three experiments, we complete the design of our scoring system, which is the heart of this research. Then, the last experiment is composed of several smaller experiments designed to provide adjustments to our system.

5.1 Experiment 1: Proof of concept

The objective of this experiment is to establish that our original idea of having multiple predictors and machine learning algorithms is helpful for our trading system. Please note that in this research,

- *Multiple Predictors* means having more than one predictor, but all of them could be the same algorithm (created from same machine learning model) but trained differently.
- *Multiple Algorithms* means having more than one machine learning model working together (as opposed to one algorithm having multiple predictors).

This experiment comprises two sub experiments. Experiment 1.1 aims to investigate whether multiple predictors work better than a single predictor. Experiment 1.2 aims to investigate whether having more machine learning algorithms is better than a single algorithm.

5.1.1 Experiment 1.1: Multiple Predictors

In this experiment, the training set is clustered into smaller groups. Multiple predictors are then created (from the same machine learning algorithm) according to the number of clusters. Each predictor will be trained on a specific cluster of data. Our hypothesis is that predictors which are trained on different clusters will act as an expert for a specific characteristic of the data. This means each predictor should work best at different times. So, when performance drops for one, another is at its best. By having them work together in this way, our trading performance will be improved.

5.1.1.1 Experiment design

The structure of this experiment follows the system framework in chapter 4, figure 4-1 on page 89 up until step 3. However, there are a number of points which are designed differently as this experiment is the very first experiment we worked on, while the framework in chapter 4 is for the completed model. The details below explain all of the differences and demonstrate how this experiment is set up step-by-step.

1. Data Preparation

Stock data was loaded and cleaned following the processes described in chapter 4, section 4.1, on page 4.1. The dataset was divided into two sets, instead of three. The two sets of data are for training and testing. The training set covers from the year 2000 until 2012, and the testing set ranges from the beginning of 2013 to the end of 2014.

Then, clustering and prediction features were created, as described in section 4.1.3 on page 91. The clustering features are the standard deviation and mean of the returns. For the prediction features, we created a series of the past five-day logarithmic returns. Examples of the clustering features are shown in figure 5-1.

Date	Close	Rets	Rets_std	Rets_mean
2000-01-03	188.25	NaN	NaN	NaN
2000-01-04	181.75	-0.035139	NaN	NaN
2000-01-05	184.75	0.016371	NaN	NaN
2000-01-06	184.75	0.000000	NaN	NaN
2000-01-07	182.25	-0.013624	NaN	NaN
2000-01-10	176.50	-0.032058	0.021710	-0.012890
2000-01-11	170.50	-0.034586	0.021570	-0.012779
2000-01-12	168.25	-0.013284	0.014453	-0.018711
2000-01-13	167.50	-0.004468	0.013080	-0.019604
2000-01-14	169.00	0.008915	0.018442	-0.015096

Figure 5-1: Example of TSCO training data, including mean and standard deviation of the returns

Figure 5-1 shows an example of the training data. The dataset here is for Tesco PLC. The training data starts from the beginning of January 2000 as can be seen from the index of the table. The first column is the closing price, shown as *Close*. The second column, *Rets*, shows the logarithmic return of that day, which can be calculated following equation 2.25 on page 43. As the return is computed using the closing prices from the previous day and the current day, the return of the first day cannot be identified and is shown as NaN. The third and fourth columns, named Rets_std and Ret_mean, represent the standard deviation and mean of the returns, respectively. These values are calculated from the returns over the past five days, including the present day. For example, the Rets_std at row index 2000-01-10 is calculated from the standard deviation of the return from day 2000-01-04 to 2000-01-10, which are -0.035139, 0.016371, 0.000000, -0.013624 and -0.032058.

2. Clustering

   In this step, we take the clustering features created in the previous section and cluster the training set into two groups as shown in figure 5-2 .

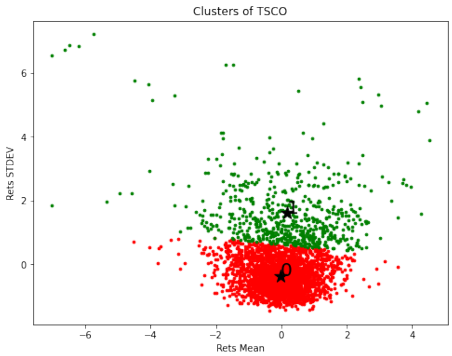


Figure 5-2: Example of the clusters of TSCO's training data

   Figure 5-2 shows the clusters of the training set of TSCO. The X and Y axis are the mean and standard deviation of the returns. Since k-means is one of the algorithms which is sensitive to different scales of data, the clustering features were normalised before being used by the k-means cluster. Clusters 0 and 1 are represented by red and green in this figure. For ease of reference, we name them the low and high volatility groups, respectively. The black stars at the middle of each group represent the centre of each. These two coordinates will be used later in order to select the predictor in the testing process.

3. Training In this step, we initialise two polynomial regression predictors according to the number of groups from the previous step. The detail of this step is described in section 4.3, on page 93. These two predictors will be named *predictor 0*, which was created for the low volatility group, and *predictor 1*, which was created for the high volatility group.

   Figure 5-3 shows that there are two clusters of training data. Cluster 0 contains data with low volatility while cluster 1 is composed of high volatility data. We initialised two predictors, predictor 0 and predictor 1. In this experiment, we used only the polynomial regression model for both predictors. These predictors were created with exactly the same initial parameters. Therefore, they were identical at the beginning.

   After creating the predictors, we trained them with different groups of data.

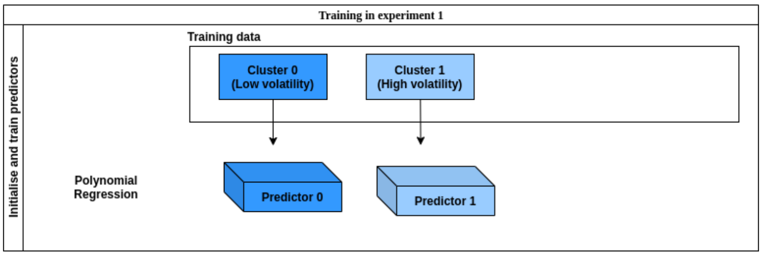


Figure 5-3: Training process in experiment 1.1

Predictor 0 was trained on cluster 0 training data, while predictor 1 was trained with cluster 1 training data. Using this method, we expected predictor 0 to be specialised for low volatility data. On the other hand, predictor 1 should work better with high volatility data. Therefore, predictor 1 will be used during fluctuating periods, but in calmer times the system will select predictor 0.

Ideally, the system should be able to switch between high and low volatility predictors. For example, during periods of high volatility, a predictor which was trained specifically on high volatility data will be called. On the other hand, another predictor which was trained with lower volatility data should be selected. If our hypothesis works as expected, these methods could provide better results than having only one predictor.

4. Testing

In the completed system shown in chapter 4, figure 4-1 on page 89, there are three steps after the training. However, this first experiment is not as complicated. At the time of this experiment, the scoring system and the signal optimisation process had not yet been designed. The trading fee was not included either.

Instead of using the scoring system to select the suitable predictor(s) during the test, as mentioned in the completed system, the method we used to select the predictor in this experiment is only the standard deviation and mean of the logarithmic return (as shown in the Rets_std and Rets_ mean columns). On each day of testing, the standard deviation and mean of the returns over a one week period are calculated and normalised with the same parameters obtained from the training set. Then these values will be fed into k-means clustering. The k-means clustering will classify the data into either cluster 0 or 1, based on the shortest distance between them and the centre of each group. If the cluster turns out 0, predictor 0 will be selected. Otherwise, predictor 1 will be selected. The process of switching between these two predictors will go on until the end of the test.

Date	Close	Log Rets	Rets_mean	Rets_std
2013-01-02	342.750000	NaN	NaN	NaN
2013-01-03	350.000000	0.020932	NaN	NaN
2013-01-04	349.450012	-0.001573	NaN	NaN
2013-01-07	349.299988	-0.000429	NaN	NaN
2013-01-08	351.500000	0.006279	NaN	NaN
2013-01-09	349.149994	-0.006708	0.003700	0.010686
2013-01-10	355.399994	0.017742	0.003062	0.009420
2013-01-11	353.200012	-0.006209	0.002135	0.010188
2013-01-14	351.200012	-0.005679	0.001085	0.010772
2013-01-15	349.600006	-0.004566	-0.001084	0.010554

Figure 5-4: Example of TSCO's testing data in experiment 1.1

Figure 5-4 shows the testing dataset of TSCO stock. As the testing data was completely separate from the training set and unknown to the predictors, the process of calculating the return and standard deviation had to start from the beginning of the testing data. Therefore, we could not calculate the return on the first day (2013-01-02), so this is shown as Nan. The Ret_mean and Ret_std columns are the mean and standard deviation of the returns over the first week (2013-01-02 to 2013-01-08), so these values could only be calculated after the first week. Therefore, the mean and standard deviation of the first week are shown as NaN too.

Please note that we deliberately include the return of the present day in the calculation and are aware of the look-ahead bias this causes. The reason for this is to control the correctness of the predictor selection and to be able to support our hypothesis that if we can select the suitable predictor(s) for the time, then having multiple clusters and predictors will be helpful. Then, in section 5.2 on page 147, we study selecting the correct predictors and investigate the most suitable strategy to get rid of the look-ahead bias issue.

After the prediction had been made, the signal creation was simply that if the predictor predicts a positive return, the stock will be bought. On the other hand, if the predictor predicts a decreasing price, we will sell that stock the next day. The system profits if predicting 1 (increasing price) on days that the stock goes up. If the stock goes down, it loses money.

5.1.1.2 Experiment results

We performed this experiment on a variety of stocks from different sectors, ranging from food producers, retailers, general industrial, construction and electronics to computer and technology companies. These stocks are mainly from the UK (listed on the London stock Exchange) and some of them are well-known companies from America (NASDAQ and NYSE markets). The list of stocks that have been selected can be seen in table 6.1 on page 233. The results include the cumulative profit over the testing period of two years, the Sharpe Ratio, maximum drawdown, accuracy and standard deviation. More details of the evaluation process can be seen in chapter 4, section 4.5.3, on page 109.

We ran this experiment on 11 different stocks as mentioned above. In this section, we present a few visualisations. All results will be shown in table 5.1 later.

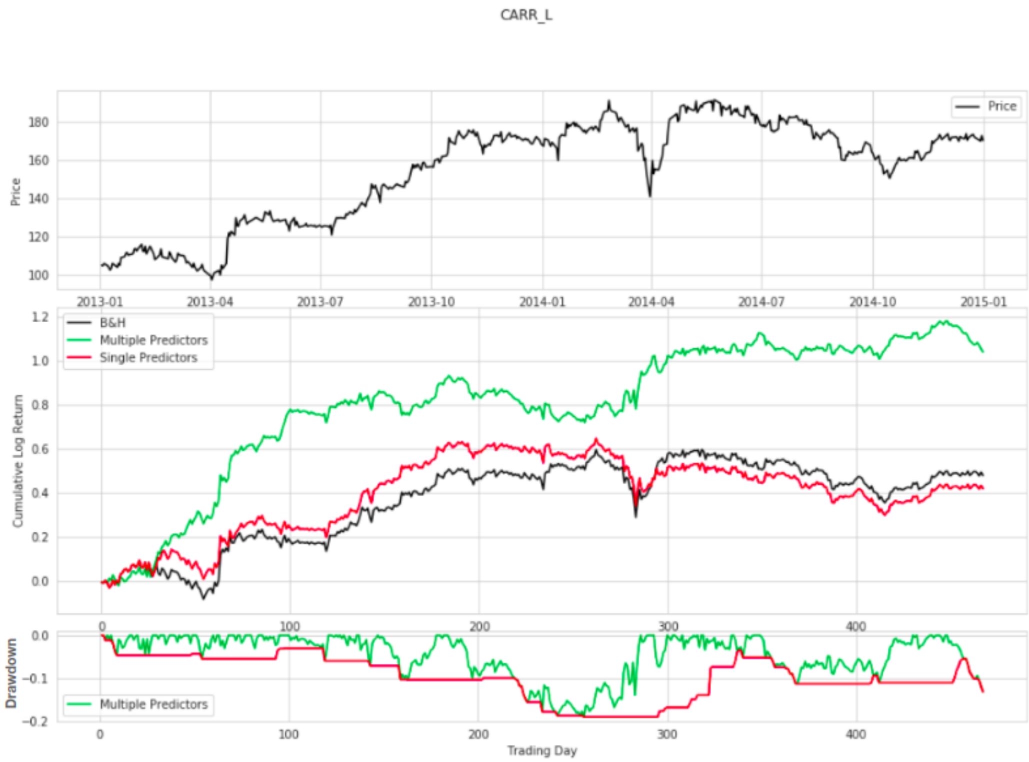


Figure 5-5: Result from experiment 1.1 on CARR

Figures 5-5 and 5-6 show the results from this experiment on CARR and TSCO stocks, respectively. Each figure comprises three sub-graphs. The top one shows the closing price of the stock. The middle graph compares the cumulative log return of three

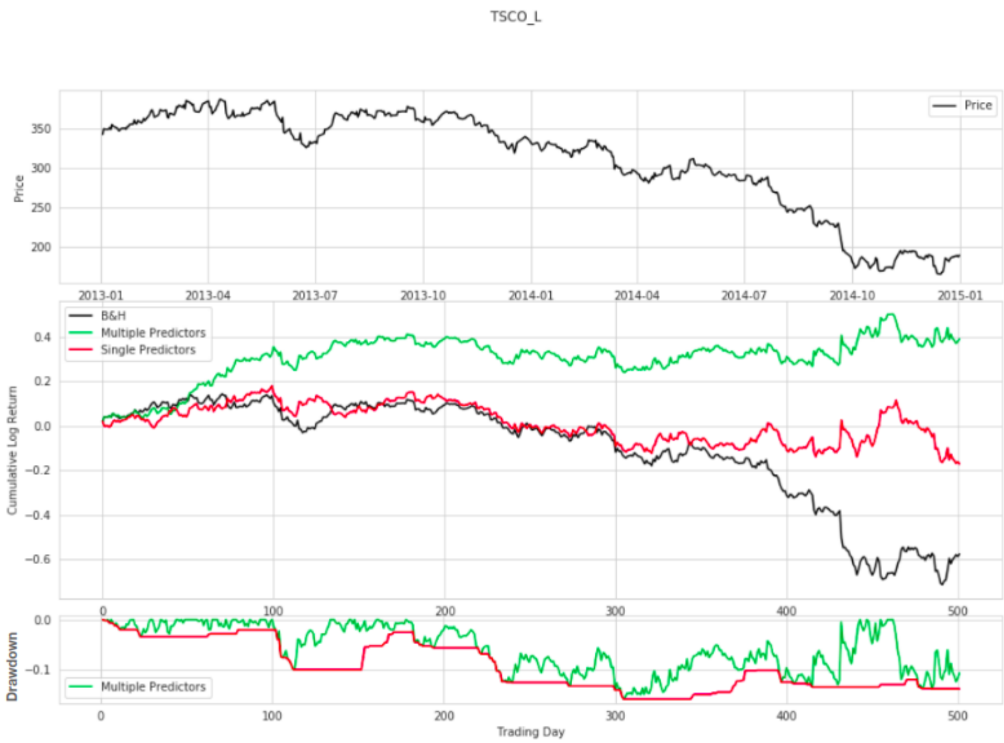


Figure 5-6: Result from experiment 1.1 on TSCO

different strategies: buy & hold (black line), multiple and single predictors (green and red graphs). The bottom graph shows the drawdown throughout the period using the multiple-predictor strategy (green line). The maximum drawdown of each period is shown by the red line.

It can be seen from figure 5-5 that the price of CARR stock starts just above 100 pence and goes in an upward direction overall during the testing period, until ending up at almost 180 pence. There are two notable drops at the beginning of April 2013 and 2014. We can see the outstanding result from the multiple predictors in the middle graph. The multiple-predictor model (green line) performs much better than the other strategies throughout. The single predictor (red) did not seem to work much better than the buy & hold, as we can see the red graph mostly tracks the black graph (buy & hold). Even though the single predictor overtook the buy & hold at the beginning, it suffered during both the April 2013 and 2014 declines and performed worse than buy & hold after the latter.

On the other hand, the multiple-predictor strategy performed very well as can be seen from the green line, which is much higher than the others throughout almost the whole period except the first two months. It also performed well during both the sharp drops in 2013 and 2014. It can be seen that when the price drops in April 2013 and 2014, the multiple predictor strategy still increased.

Finally, the buy & hold strategy ends up with 62.38 % profit over two years. The single predictor could not do better and ends up with lower profit than the buy & hold, at 52.92 %. Meanwhile, the multiple-predictor model finishes with an outstanding profit of 183.38 %, which is almost three times higher than the benchmark (buy & hold). Not only is it better than the others in terms of profit, but the multiple predictor strategy also experienced a smaller drawdown. While the other strategies had maximum drawdowns of more than 25 % (26.32 % and 29.33 % drawdown from the buy & hold and the single-predictor strategies respectively), using multiple predictors gave a maximum drawdown of only 19.05 %.

The-multiple predictor strategy also provides a much higher Sharpe Ratio than the others, suggesting higher profit with lower volatility. The multiple-predictor strategy has the highest Sharpe Ratio of 1.78, while the buy & hold and single-predictor strategies achieved only 0.77 and 0.66 respectively. The accuracy of multiple predictors is also higher than the others, but not by very much. Its accuracy is over 54 % while the buy & hold and single predictor only managed about 52%. However, as we mentioned in section 5.4.1, on page 192, accuracy is not the most important measure to evaluate trading performance. We are not concerned about this value unless it is dramatically different.

Figure 5-6 shows the result from another UK stock: TSCO. The period of testing is the same as the previously presented result (CARR), starting from the beginning of 2013 until the end of 2014. However, the character of TSCO is totally different from CARR. While the price of CARR goes up, TSCO's price goes down throughout the period of testing. The price starts from about 350 pence and ends up below 200 pence. Even though the character of the TSCO share price is different, the results are similar to CARR. From the second graph we can see the multiple-predictor strategy (green line) performs much better than the other strategies throughout the period of testing, except in the first few months. During the first few months of testing, both the single and multiple predictor strategies could not perform better than the buy & hold. But it does not take long for the multiple-predictor strategy to outperform the buy & hold, which it does after the 3rd month and then continuously performs better until it ends with a profit of 44.84 % in two years. Meanwhile both the buy & hold and single predictors end up losing 44.86 % and 17.42 %, respectively.

In addition to greater profit, the accuracy of the multiple-predictor model is also higher than the other models. It obtains about 50.5 % accuracy while the others only have 46.91 %. The drawdown of multiple predictors is lower than the others too. Almost all of the key measurements are better when using multiple predictors; the Sharpe Ratio, which is the most important value here, is positive (0.64) while the others obtained negative values.

Although using multiple predictors performed very well in both of the stocks we mentioned above, this was not always the case. There are a few cases where it did not perform as well as expected, such as the results from MKS and FERG shown in figures 5-7 and 5-8, respectively. Therefore, further investigation will be performed later in this chapter.

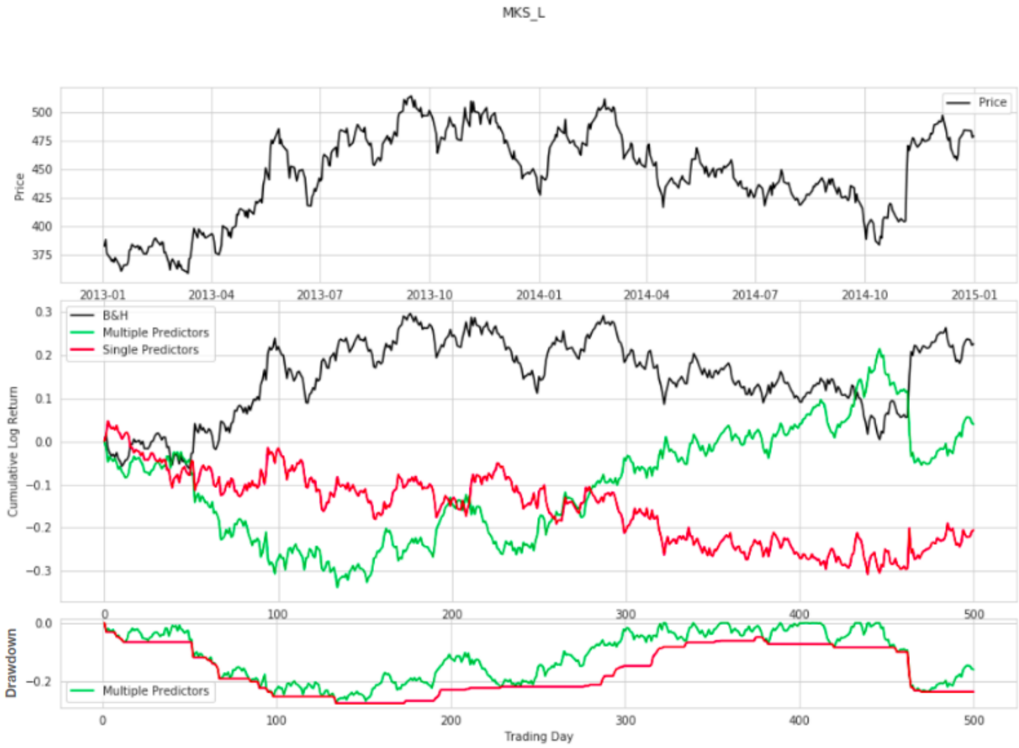


Figure 5-7: Result from experiment 1.1 on MKS

Figure 5-7 shows the result from MKS. As we can see from the middle graph, both the single and multiple-predictor strategies did not perform well. Both of them underperformed the benchmark (buy & hold). There was only a small period before the end of the test (between September and November 2014) that multiple predictors outperformed buy & hold. Performance declined again shortly after. The single predictor

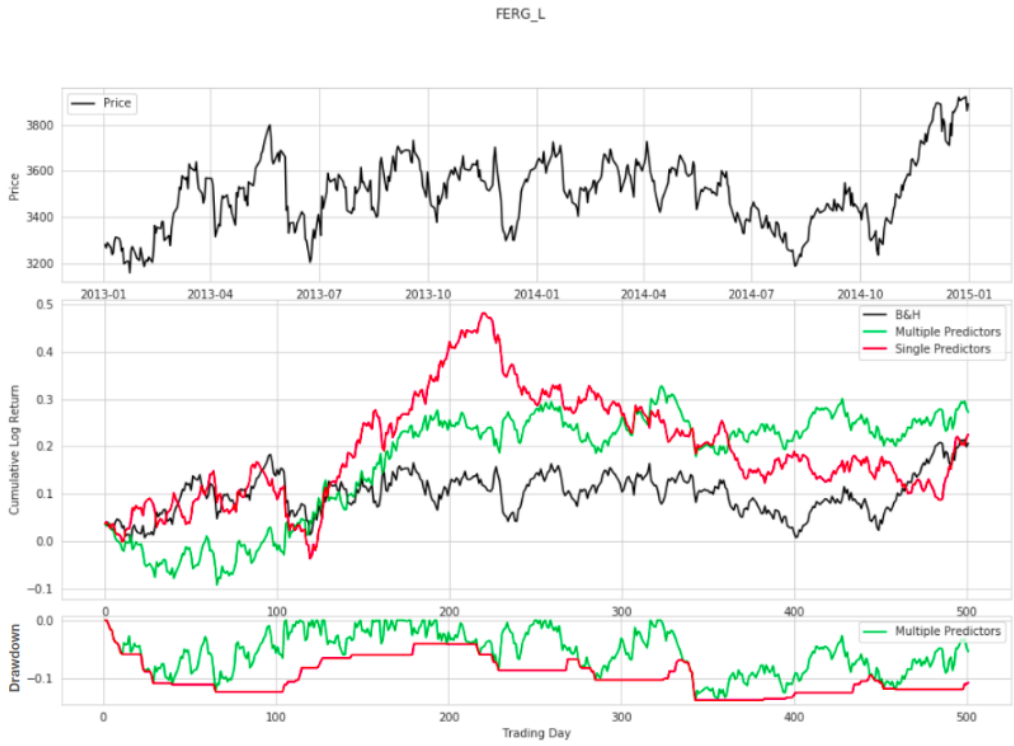


Figure 5-8: Result from experiment 1.1 on FERG

seemed to work better than multiple predictors in 2013. On the other hand, shortly after January 2014, the multiple-predictor model showed a much better result than the single predictor. The best strategy for this specific dataset is the buy & hold, which gained more than 25 % profit over two years, followed by the multiple predictor strategy which was able to make just over 4 %. The single predictor performed worse than the others and ended up with a loss over 18 %.

The second dataset that did not go as well as we thought is FERG. In figure 5-8, the middle graph shows that multiple predictors performed significantly better than the buy & hold strategy throughout the testing period, except for the fist half of 2013. As for the single predictor, it also performed much better than the buy & hold, even better than the multiple-predictor model most of the time, however the performance was less stable. To decide on the best strategy for this data, we consider the Sharpe Ratio because it takes into account both profit and risk. From the Shape Ratio in table 5.1, the best Sharpe Ratio is 0.42 and belongs to the multiple-predictor strategy, followed by the single predictor with a Sharpe Ratio of 0.31. Although the result from the multiple predictors is not we expected, it can still be considered the best strategy in this test.

There are many reasons that the results from MKS and FERG were not as good as expected. Firstly, the price of these two stocks was ranging, while CARR and TSCO had clear overall trends (see figure 5-5 and 5-6). Therefore, there is a possibility that multiple predictors might not be suitable to deal with datasets with ranging patterns. Secondly, in figure 5-7 it can be seen that both the single and multiple-predictor strategies did not perform better than the buy & hold. The reason for this might be because of the limited predictor used. In this experiment, we used only one machine learning predictor (polynomial regression). This predictor might not be a suitable predictor for this data, resulting in both strategies not performing well. Finally, the problem might relate to the criteria for predictor section. To investigate further, we will experiment with adding more machine learning predictors in section 5.1.2 on page 131 and testing different criteria for predictor selection in section 5.3 on page 159.

After discussing some of the results for both good and bad cases, we show all of the results from the 11 different datasets in table 5.1. This table shows the comparison of three different strategies, which are the buy & hold, single predictor and multiple predictor strategies. The values presented are profit, Sharpe Ratio, drawdown, accuracy and standard deviation. Please note that the accuracy of the buy & hold in this research refers to percentage of the days when a stock's price goes up. The details of how these values are calculated can be seen in chapter 2 section 2.3.4 on page 42. The details of each company in this table can be seen in table 6.1 on page 233.

Table 5.1: The comparison between Buy & Hold, Single and Multiple predictor methods

Data & Model	**Profit (%)**	**Sharpe Ratio**	**Max DD (%)**	**Accuracy (%)**	**Stdev**
CARR					
Buy & Hold	62.38	0.77	-26.32	52.89	0.302
Single Predictor	52.92	0.66	-29.33	52.46	0.302
Multiple Predictors	183.38	1.78	-19.05	54.39	0.300
COST					
Buy & Hold	11.57	0.09	-22.26	51.26	0.296
Single Predictor	81.97	0.97	-15.72	51.46	0.295
Multiple Predictors	-14.54	-0.38	-38.90	49.37	0.296
D4T4					
Buy & Hold	39.10	1.19	-37.76	51.75	0.586
Single Predictor	-27.22	-1.25	-46.08	49.12	0.586
Multiple Predictors	70.71	1.98	-17.22	58.77	0.583
FERG					
Buy & Hold	18.59	0.26	-16.09	49.10	0.211
Single Predictor	20.85	0.31	-32.67	48.70	0.211
Multiple Predictors	26.61	0.42	-13.84	50.90	0.211
GOOG					
Buy & Hold	45.71	0.74	-18.72	50.10	0.213
Single Predictor	48.67	0.79	-15.49	50.50	0.213
Multiple Predictors	-8.45	-0.35	-24.30	48.31	0.213
MACF					
Buy & Hold	26.96	0.65	-28.06	49.81	0.312
Single Predictor	-1.83	-0.15	-36.28	52.92	0.313
Multiple Predictors	114.62	2.33	-19.23	58.37	0.309
MCD					
Buy & Hold	3.97	0.08	-14.61	54.6	0.124
Single Predictor	1.78	0.17	-14.61	54.2	0.124
Multiple Predictors	23.87	0.63	-10.33	54.2	0.124
MKS					
Buy & Hold	25.11	0.35	-25.24	48.8	0.237
Single Predictor	-18.61	-0.56	-29.86	48.6	0.237
Multiple Predictors	4.13	0.04	-27.50	53.2	0.238

Continued on next page

Table 5.1 – *Continued from previous page*

Data & Model	Profit (%)	Sharpe Ratio	Max DD (%)	Accuracy (%)	Stdev
OXIG					
Buy & Hold	-15.00	-0.33	-46.91	48.07	0.346
Single Predictor	21.61	0.20	-27.29	50.10	0.346
Multiple Predictors	357.01	2.19	-16.29	52.55	0.343
SXS					
Buy & Hold	0.67	0.10	-35.85	51.70	0.263
Single Predictor	-45.88	-1.30	-52.22	47.29	0.262
Multiple Predictors	67.36	0.88	-22.65	53.91	0.262
TSCO					
Buy & Hold	-44.86	-1.35	-57.50	46.91	0.244
Single Predictor	-17.42	-0.52	-29.66	46.91	0.244
Multiple Predictors	44.84	0.64	-15.80	50.50	0.244

Table 5.1 shows eight datasets - CARR, D4T4, MACF, MCD, OXIG, SXS, TSCO, FERG, - where the multiple predictor strategy performed as expected or better than the other strategies. Please note that for FERG, it did not perform as expected but still better than the other strategies. Some of these datasets already do well using the buy & hold strategy (profitable stocks) but the multiple predictors performed even better, for example CARR, D4T4 and MACF. Not only does the multiple-predictor strategy perform well on profitable stocks, but also unprofitable stocks (stocks that lost money with the buy & hold strategy), for example OXIG and TSCO.

Although the multiple-predictor strategy performed well on the stocks we mentioned above, there are three stocks that it could not work well with. These three stocks are COST, GOOG and MKS. To improve this, we will run more experiments later on in this chapter in order to find the most suitable model for a wide range of stocks.

We ran a paired T-test on the Sharpe ratios of these eleven datasets in the table to compare between the multiple-predictor and single-predictor strategies (polynomial regression). We obtained a p-value of 0.042, which indicates that our model was significantly better than the single predictor with greater than 95% confidence. The mean Sharpe Ratios of the single and multiple predictor strategies are -0.092 and 0.92, respectively. Although our system made a significantly higher profit, the standard deviation was slightly higher. This indicates slightly more variance in the Sharpe Ratio across stocks, however the p-value is small enough to show our results were significant.

We also ran a non-parametric statistical test called Wilcoxon signed-rank test on the Sharpe ratios of these results, comparing between the multiple-predictor and single-predictor strategies. The obtained two-tail p-value in this test is 0.0329 which also indicates that the multiple-predictor strategy was significantly different from the single predictor with greater than 95% confidence. Therefore, in the following experiments we continue using multiple predictors.

5.1.1.3 Discussion

This experiment was designed to investigate whether clustering training data, training a specific machine learning predictor on each group of data, and then putting multiple trained predictors to work together will improve stock price prediction. From the results, we can see that the multiple-predictor strategy performed very well on most of the stocks we tested on. Therefore, we conclude that having multiple predictors is helpful for the system. However, there were a few datasets for which this strategy did not work well. There are multiple possible reasons to explain the problem. Firstly, the machine learning algorithm we used in this experiment might not be good enough for all the datasets, or the criteria used to select the predictors was not suitable. Therefore, we will investigate these issues later on in this chapter in order to obtain a better model.

5.1.2 Experiment 1.2: Multiple Machine Learning Predictors

From experiment 1.1, we believe one of the problems that made some results turn out badly (for example MKS in figure 5-7 on page 126) was the insufficiency of the prediction model. This means that a single machine learning model is not good enough for some stocks. Therefore, this experiment aims to resolve this problem by having multiple machine learning algorithms added to the system. Our hypothesis is that having multiple machine learning algorithms will help achieve better results for a wider range of stocks. We will perform an experiment to test our hypothesis by using multiple well-known machine learning algorithms, such as linear regression, support vector machine and random forest. The reasons for selecting these well-known and simple algorithms are, firstly, that we would like to establish that our system is as good as or even better than more complicated machine learning algorithms, even though it is composed of simple algorithms. Secondly, the choice of machine learning algorithm is not the main concern of this research, as we are focused on selecting the best predictor for a period of time. Therefore, the system will be able to add more predictors in the future.

If trading performance can be improved by using multiple machine learning algorithms, we will then investigate further the techniques for predictor selection. Please note that in this experiment, we only aim to confirm that multiple machine learning models can increase the trading performance, but the technique to make them work together effectively (to select which algorithms/predictors for the time) will not be discussed in this section. Also, the biases are again included in this experiment.

5.1.2.1 An issue from the previous experiment

In the previous experiment (5.1.1), we found that the machine learning algorithm we used, polynomial regression, did not work well on some datasets, for example MKS. For ease of reference, the result for experiment 1 of MKS is shown here in figure 5-9 below:

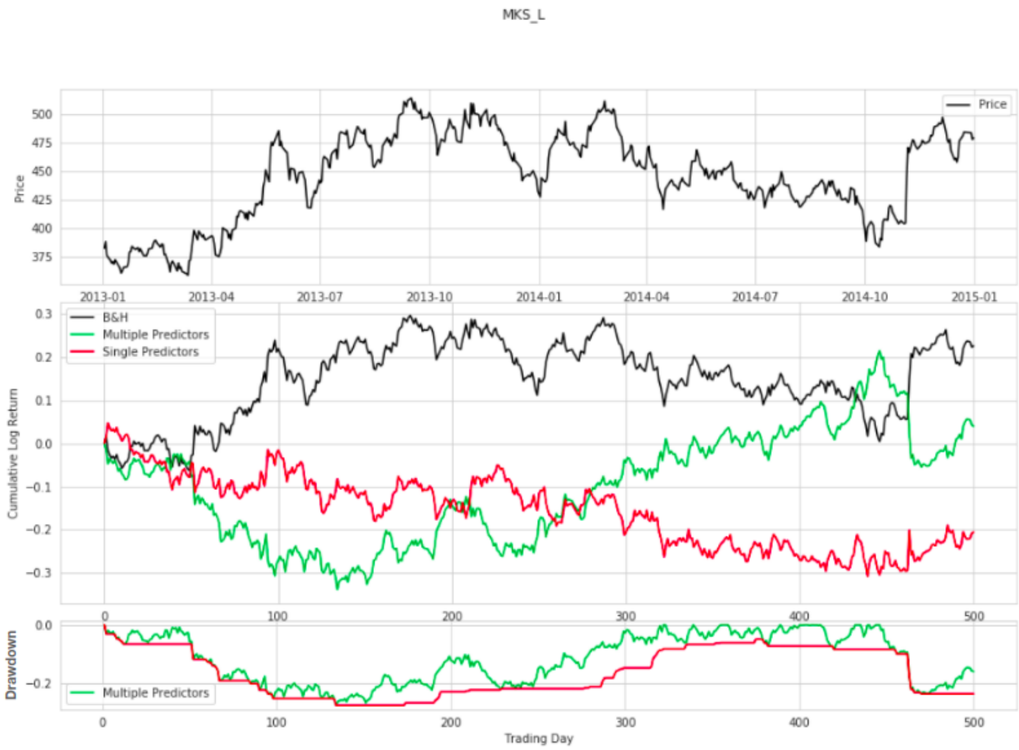


Figure 5-9: Result from experiment 1 on MKS

From figure 5-9, the middle graph (cumulative log return) shows that the result from only a single polynomial predictor (red graph) could not overcome the buy & hold (black graph) result for most of the testing period. Using multiple predictors (green graph) - two of the same machine learning predictors (polynomial regression) trained differently - performed better than the single predictor (red graph), but was still unable to overcome the buy & hold result for almost the whole period of testing. Therefore, this experiment will try to solve this problem by adding different types of machine learning algorithm. The bottom graph shows the drawdown. There are two graphs in the drawdown result, the green graph shows the drawdown value and the red graph shows the maximum drawdown.

5.1.2.2 Experiment Design

This experiment is set up similarly to Experiment 1 but adds different models of machine learning predictors. Therefore, we will not mention all the details of the experiment again. Readers can find the details of step 1 to step 3 from the previous experiment on page 118. As for steps 4 and 5, the training and testing steps, these are different in this experiment. In step 3, training, we added more machine learning algorithms as shown in figure 5-10.

Figure 5-10 shows that the training data is clustered into two groups, high and low volatility, as in the previous experiment. However, the machine learning predictors are not only polynomial regression. There are 5 more types of machine learning predictor. As previously, for each type of machine learning model, we created two predictors, one trained to predict using the low-volatility data, while another is trained for high-volatility data prediction. All of the predictors created for low-volatility (the predictors whose names end with _c0) will be trained with the same training data (cluster 0). On the other hand, the predictors created for high-volatility (the predictors whose names end with _c1) are going to be trained with another cluster of training data (cluster 1).

To perform this experiment, we add new predictors one at a time in order to observe the result of having more predictors. As for step 5, testing, the method for selecting the predictor from one model (for example, selecting between Poly_c0 and Poly_c1) is the same as mentioned in the previous step in 4 on page 121. However, we do not have a method to select the predictors across different algorithms yet. Therefore, the results from all algorithms will be shown in the form of average values in this experiment. If the averaging method does not work well, we will investigate a new method for selecting the predictors across different algorithms later in this chapter.

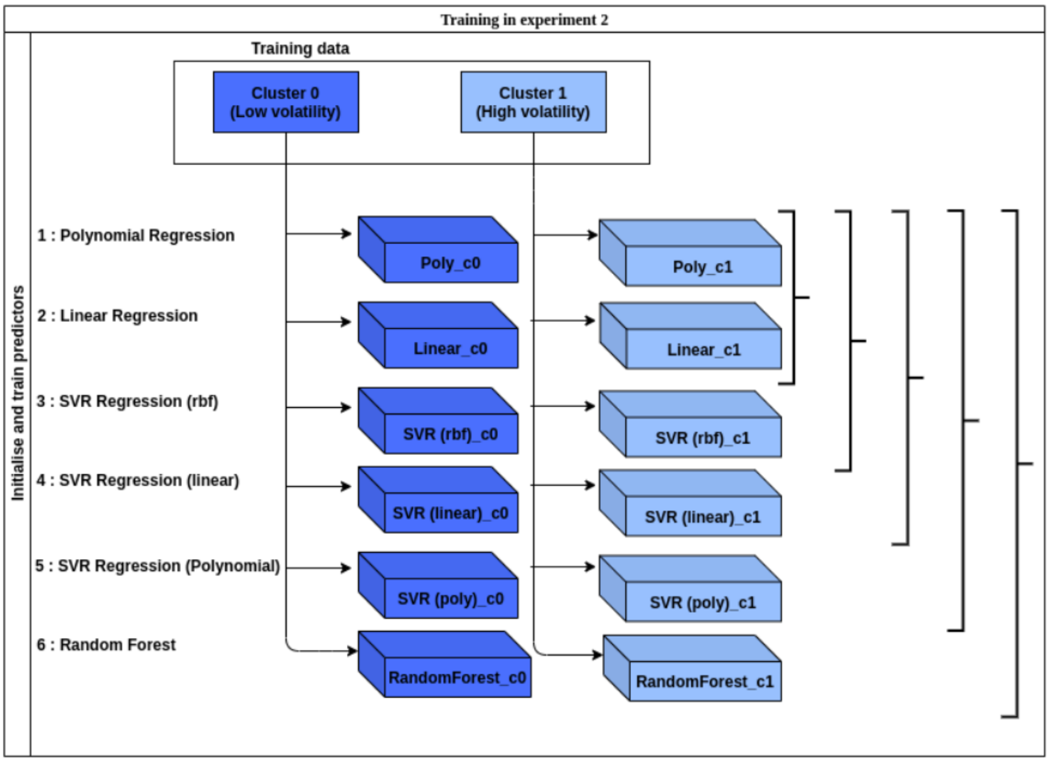


Figure 5-10: Training process in experiment 1.2

5.1.2.3 Experiment Results

This section begins with demonstrations of some of the results. Then, we will summarise all results at the end of the experiment.

- Examples of the results
  - **Results from individual algorithm** In this section, we present the benefits of using multiple machine learning algorithms. We start with polynomial regression (as in the previous experiment) and linear regression as an additional algorithm. Firstly, we would like to show the results from these algorithms individually in figure 5-11 and 5-12.

    Figures 5-11 and 5-12 show the results from using polynomial regression and linear regression on MKS, respectively. The top graph shows the cumulative returns, comparing between buy & hold (black line) and a regression algorithm (green line). Please note that each green graph is the result from using two predictors of the same algorithm. One predictor was trained on the low-

Figure 5-11: Result from using polynomial regression on MKS

volatility data while another was trained with high volatility, as mentioned in the previous experiment. The bottom graph shows the drawdown obtained from the regression. In this experiment, we run each machine learning algorithm separately. Then, we will bring them to work together in the next experiment. The bottom graph indicates drawdown. The drawdown values can be seen from the green graph, while the maximum drawdown is shown by the red graph.

As mentioned in the previous experiment, using polynomial regression did not work well for MKS (see figure 5-11). Therefore, we selected MKS to demonstrate this experiment. Figure 5-12 shows the result from changing polynomial regression to use linear regression on the same set of data from MKS.

Comparing the top graphs from these two figures, it can be seen that polynomial regression (see figure 5-11) did not work well on MKS for most of the period of testing, as the green line is lower than the black line (buy & hold). The linear regression worked much better, as can be seen from figure

Figure 5-12: Result from using linear regression on MKS

5-12. The cumulative return from using linear regression (green line) was better than the buy & hold for the most of the second half of the testing period. Intuitively, if our model is able to switch the predictor to a linear regression predictor for the second half of testing, the model's performance would surely be better.

Using polynomial regression resulted in gaining about 4% profit with around a 47% error rate. The error shows the percentage of incorrect predictions. Even though it ended with a positive return, the overall performance was not good throughout the testing period. Most of the time performance was below that of the buy & hold strategy. However, using linear regression provides 38% profit with 46% error. Even though the error rate is not much lower, the profit increases sharply. This means that even though the accuracy of using linear regression does not increase much (only about 1%), the performance is much better. The explanation for this is that linear regression is able to provide correct predictions on more important days (higher profit days), which is of paramount importance in this area. The return distribution from the correct and incorrect days is shown in Figure 5-13.

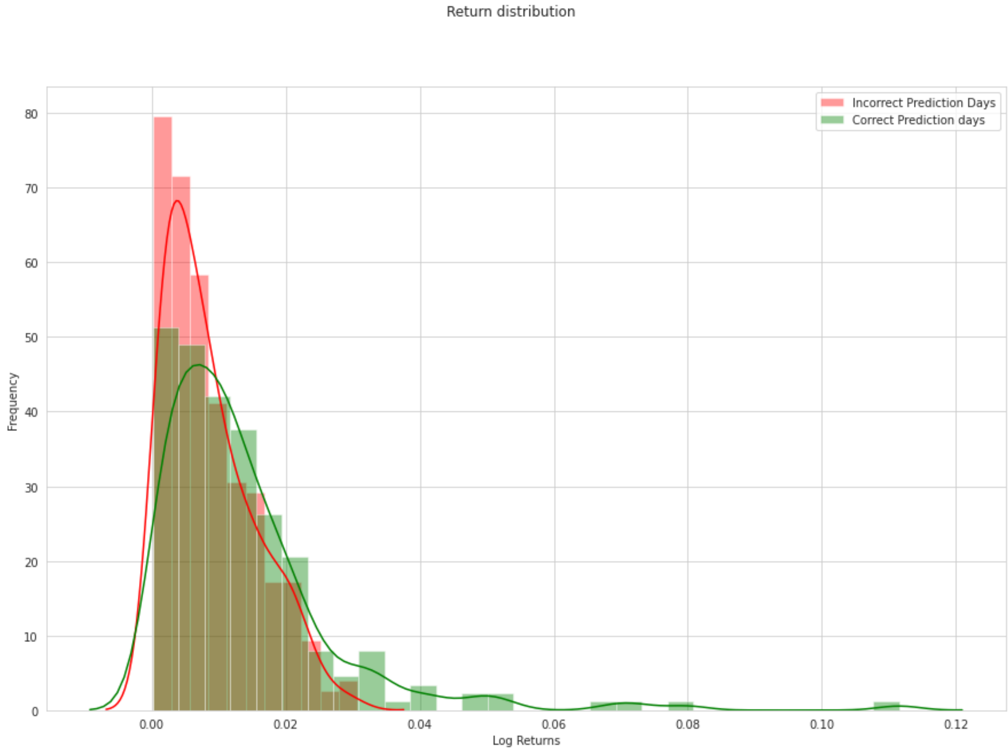


Figure 5-13: The comparison between return distribution from the correct and incorrect prediction days of MKS

There are two sets of return distribution in figure 5-13 represented by green and red colours. The green colour shows the return distribution on the days where linear regression predicted the correct direction, while the red colour shows the return distribution on the incorrect prediction days. The X-axis represents the daily logarithm returns. The Y-axis shows the frequency. From this figure, it can be seen that the first three red bars are higher than the green bars, which means that on those days profits were missed. However, the available profits on those days were very small. This could be benefit our system because when the profit is too small, it is not worth incurring the trading fees and risk that apply when taking a position. Therefore, it is desirable for the system to avoid these trades, thereby avoiding paying the trading fees for the unworthy positions. Looking at the green coloured distribution, representing the days when linear regression predicted the correct direction, it can be seen that once the return gets bigger, linear regression is still able to provide correct answers, as can be seen from the green bars on the right side of the graph. This is the reason why linear regression - which only reduces the error rate 1% - is able to increase the profit from 4 % to 38 %. Linear regression does not particularly increase the number of correct predictions, however it is able to avoid trading on very low profit days and correctly spots the days with higher profits. The average profit on the incorrect days (that linear regression missed) was only 0.87 %, while the average profit on the correct prediction days is 1.15 %. These daily profits might seem small, however, once profits accumulate and the avoidance of unnecessary trading fees are taken into account, this leads to improved results.

– **Results from combined algorithms** In the previous section, we showed the results from polynomial and linear regressions individually. We also mentioned the advantages we could have from being able to use these two algorithms together. In this section, we will compare between the results from those two algorithms and discuss the different ways we put them to work together.

  As mentioned at the beginning of this experiment, we do not yet have a method for selecting predictors across different algorithms yet. In this section, we use two basic methods to combine the results from multiple algorithms, averaging (realistic) and maximum (unrealistic).

  * **Averaging Selection** : This method will average the results from all algorithms involved. For example, from figure 5-10 on page 134, if the standard deviation shows that the low volatility group of predictors should be selected, all predictors in the low-volatility group (Poly_c0, Linear_c0 ... RandomForest_c0) will be equally weighted in the final result. On the other hand, if high-volatility is selected, all the results from predictors in high-volatility (Poly_c1, Linear_c1, ... , RandomForest_c1) will be averaged to create the final result.

  * **Best Selection (Maximum)** : This method is chosen only for comparison and to observe the range of possible results. It is not going to be added to our system since it deliberately selects the better predictor(s) by looking at the price of the next day (look-ahead bias included). Therefore, it is able pick the better predictor(s) at all times. Results from this method only show how well the result can be if we are able to select the better predictor(s) 100% of the time. However, this would

not happen when applied in the real world.

The visualisation of the results comparison is shown in figure 5-14.

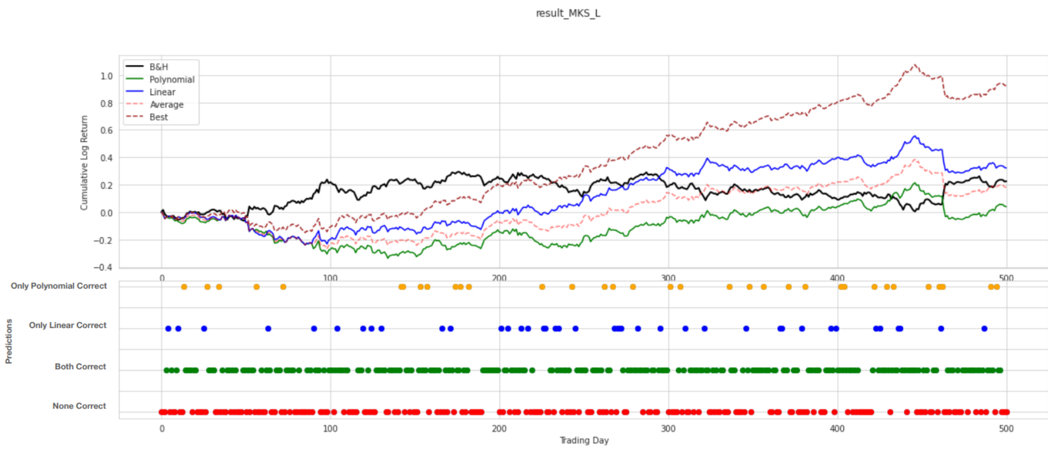


Figure 5-14: The comparison of results from buy & hold, polynomial and linear regressions, as well as averaging and best selection strategies on MKS

In figure 5-14, the top sub-figure compares the cumulative log return of the buy & hold strategy (black), polynomial regression (green) and linear regression (blue). The X and Y axis represent the trading day and cumulative return, respectively. As previously mentioned, linear regression seems to perform much better than polynomial regression, not only for the second half of testing period, but for most of the testing period. From this graph, it can be seen there are only a few days during the first 100 days during which polynomial regression appears to work better than linear regression.

There are two dashed lines in this sub-figure representing the results from the basic methods of combining values from the two machine learning algorithms (polynomial and linear regression). The average of the results is represented by the light pink graph. The result of this averaging method is located as expected in between the two results. Being the average of multiple algorithms, this method will not be as bad as the result of the worse algorithm or from the consistent selection of the worse algorithm, but it will not show the best possible result either. Therefore the brown dashed line shows the combined results if the best predictor is selected for each day. This is called the best selection method.

Please note that the only way to select the best predictor 100% of the time is to select it after the end of that day (or after knowing the actual closing price). However, this bias makes the result very unrealistic as it is impossible to trade after the end of the day and extremely difficult or almost impossible to select the best predictor in advance every time. We only show this result so that the reader has an idea of the best possible result these combined algorithms could achieve. The expected result we aim for will fall somewhere between the average and the best selection results. As for the predictor selection strategy, we will discuss and investigate this further in section 5.3 on page 159.

From figure 5-14, one might think that the only way to get the closest result to the best selection result (brown dashed line) is to follow the better algorithm (linear regression) for the whole time. In fact, this is not true, as can be seen from the bottom sub-figure. This sub-figure shows the correct predictor or the best predictor which could be selected at each time in order to get the best result (brown dashed line). The X axis shows the trading days corresponding to the top sub-figure. The Y axis represents the choice of predictor(s) with the correct answer at the time. There are 4 lines of points in this sub-figure. When only the polynomial predictor provides the correct answer, points show on the top line in the yellow colour. On the other hand, if only the linear predictor is correct, points are shown in blue in the second line. In cases where both predictors are correct, points are shown in green in the third line. On the other hand, if neither of the predictors are correct, points are shown in red colour in the bottom line.

There are only 34 days in total where only the polynomial regression predictor needed to be correctly selected and linear regression did not, as shown by the orange points. Therefore, this is not the same as the accuracy of polynomial regression shown in table 5.1 on page 129. Meanwhile, there are about 40 times in total where only the linear regression predictor was needed (shown in the blue points). The rest of the time it did not matter which predictor was selected. There are 232 days on which both of them provided correct predictions, and 195 days where neither was correct, represented by the green and red points respectively. In these two cases, it does not matter which predictor is used because the result is not going to change. Our next task is to investigate effective methods of selecting a suitable predictor(s) among different algorithms. For example, in this case we needed to correctly select the polynomial predictor on 34 days and the linear predictor on 40 days.

The accuracy we will discuss in terms of the error rate. In this experiment, an error happens on a day when the predictor(s) cannot provide a correct prediction, for example predicting the share price will increase on days when the price actually decreases. The error rate of using only polynomial regression is 46.8%, which means that 46.8 percent of the testing period (or 235 days out of 501 days) the polynomial regression cannot provide the correct predictor. As for linear regression, the error rate reduces to 45.6% (or 229 days out of 501 days). However, when using these two predictors together, the error rate decreases to 38.8% (or about 195 days out of 501 days) which is much lower than using those predictors individually. As expected, more predictors means a higher probability of getting the correct answer. If one fails, the other might succeed, especially when they were trained differently because they will provide better predictions for different situations.

So far, we have explained how we have done this experiment and showed visualisations of the results using only two machine learning algorithms for clarity. In practice, we have performed further iterations of this experiment, adding more algorithms one-by-one. The summarised result of using various numbers of predictors for the stock MKS can be seen in figure 5-15.

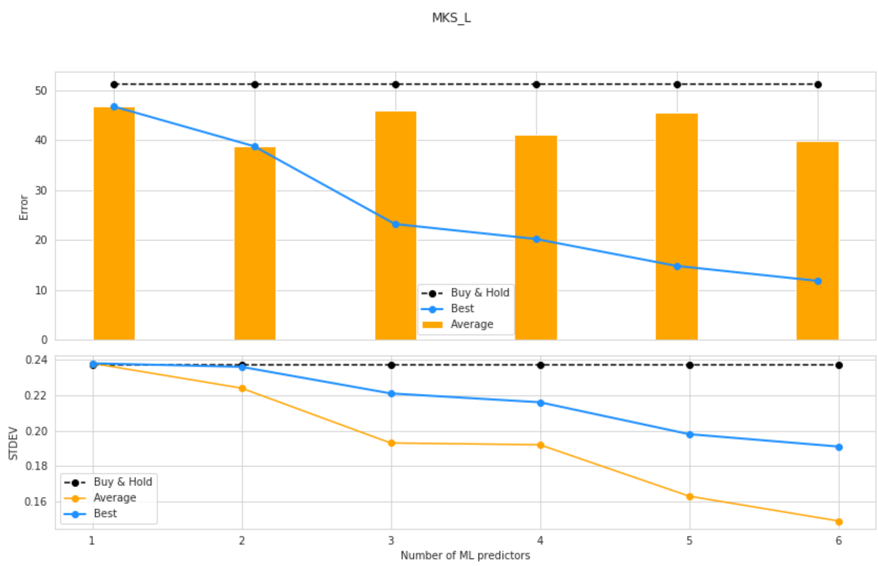


Figure 5-15: Result from experiment 1.2 on MKS

Figure 5-15 shows a summary of the results of using different numbers of predictors for MKS stock, adding one predictor at a time. The top sub-figure shows the error from the experiments, while the bottom sub-figure illustrates the standard deviation. These two sub-figures share the x axis, which shows the number of machine algorithms used in each experiment.

The top sub-figure shows the error rate of using combined polynomial and linear regression in different ways. The orange bars indicate rates of error from the averaging model, while the blue graph shows the error from using the best selection model. As can be seen, the error rate decreases when using more predictor (blue line). Starting with one algorithm, the error rate is 46.8% and reduces to 38.8% when using two algorithms. This error decreases further to only 11.8% when we increase the number of algorithms to six. This means that, when using multiple predictors created from six different algorithms working together, there are only 60 days out of 501 days when no predictor is able to provide a correct prediction. This is a notable improvement. However, we still have a lot more work to do to get to this point. In order to get to closer to this great result, we need to have the right strategy to select the correct predictor(s) each time. This issue is going to be discussed later in section 5.3 on page 159.

On the other hand, the result from the averaging model, represented by the blue bar plot, does not show the same trend. As can be seen, the size of the bars does not get smaller when adding more predictors. The error seems to fluctuate since the average value suffers from the worst predictors. Even though the error rate of the average model does not decrease significantly when more predictors are added, it is still lower than the error rate of the buy & hold strategy, represented by the black dash-graph on the top sub-figure. The error rate of buy & hold is 51.2%, which is significantly higher than the error rate from both combined algorithms.

The bottom sub-graph compares the standard deviation of buy & hold and the two combined algorithm models. The back dashed line shows the standard deviation of the buy & hold strategy, which is higher than that of the averaging model and the best selection models when using two or more algorithms. As can be seen, the standard deviation of both combined models decreases as more algorithms are added. The lowest standard deviation comes from the averaging model, which makes sense. Averaged values result in lower variance but cannot guarantee better predictions. As for the best selection model, the standard deviation values are higher than those from the averaging model. However, they are still significantly lower than the buy & hold strategy, which is as we expected. From these results, the best standard deviation is a trade off between the best selection and averaging models.

All the results mentioned above come from one dataset, MKS. In practice, we have experimented with the datasets of many other stocks. The rest of the results will be shown in the following section.

- All Results

This section discusses the results of the previous experiment conducted on 11 datasets. The list of these stocks can be seen in table 6.1 on page 233. Instead of showing the result from each dataset separately, the results of all stocks are averaged and shown in figure 5-16.

Figure 5-16 is composed of three sub-figures comparing the results from the best selection, averaging and buy & hold strategies. The X-axis shows the number of machine learning algorithms while the Y axis of these three sub-figures shows the error rate, standard deviation and Sharpe Ratio, respectively. The top figure illustrates the distribution of the errors. The blue colour relates to errors from the best selection model, which are more skewed to the right (or to the bottom in this figure) with the increasing number of machine learning predictors added to the model. This means that fewer errors are possible when having more algorithms than when using fewer. The average error value of each violin plot is shown as a dash in the middle. It can be seen clearly from the average values that the average errors are smaller when a higher number of machine learning algorithms is involved. The average error with a single algorithm is 46.87%, which decreases to 40.76% and 28.05% when increasing the number of algorithms to two and three, respectively. Finally, when using six algorithms, the average error ends up being only 15.43%. These blue graphs clearly show that the error rate decreases when adding more machine learning algorithms.

However, the results from the averaging model (in orange colour) show the opposite. That is, the error rate does not decrease clearly with additional machine learning algorithms. When the number of machine learning algorithms is one (comprising low and high volatility predictors), the results from the best selection and the averaging method are identical as the system will only pick one predictor at a time (either a predictor for high or low volatility). When increas-

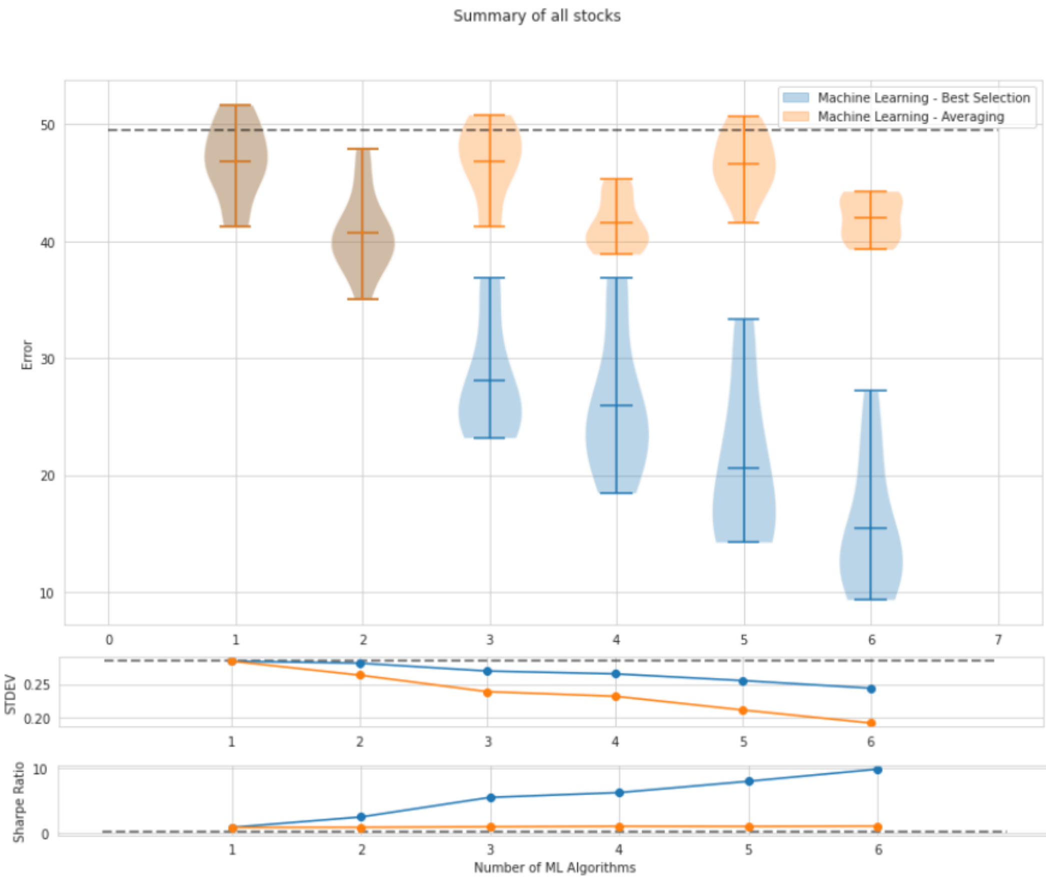


Figure 5-16: Result from experiment 1.2 averaging from all 11 datasets

ing the number of machine learning algorithms to 2 (4 predictors), the errors are smaller and similar to the error for the best selection method. However, when the number of algorithms increases to three, four, five and six, instead of seeing the expected decrease in errors, the errors seemed to fluctuate. Even the distribution changed, as there is no downward trend here. The errors from this method do not show better results, but are still lower than the error rate of the buy & hold strategy (black dashed line).

We have noticed that in the best selection model, the error rate is lower when the number of machine learning algorithms is even. We attribute this effect to the averaging of the final predicted direction (-1, 1). Therefore, the averaging here functions similarly to majority voting. When the number of algorithms is even, there is a higher probability of getting the answer 0, meaning the system will not do anything. This means when the predictors are not very sure of their answers, holding any positions or do nothing can result in better performance. This point will be taken in to account subsequently. If the price is stable, the system will be able to provide a signal for holding.

From the comparison of blue and orange violin plots in the top sub-figure, it can be concluded that the error can be lower when more machine learning algorithms are used. However, this can happen only when the correct predictors are selected, as the error rate reduces only for the best selection model. Therefore, we will run further experiments to find an effective method of selecting predictors.

The middle sub-figure shows the average standard deviation of the returns at different numbers of machine learning algorithms. The blue and orange graphs represent the standard deviation of the best selection and averaging methods, respectively. These results turned out as we expected; having more algorithms decreases variance. For the best selection (blue graph), the decrease in the standard deviation is very clear as the best predictor(s) is selected. The standard deviation from the averaging technique also gave expected results, being lower than the best selection method since it averages all predictors. The standard deviation from both methods were lower than the standard deviation of the benchmark buy & hold strategy (black dashed line).

The average Sharpe Ratio at different numbers of algorithms is shown in the bottom sub-figure. The blue graph shows the Sharpe Ratio when the system is able to select the best predictors at all times. The ratio keeps increasing but it must be remembered that these are the most optimistic results, which can only happen with the benefit of hindsight. Therefore, we are not hoping to achieve this very high Sharpe Ratio of 10, but one that falls between this and the Sharpe Ratio from averaging (orange graph). This was still relatively low (raging from 0.9 to 1.2), showing that having more machine learning algorithms (more predictors) will not be helpful without an effective strategy to select the right predictors for the right time. Nonetheless, all Sharpe Ratio values were higher than the benchmark buy & hold strategy (black dashed line) ratio.

From the results, we conclude that increasing the number of machine learning algorithms can improve the performance of the system. However, it only works well when suitable predictors are selected at a time. We can see the outstanding results from the best selection method, but as we have mentioned, it is practically impossible to get these results. Therefore, the results we expect to fall somewhere between the best result and the averaging method. Our focus going forward is to find an effective method to select the right predictor(s) in order to get closer to the best selection method's result.

5.1.2.4 Discussion

This section started with the premise established by the previous experiment, that multiple predictors could outperform a single predictor. However, if both predictors use the same machine learning algorithm, performance may be reduced if this is not the best algorithm for the data. This problem could be solved by adding more machine learning algorithms to the model. Adding more machine learning algorithms means having more different types of predictors and we can expect to predict better since each predictor specialises in a different characteristic of the data (as they were trained differently). Each added machine learning algorithm has two predictors, trained for high and low volatilities. All low-volatility predictors were trained with the same datasets (cluster 0). The rest of the data (cluster 1) was used to training all high-volatility predictors. We have experimented with six different machine learning algorithms in total. We increased the number of algorithms one at a time so we could see the effect of having more types of predictors.

In the absence of a specific method to select the best predictor, we opted during these experiments to use the average result from all predictors. Another method we used was best selection. The best selection method simply selects the best predictor by looking at the next day's prices. In reality, we would not be able to see this information. Therefore, the best selection method was included only to show the best results that could be achieved using these six algorithms. To compare the results, we used the buy & hold strategy as a benchmark.

The results show that using the average value is not a useful method as the results suffer from the bad predictors. However, from the best selection method results we saw that using multiple algorithms provides scope for significant improvement of trading performance. Even though this method cannot be used since we cannot know prices in advance, we conclude that performance can be improved with an effective method to select predictor(s). This leads us to the next experiment, which focuses on different methods to select the best predictors.

5.2 Experiment 2: Bias Removal and Predictor Selection with a Statistical Approach

This research aims to use multiple machine learning predictors to build a trading system, as we have shown that using multiple predictors can potentially improve trading performance. The premise of this system is that each predictor will specialise in a different type of data, which means that in dynamic and changing financial markets, one can consistently profit since when performance drops in one predictor, another will start performing better. Thus far, the criteria used to select the correct predictors (within the same algorithm) has included the look-ahead bias, as the experiments only intended to show the validity of our hypothesis that multiple predictors perform better, as mentioned in section 4 on page 121. This section aims to remove the bias, necessitating a method to consistently select good predictors for unseen times in the future. The first approach we have chosen is called statistical selection. Then, in the following experiments, other approaches will be considered.

Please note that we will not address the selection of predictors across different algorithms in this experiment. We will continue using the averaging method to select the predictors, as in experiment 1.2.

5.2.1 Statistical Selection Method

This is the first and easiest method we came across when starting to investigate predictor selection methods. The reason we classified this method as the easiest one is because it is very similar to the selection method in the first experiment, mentioned in section 4 on page 121. This method selects the predictor by considering the mean and standard deviation (the same as the method which is used in experiment 1). The only difference between the selection process in this experiment and the one in the previous experiments is we take out the look-ahead bias in this experiment. The comparison of this difference can be seen in figure 5-17.

Figure 5-17 shows the data for the testing period which started at the beginning of 2013. There are two predictors in this experiment (for high and low volatility). In order to predict the direction of the price each day, the system will have to decide which predictor to use. The selected predictor is supposed to be the most suitable predictor for the data in that region. To perform this, we use k-means clustering. The k-means clustering will classify the data into either cluster 0 or 1, based on the shortest

Date	Close	Log Rets	Rets_mean	Rets_std
2013-01-02	342.750000	NaN	NaN	NaN
2013-01-03	350.000000	0.020932	NaN	NaN
2013-01-04	349.450012	-0.001573	NaN	NaN
2013-01-07	349.299988	-0.000429	NaN	NaN
2013-01-08	351.500000	0.006279	NaN	NaN
2013-01-09	349.149994	-0.006708	0.003700	0.010686
2013-01-10	355.399994	0.017742	0.003062	0.009420
2013-01-11	353.200012	-0.006209	0.002135	0.010188
2013-01-14	351.200012	-0.005679	0.001085	0.010772
2013-01-15	349.600006	-0.004566	-0.001084	0.010554

(a) Testing in experiment 1 with look-ahead included

Date	Close	Rets	Rets_mean_shift	Rets_std_shift
2013-01-02	342.750000	NaN	NaN	NaN
2013-01-03	350.000000	0.020932	NaN	NaN
2013-01-04	349.450012	-0.001573	NaN	NaN
2013-01-07	349.299988	-0.000429	NaN	NaN
2013-01-08	351.500000	0.006279	NaN	NaN
2013-01-09	349.149994	-0.006708	NaN	NaN
2013-01-10	355.399994	0.017742	0.003700	0.010686
2013-01-11	353.200012	-0.006209	0.003062	0.009420
2013-01-14	351.200012	-0.005679	0.002135	0.010188
2013-01-15	349.600006	-0.004566	0.001085	0.010772

(b) Testing in experiment2 with look-ahead bias removal

Figure 5-17: The comparison between testing data in experiment 1 and 2

distance between the data and the centre of the group. If the data is classified as cluster 0, the system will select the low-volatility predictor, otherwise the other predictor is selected.

As for the difference between these two sub-figures, Sub-figure (a) shows that in order to select the correct predictor to predict the price direction on the 9^{th} of January, it takes the values of the mean and standard deviation of the return (0.003700, 0.010686) which were calculated by including the return on that day (the red rectangle in column Rets). This situation is impossible since we will never know the closing price of that day until the end of the day, when we can no longer trade. This is an example of how the look-ahead bias was included in the first experiment to ensure that selecting the correct predictor among multiple predictors did improve the results. More detail can be seen in section4 on page 121.

Sub-figure (b) shows the selection method in this experiment (with bias removed). It can be seen that we have removed the bias by taking the mean and standard deviation only up to the day before. Therefore, in order to predict the price direction for the 10^{th} of January, making the decision to buy or sell after market close the previous day. Only the returns up to the 9^{th} are included. This is our first predictor selection strategy, called *Statistical Selection Method*. Removing the bias this way makes the predictor selection realistic.

5.2.1.1 Experiment Design

This experiment is designed to investigate whether a multiple-predictor and multiple-algorithm trading system continues to work well after removing the look-ahead bias. The design and result sections will be divided into two sections as follows.

- **Section 1** shows the results after removing bias from the multiple-predictor experiment. Therefore, the results from section 1 should be compared with the results from experiment 1.1.

- **Section 2** shows the results after removing bias from the multiple-algorithm experiment. Therefore, the results from section 2 should be compared with the results from experiment 1.2.

As mentioned above, this experiment uses the same predictor selection method as in Experiment 1: the mean and standard deviation of the returns. The purpose of this experiment is to compare the results after removing the bias with the initial results from Experiment 1 which included the bias. If the results turn out well, it means that 1) multiple predictors and multiple algorithms continue working well even without the benefit of looking ahead, and 2) statistical predictor selection can be used in order to switch between predictors over time.

The design of this experiment will be the same as in Experiment 1, except for the bias removal. Section 1 of this experiment is designed the same way with Experiment 1.1, while Section 2 follows the design of Experiment 1.2. Therefore, the training processes of Sections 1 and 2 can be seen in figure 5-3 on page 121 and 5-10 on page 134, respectively.

5.2.1.2 Experiment Results

The results in this experiment will be separated into two sections. Section 1 presents the results from only a single algorithm (but multiple predictors). Therefore, these results are meant to be compared with the results from Experiment 1.1 (5.1.1 on page 118). Section 2 illustrates the results of using multiple machine learning algorithms. Therefore, the results from this section will be compared with those from Experiment 1.2 (5.1.2 on page 131)

- **Section 1: Results from multiple predictors with bias removal**

  The results from this section are to be compared with the results from Experiment 1.1 since they have the same experimental design (using only one algorithm but two predictors). The only difference is the removal of the look-ahead bias from the testing process so we can compare performance with and without the bias. Before presenting all the results, examples will be shown in order to illustrate some of the differences between the results between before and after removing the look-ahead bias. We have selected two stocks, CARR and TSCO, which are the same stocks whose results were presented in Experiment 1. The results from Experiment 1 (including look-ahead bias) of CARR and TSCO can be seen in Figure 5-5 on page 123 and 5-6 on page 124, respectively.

  The results of CARR and TSCO were very good in Experiment 1. However, when we re-run the experiment with the bias removed, the results were worse, as can be seen in Figure 5-18 and Figure 5-19.

  Figures 5-18 and 5-19 show the comparison of the results of the new method - Statistical Selection - and the other strategies (Buy & Hold benchmark, multiple predictor with bias included, and single predictors). Each result is composed of three sub-graphs. The top and bottom sub-graphs represent the daily closing price and the drawdown of the Statistical Selection method. The middle graph is the most important because it compares the cumulative return of all the strategies mentioned earlier. The black line is the closing price. The red and green dashed lines represent the results from the biased single and multiple predictors (as in experiment 1). As before, the bottom sub-figure shows the drawdown and the maximum drawdown (red line).

  From CARR's results, the Statistical Selection method's performance - shown by the blue line - was worse than that of the original multiple predictor from Experiment 1 (green line). Some decrease in performance was to be expected

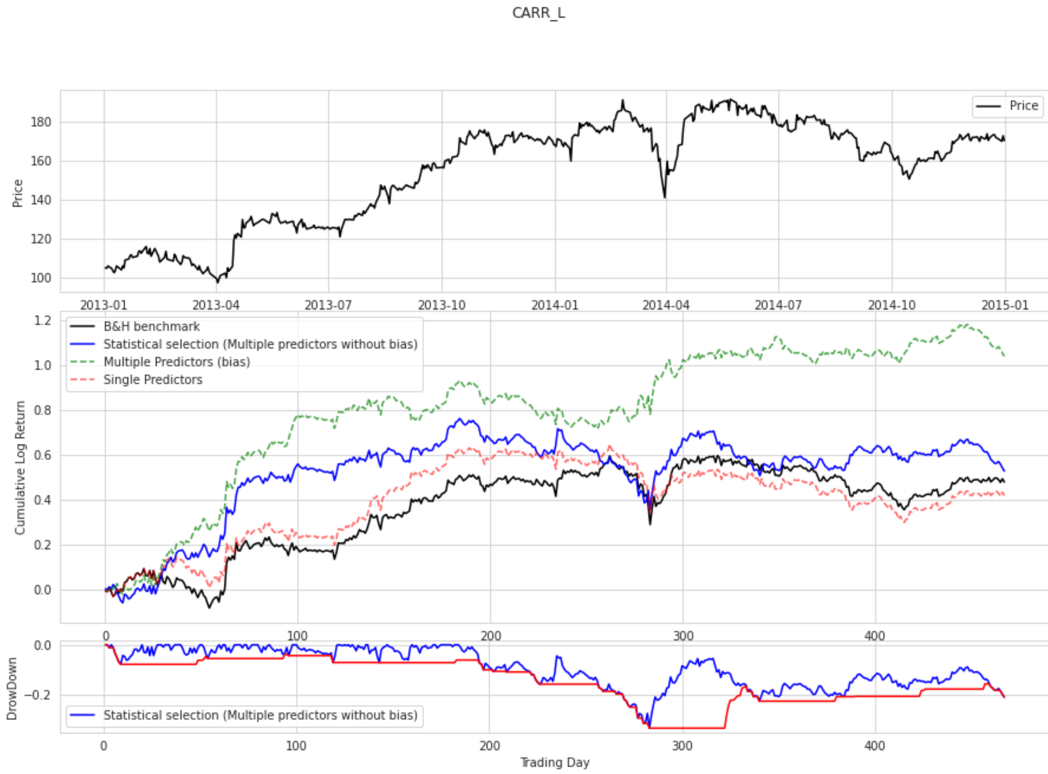


Figure 5-18: Result from experiment 2 on CARR

given removal of the look-ahead bias. The Statistical Selection method ended up with 68.83 % profit while the multiple predictors with bias obtained an unrealistic result of 183.38 % profit. The Statistical Selection method saw the Sharpe Ratio decrease to 0.84 from 1.78 with bias. It can be seen clearly that for this data, when using Statistical Selection (removing bias) to select a suitable predictor, the results are not great. The system could not change to the more suitable predictor as quickly as needed. For example, after the beginning of April 2014, it was supposed to use the high volatility predictor, but it did not change quickly enough, resulting in a big loss (-33.23 %) during this testing period. It is overfitted to a specific predictor (or the specific group of data which that predictor was trained on). However, this result still outperformed the buy & hold strategy and single predictor strategy.

The result from TSCO in Figure 5-19 is worse. The cumulative return of the Statistical Selection method is worse than expected in comparison to the multiple predictors (with bias). From this result, it is plausible that the Statistical

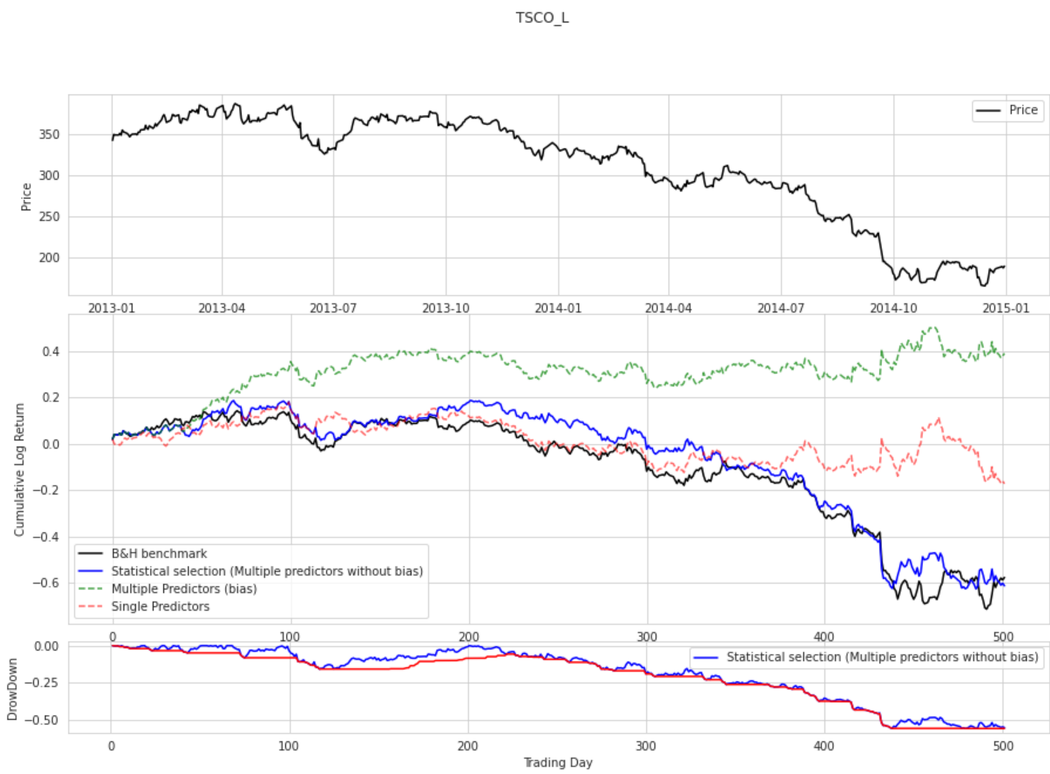


Figure 5-19: Result from experiment 2 on TSCO

Selection method (using the standard deviation of returns for predictor selection) is unsuitable for this data. Although, despite performing worse than the multiple predictor model, the Statistical Selection method seems to be competitive among the rest of the strategies. The exception is after the end of the fist half of the year 2014, where it performed worse than the single predictor but still better than the benchmark.

Results from the two datasets above show that using only a statistical value (standard deviation) does not work very well. It appears the Statistical Selection method cannot select a suitable predictor in advance. However, these two results (two datasets) might not be enough to draw this conclusion. Therefore, we experimented further with the rest of the stocks. All results are shown in Table 5.2.

Table 5.2: The comparison between Buy & Hold, Single predictor, Multiple predictor (with bias) and Statistical Selection (Multiple predictor with bias removal) methods

Data & Model	Profit (%)	Sharpe Ratio	Max DD (%)	Accuracy (%)	Stdev
CARR					
Buy & Hold	62.38	0.77	-26.32	52.89	0.302
Single Predictor	52.92	0.66	-29.33	52.46	0.302
Multiple Predictors	183.38	1.78	-19.05	54.39	0.300
Statistical Selection	74.01	0.89	-33.23	49.89	0.302
COST					
Buy & Hold	11.57	0.09	-22.26	51.26	0.296
Single Predictor	81.97	0.97	-15.72	51.46	0.295
Multiple Predictors	-14.54	-0.38	-38.90	49.37	0.296
Statistical Selection	-10.94	-0.31	-25.88	49.79	0.296
D4T4					
Buy & Hold	39.10	1.19	-37.76	51.75	0.586
Single Predictor	-27.22	-1.25	-46.08	49.12	0.586
Multiple Predictors	70.71	1.98	-17.22	58.77	0.583
Statistical Selection	105.02	2.69	-31.57	57.89	0.579
FERG					
Buy & Hold	18.59	0.26	-16.09	49.10	0.211
Single Predictor	20.85	0.31	-32.67	48.70	0.211
Multiple Predictors	26.61	0.42	-13.84	50.90	0.211
Statistical Selection	-43.56	-1.51	-45.74	48.7	0.210
GOOG					
Buy & Hold	45.71	0.74	-18.72	50.10	0.213
Single Predictor	48.67	0.79	-15.49	50.50	0.213
Multiple Predictors	-8.45	-0.35	-24.30	48.31	0.213
Statistical Selection	5.12	0.02	-24.23	47.91	0.213
MACF					
Buy & Hold	26.96	0.65	-28.06	49.81	0.312
Single Predictor	-1.83	-0.15	-36.28	52.92	0.313
Multiple Predictors	114.62	2.33	-19.23	58.37	0.309
Statistical Selection	56.54	1.31	-18.51	55.64	0.312

Continued on next page

Table 5.2 – *Continued from previous page*

Data & Model	Profit (%)	Sharpe Ratio	Max DD (%)	Accuracy (%)	Stdev
MCD					
Buy & Hold	3.97	0.08	-14.61	54.6	0.124
Single Predictor	1.78	0.17	-14.61	54.2	0.124
Multiple Predictors	23.87	0.63	-10.33	54.2	0.124
Statistical Selection	8.66	0.1	-12.20	53.6	0.124
MKS					
Buy & Hold	25.11	0.35	-25.24	48.8	0.237
Single Predictor	-18.61	-0.56	-29.86	48.6	0.237
Multiple Predictors	4.13	0.04	-27.50	53.2	0.238
Statistical Selection	-19.13	-0.58	-27.35	50.8	0.237
OXIG					
Buy & Hold	-15.00	-0.33	-46.91	48.07	0.346
Single Predictor	21.61	0.20	-27.29	50.10	0.346
Multiple Predictors	357.01	2.19	-16.29	52.55	0.343
Statistical Selection	13.92	0.11	-37.62	49.49	0.346
SXS					
Buy & Hold	0.67	0.10	-35.85	51.70	0.263
Single Predictor	-45.88	-1.30	-52.22	47.29	0.262
Multiple Predictors	67.36	0.88	-22.65	53.91	0.262
Statistical Selection	79.62	1.01	-17.98	53.51	0.262
TSCO					
Buy & Hold	-44.86	-1.35	-57.50	46.91	0.244
Single Predictor	-17.42	-0.52	-29.66	46.91	0.244
Multiple Predictors	44.84	0.64	-15.80	50.50	0.244
Statistical Selection	-51.88	-1.64	-59.82	45.71	0.243

Table 5.2 shows the results from our experiment performed on 11 stocks, as they were in the previous experiment. We also ran a paired T-test on the Sharpe ratios of these datasets to compare between the multiple and single-predictor (polynomial regression) strategies, this time with the intentional look-ahead bias removed. The p-value on this occasion was 0.59, which is not statistically significant. This is much higher than the critical value, so we accept the null hypothesis

that there is no statistical difference in the Sharpe ratios between the multiple and single predictor approaches. Our expectation for this experiment was for the results to turn out in one of three ways. Firstly, the results could be better than the benchmark and the single predictor, even though the multiple predictors would perform worse after removing the bias. This would mean that the statistical selection method works well overall. This result was met for the majority of the stocks, such as CARR, D4T4, MACF, MCD, OXIG and SXS.

Secondly, we hypothesised that the results of Statistical Selection for some stocks might be worse than the multiple predictor with bias and the benchmark. The stocks which fall into this group are FERG and TSCO. For these stocks, the results from multiple predictors in Experiment 1 (including the bias) were good, which meant that having multiple predictors could be helpful. However, the Statistical Selection method is unable to select or change the predictor effectively to deal with the fluctuations, leading to worse performance in this experiment. When analysing, we found that prices of the stocks in this group fluctuated highly. Therefore, the predictor selection method needs to be able to switch predictors quicker. Without the benefit of look-ahead bias, the Statistical Selection method predicts a continuation of the past, for example by selecting the low volatility predictor because the most recent period has seen low volatility. However, it is not the case in financial markets that this will continue, and a sudden and unexpected highly volatile market could harm performance. This problem will be brought up again when we investigate on the different methods of predictor selection (section 5.3.2 on page 167)

Thirdly, we expected that stocks results that used to be bad (for example MKS) would not be improved. These are the stocks for which we concluded the predictor could not deal with them. As this result came from using only one machine learning predictor (polynomial regression) which was unsuitable for the stock, it does not matter if the method for selecting the predictor has changed because polynomial regression itself could not perform well on this stock. The only way to improve the performance on such stocks is to add more useful predictors. The stock in this group is MKS, which we discussed in the previous experiment, section 5.1.2 on on page 5.1.2. Therefore, we did not expect a better result here. However, this result is expected to be better when more machine leaning algorithms are involved in the following section.

We have discussed the three groups of stocks that behaved as we expected. Now, we will discuss the unexpected results. These are stocks for which predictor selection methods performed poorly with and without look-ahead bias included. There are two stocks in this group: COST and GOOG. The reason for these results could be either that the current predictor selection method, Statistical Selection, is ineffective, or that the idea of having multiple predictors does not work with these datasets. To deal with this problem, we will investigate further predictor selection strategies in section 5.3 on page 159.

The results from table 5.2 are obtained on a single-stock training basis. This means that the predictors were trained on a single stock's historical data and tested on data from the same stock but a different test period. However, an alternative way to train predictors would be a multiple stock basis, with the trained predictors then applied to individual test set for each stock. Therefore, we performed multiple stock training on five of our stocks to see the difference in results. Details of this experiment can be seen in appendix D on page 333. We are not including it here since the results did not affect the design of our trading model and we want the development of our model in this chapter to be easy to follow for the reader.

- **Section 2: Results from multiple machine learning algorithm with bias removal**

This section aims to compare the results before and after removal of the look-ahead bias in experiment 1.2 5.1.2 on page 131. The results are shown in Figure 5-16 on page 144. In this experiment, the conditions are the same as in experiment 1.2, except that the bias is removed. The results after removing the bias can be seen in figure 5-20. Please note that the results shown in this section are averaged from all 11 datasets.

Figure 5-20 comprises three sub-figures, representing the results from the best selection model, the averaging model and the buy & hold strategy. The X-axis shows the number of machine learning algorithms while the Y axis of the sub-figures represent the error, standard deviation and Sharpe ratio. The top figure illustrates the distribution of error. The blue violin plots relate to errors from the best selection model. The average error from the best selection decreases over the number of machine learning algorithms involved in the system, from about 48% to 18 %. As in experiment 1.2, the orange colour shows errors from the averaging model, which do not decrease with an increasing number of machine learning algorithms. This is not unexpected as the results are similar to those from experiment 1.2. Therefore, it confirms the same conclusion; that the error

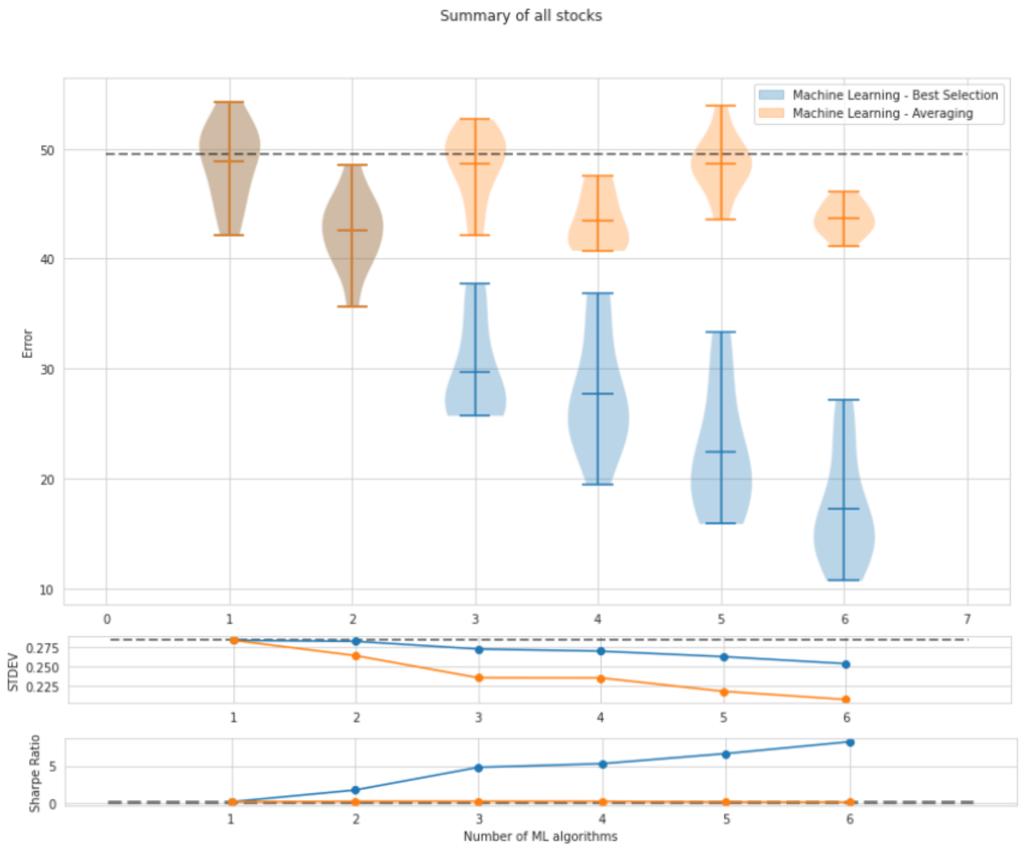


Figure 5-20: Result from experiment 2 averaging from 11 datasets

is lower when more machine learning algorithms are included. However, this can happen only when the correct predictors are selected. Therefore, we run further experiments to find an effective method of selecting predictors.

The middle sub-figure shows the average standard deviation of the returns at different numbers of machine learning algorithms. The blue and orange graphs represent the standard deviation from the best selection and averaging methods, respectively. The standard deviation steadily decreases for the best, fluctuating a little but the overall trend is still downward. Both of the methods provide a smaller standard deviation than the buy & hold strategy (black dashed line)

Comparisons of the Sharpe Ratio are shown in the bottom sub graph. The blue and orange graphs represent the Sharpe Ratio from the best selection and the averaging methods, respectively. As expected, the Sharpe Ratio from the best selection method increases clearly. It follows the same trend as that seen in experiment 1.2 (before removing the bias), but the values are lower. Meanwhile, the Sharpe Ratio from the averaging method (orange graph) is still relatively low (raging between 0.1 to 1.4), which is much lower than the best selection method but still greater than the benchmark.

It can be seen clearly that the results before removing the bias (5-16) and after removing the bias(5-20) behave in the same way. The results from the best selection method are excellent. However, as we have mentioned, these optimistic results will only happen if the most suitable predictors are selected at all times, which is extremely difficult or almost impossible. On the other hand, the results from the averaging method did not demonstrate the ability to select suitable predictors. This means that the conclusion we made at the end of experiment 1, that increasing the number of machine learning algorithms can improve the performance of the system, remains correct but only holds when we have a good strategy to select suitable predictors. Therefore, the next section aims to investigate predictor selection strategies. The expected performance of this strategy falls somewhere between the results of the best selection and averaging methods.

5.2.1.3 Discussion

In this experiment, we separate the results into two sections based on our objectives. The objective of section 1 was to investigate a method for selecting predictors, called Statistical Selection. The objective of section 2 is to investigate a method to combine the predictions of multiple predictors.

In section 1, we investigated the possibility of using Statistical Selection to select suitable predictors. The Statistical Selection method is a simple way to select predictors since it make the choice of predictor(s) based on the standard deviation of the data. This method was also used in Experiment 1 when trying to establish our concept. However, the important difference between the statistical selection in Experiment 1 and in this experiment is the look-ahead bias. In Experiment 1, we deliberately included the bias only to gauge the best potential performance should the most suitable predictor always be chosen, but the bias has been removed in this experiment to make the system realistic.

The results from Section 1 were compared with the results from Experiment 1.1. Therefore, each stock was only tested with a single machine learning algorithm, a Polynomial Regressor. There were two predictors, specialising in low and high volatility. Both predictors are polynomial regressors which were created identically but trained with different clusters of training data. The results show that after removing look-ahead bias, the majority of stocks performed as expected; worse than before removing the bias but still better than the buy & hold benchmark. However, there are a few stocks for which multiple predictors still performed badly, which we attribute to not having enough machine learning algorithms to deal with the different characteristics of datasets. This leads us to the investigation in Section 2.

In order to solve the problem of not having enough machine learning predictors, Section 2 adds more machine learning algorithms into the system. When the number of machine learning algorithms increases, the number of predictors also goes up by twice as much. For example, when using two machine learning algorithms, there will be 4 predictors (each algorithm will have two predictors, one each for low and high volatility). In this experiment, the number of machine learning algorithms increases gradually from one to six. The results show that having more machine learning algorithms can increase trading performance, but only if we have an effective method to select suitable predictors. However, up to this point, we have not found such a selection method. Therefore, in the next section, we will investigate a few predictor selection strategies in order to find the best method to incorporate into our trading system.

5.3 Experiment 3: Prediction Selection

In this experiment, we investigate multiple strategies for predictor selection. We would like to find a more effective way to select the correct predictors than using the statistical selection or averaging method used in the previous experiments. There are two predictor selection strategies in this experiment, which are Upper Confidence Bound (UCB) and scoring system. Some examples of using these methods individually will be shown, followed by a comparison of the results between all of the predictor selection strategies, including the averaging method from the previous experiment.

5.3.1 Upper Confidence Bound

This section applied the Upper Bound Confidence selection method to select the predictor, as explained in Chapter 3 section 3.2.5 on page 76. The idea of using UCB comes from its application to general problems in recommendation or advertisement systems. The main problems of these systems are the non-stationary environment and the large-state spaces. In advertisement, there are many choices of advertisement pictures and the company wants to select the most loved picture to put on their main page. However, the advertising team cannot spend so much time and money to run tests on the massive potential customer base. So, using the UCB algorithm, the team start randomly showing these pictures to their customers (showing one picture to one customer). Once every picture has been shown, the algorithm will start selecting the next picture by analysing the obtained responses. Using this process, the selected picture will be changed over time.

As there are many predictors in our system which need to be managed, we considered the idea of UCB to manage the selection process. In our analogy with the advertising problem, our predictors function the same way as the advertising pictures. The system needs to start with exploration by allowing every predictor to perform a prediction. After every predictor has made their prediction, the system then starts to select predictors based on their past performance. This means that the system is now exploiting instead of exploring. However, as the surrounding environment changes, the system cannot keep exploiting forever and must switch to exploration occasionally. This exploitation-exploration process is controlled by the UCB algorithm. The details of our experimental setup can be seen in the next section.

5.3.1.1 Experimental Design

The main idea of this experiment is to test a new method of selecting predictors. In experiment 2, we used the statistical selection method to select the predictors, as mentioned in 5.2.1 on page 147. Since this experiment is set up similarly to Experiment 2, we will only detail the differences here. There 12 predictors in this experiment, as shown in 5-10 on page 134. All the predictors are created and trained the same way, just as in Experiment 2. The only difference is the method of selecting a suitable predictor for the time. We applied the UCB algorithm instead of Statistical Selection. The framework of using UCB for predictor selection is shown in Figure 5-21.

Figure 5-21 shows how UCB was applied to our predictor selection problem. The framework starts once all the predictors have been trained and ends after obtaining the predictor to be selected for the next round. The processes before and after this framework is applied have been discussed previously, therefore they will not be discussed again in this section. The UCB framework is composed of 4 steps. Step 1 is the initialisation. There are 12 trained predictors which are shown in blue rectangles in this step. The dark and light blue rectangles represent the predictors for low and high

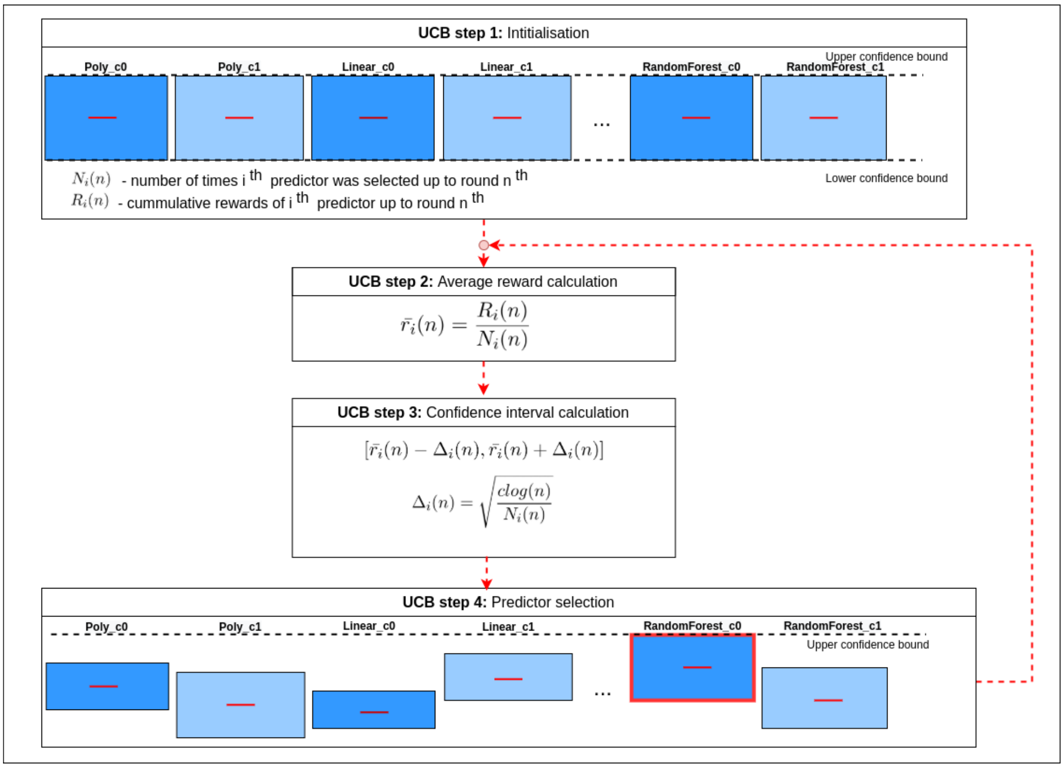


Figure 5-21: UCB framework

volatility clusters, respectively (details can be seen in figure 5-10 on page 134). All of these predictors start with the same default parameters, which are the number of times the predictor was selected (N_i) and the cumulative reward (R_i). These two values are set to zero at the beginning. The red line in the middle of each box shows the average reward of each predictor. The horizontal black dashed lines at the top and bottom of the boxes are the upper and lower confidence bounds, respectively. Please note that all three of these values start at zero and we show them in top-middle-bottom positions for visualisation only.

Step 2 is the average reward calculation. Once the system starts, the predictor will earn a reward if it produces the right predictive signal. This step calculates the average reward of each predictor up until that point of time. Step 3 is the confidence interval calculation. From this stage, each predictor will recalculate the upper and lower confidence bounds, which will be taken into account in order to select the predictor in the next step. The final step is predictor selection. In this step, the predictor with the maximum upper confidence bound will be selected. Only this selected predictor is used to make a decision for the next day.

The process from step 2 to step 4 will be repeated until the end of the testing period. The average rewards and confidence bounds of the selected predictor will be recalculated each round based on performance. During the testing period, the choice of predictor will keep changing to try and balance between exploration and exploitation, except at the beginning when all predictors will be given an equal chance to perform the prediction.

5.3.1.2 Results

In this section, the results from a few stocks are shown as examples. These results will be illustrated in graphs for clarity. At the end of this experiment, the results from all 11 stocks we have investigated will be shown in a table. Moreover, these results compare the performance of every predictor-selection method we investigated. The first stock we experimented on is CARR, the result of which can be seen in figure 5-22.

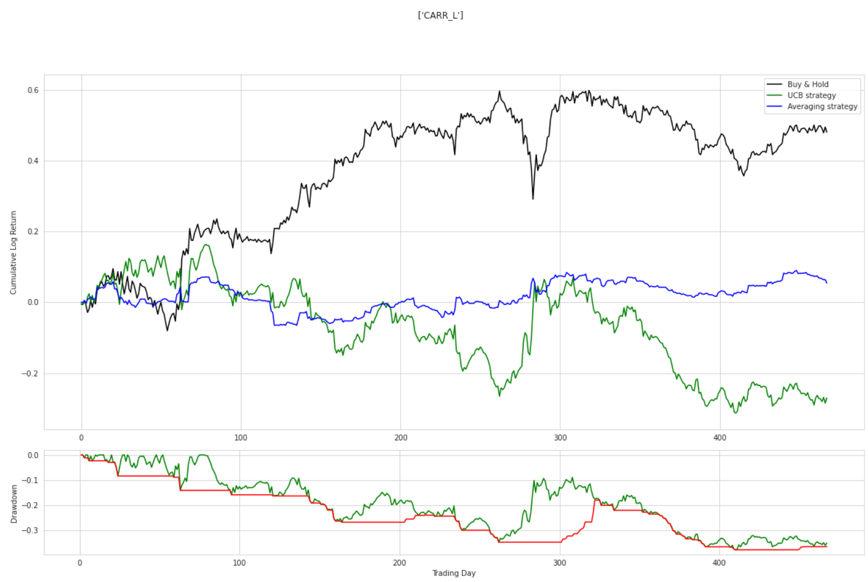


Figure 5-22: Comparison results of using UCB and averaging strategies with the buy & hold on CARR

When discussing the result in figure 5-22, we will not refer to the results from experiment 1 as the bias was included in that experiment. The results we will discuss start from experiment 2. Figure 5-18 on page 151 shows the result of using polynomial regression on CARR. Since the result was not too bad (Sharpe Ratio equal to

0.84), one might question *why we would add more machine learning predictors*. Recall from experiment 2, polynomial regression did not work well with all datasets, such as MKS (discussed in section 5.1.2.1 on page 132). That is why we added more algorithms. Having more algorithms increases the ability to predict across a wide-range of datasets. Moreover, we have shown in figure 5-20 on page 157 that having more algorithms is helpful. However, it will only work when we have an effective method to select the correct predictor. Otherwise, we cannot obtain good results. For example, averaging the answers from all predictors does not work in figure 5-20. Therefore, in this section we start to investigate predictor-selection methods.

The first predictor selection method we used was UCB. The result of applying UCB on CARR is shown in figure 5-22 on page 162. This figure comprises two sub graphs. The top sub graph compares the cumulative returns of the buy & hold benchmark strategy (black), UCB strategy (green) and averaging strategy (blue). It can be seen clearly that averaging answers from every predictor does not work at all as the graph shows how badly the blue line does. Even though we know polynomial regression works well on this stock, the averaged result suffers from the bad predictions.

The result from using UCB to select predictors is even worse than those of the averaging method. The green graph illustrates the worst result out of these three strategies. UCB ends up with a -23.31% loss, while the buy & hold and averaging methods do better, profiting 62.38% and 5.55% respectively. The drawdown of UCB is also worse than the other strategies, as can be seen in the bottom sub-figure. The maximum drawdown of UCB is -37.90%, while the buy & hold and averaging strategies have smaller drawdowns of -26.32% and -12.79% respectively. It is clear from the results that UCB does not work on CARR and that we must run the experiment on more stocks. The next stock we have selected to show as an example is TSCO, the results of which can be seen in figure 5-23.

Figure 5-23 shows the result from TSCO. UCB does not work well for TSCO either as we can see from the top sub graph. This sub graph compares the cumulative return of the UCB (green), averaging (blue) and buy & hold (black) strategies. These three strategies all end up with losses: -32.43%, -5.93% and -44.86%, respectively. Although UCB ends up with a smaller loss than the buy & hold benchmark, it performed poorly most of the time and could be considered too risky for an investor. An investor would have lost money for the whole time until the very end of the testing period. It is likely that an investor could not bare with such a loss and would have exited positions before the end of this period.

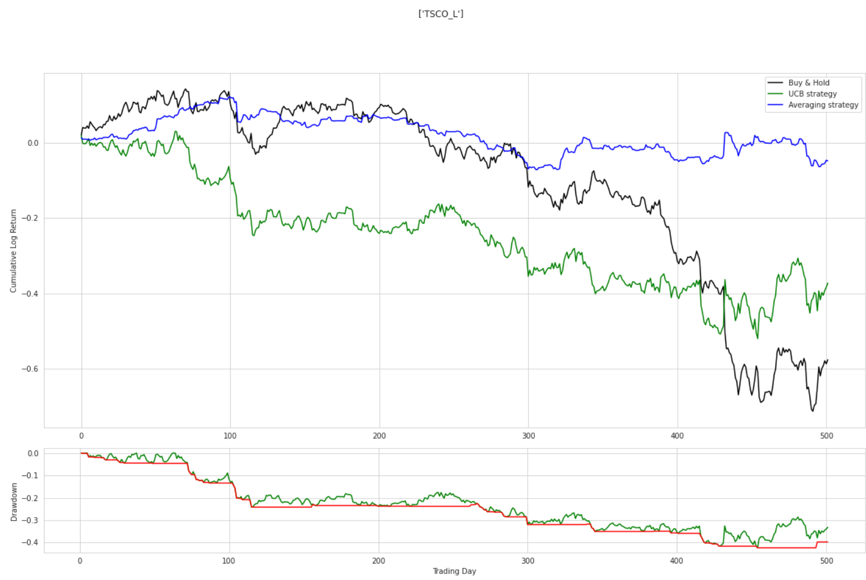


Figure 5-23: Comparison results of using UCB and averaging strategies with the buy & hold on TSCO

Unlike for CARR, the averaging strategy seems to work better than the buy & hold. The result from the averaging model was very good for the last 200 days. When the buy & hold 's performance started to go downhill, the averaging model stayed flat (although we would prefer to see it going up). As for the first 300 days, even though the averaging method could not win over the buy & hold completely, it was less volatile. This means that an investor faces less risk when investing using the averaging model.

The bottom sub graph shows an even worse result from UCB. The green line in this sub graph shows the drawdown of the UCB strategy. It can clearly be seen that the loss continuously gets worse over time. The downward trend of this graph means the loss kept getting bigger. The biggest loss or the maximum drawdown is -42.32%, which means that the investors need to stomach a loss of almost half their money. A bigger loss means much more difficulty and more time taken to return to profit. Therefore, this level of drawdown is likely unbearable for most investors.

5.3.1.3 Discussion

From the results of the two stocks above, it can be seen that UCB did not work well as a predictor-selection method. The results from the other stocks tested show the same conclusion. All results from UCB can be seen later in the comparison section, in table 5.3 on page 187. In this section, we will discuss why UCB did not work as well as we expected. To understand how UCB works, we will show the results from less complicated experiments. The results we will show in this discussion were obtained by running experiments on CARR using UCB with 2 predictors. Both of them are polynomial regressors, but each of them was trained with a different cluster of data (high and low volatility). Other parameters, such as the period of testing, stay the same as in the experiment above. The results are shown in figure 5-24.

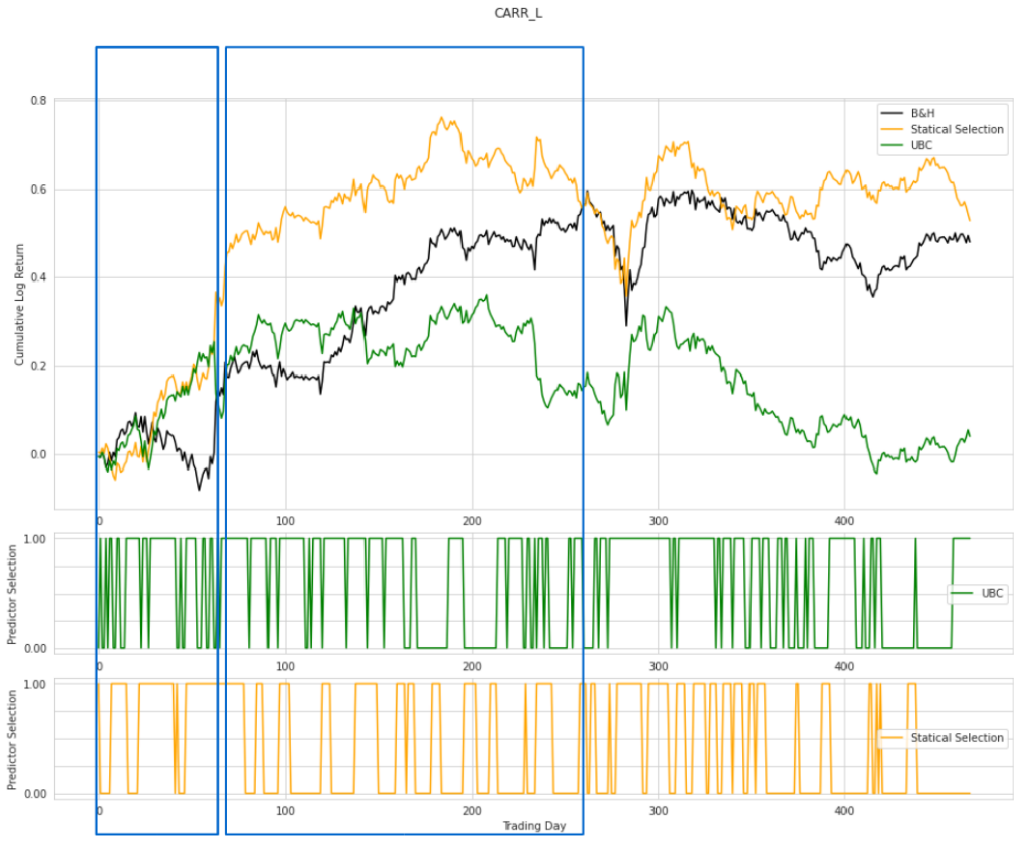


Figure 5-24: Comparison results of using UCB and averaging strategies with the buy & hold on TSCO

Figure 5-24 comprises three sub-figures. The top sub-figure shows a comparison of the cumulative returns from the buy & hold (black), statistical selection (yellow) and UCB (green) strategies. Please note that when we have only 2 predictors we can use the statistical selection method instead of the averaging method as in 5-22 or 5-23. We list the differences between these two methods below:

- Statistical Selection method: This method selects the predictor based on the mean and standard deviation. Therefore, when we have only two predictors, the system can select predictors based on the cluster that the data fell into: high or low volatility.

- Averaging method: This method is used in cases with more than two predictors. We cannot select a suitable predictor by considering whether they are high or low volatility. Therefore, we average the results from all predictors to create the final answer. This method is used when we do not have any other strategies for predictor selection.

It can be seen from the top sub-figure that the best predictor selection method here is statistical selection since it performs better than buy & hold most of the time. The worst model is UBC, which ends up with only 4.70% profit while the buy & hold and statistical selection end up with 62.38& and 68.83%, respectively.

The middle and bottom sub-figures represent the choice of predictors. The X-axis shows the trading days while the Y-axis shows which predictor is selected at a time. Y equal to 0 means predictor 0 (a predictor for low volatility) is selected. On the other hand, when Y equals 1, it means that predictor 1 is selected on that day. The middle sub-figure shows choice of predictor made by the UCB strategy, while the bottom one shows the choice from the statistical selection method.

Overall, it can be seen that the UBC strategy changes predictors more often than the statistical selection method. Another difference is that UBC seems to favour predictor 1 more than predictor 0, while statistical selection has more of a balance between the two predictors. Having a closer look into the beginning of the test period, in the first blue rectangle (about the first 70 days), UBC tries to perform exploration by changing the predictors often. As mentioned in section 3.2.5 on page 76, the UBC system is about the trade off between exploration and exploitation. In the second blue rectangle, UCB starts to exploit. It mostly stays with predictor 1 while the statistical selection method mostly selects predictor 0. In this case, predictor 0 seems to be the right predictor since the cumulative return of statistical selection is distinctively better than the UCB (as can be seen in the top sub-figure).

From the results above, it can be concluded that UCB as used here does not work. UCB changes predictor more often than the statistical selection method because it tries to perform exploration. Under more realistic conditions, such as including brokerage fees, UCB might have worse performance as it costs investors more by keep changing the position. Therefore, even though UCB is very suitable for the advertising problem, it does not work well for stock prediction. In the next section, we will investigate more predictor-selection methods in order to find a better strategy than UCB.

5.3.2 Scoring System

Since the predictor-selection techniques we have investigated so far - statistical selection and UCB - did not perform as well as we expected, this section introduces a new predictor-selection strategy: the scoring system. The basic idea behind the scoring system is the reward and punishment of predictors. After all of the predictors have been trained, they start working (predicting). In the beginning, all predictors have the same chance to predict. Then, the scoring system will reward the predictors that gave the correct answer. The successful predictors will be given different scores, higher scores if their answers lead to more profit. On the other hand, predictors with the wrong answer will be punished by having their score reduced. Similarly, a higher score will be taken away if their answers lead to a bigger loss. Therefore, the predictor with the highest score is not only able to provide the most correct answers, but also provide it on the correct days (days with the highest profits). Using this method, the scoring system will start to select predictors based on past performance. Predictors with high scores will be selected more often, potentially leading to better overall trading performance.

The scoring system sounds sensible, but since the score is based on the past we do not know whether selecting predictors using their scores will work. The movement of a stock price can be extremely volatile; Can a scoring system switch between predictors quickly enough to catch up with the changes? With this question in mind, we started the following series of experiments. There are three main experiments in this investigation. We will start with an explanation of each experiment and show some stocks' results as examples. Then, at the end of this section, a comparison of the results from each experiment will be provided, alongside the results of the final scoring model we designed.

5.3.2.1 Experiment Design

This experiment was designed the same way as the previous experiments. The only difference is the predictor-selection method. Therefore, we will only explain the predictor-selection process. For the details of the other steps, we will provide references to other sections where these have already been discussed. There are six steps in this experiment. Steps 1 - 3 are data preparation, clustering and training, which have been discussed already. The details can be seen in 4 on page 88.

In step 2, the number of clusters is two, representing high and low volatility. The number of machine learning algorithms in step 3 is six. Therefore, there are 12 predictors altogether in this experiment. The training process in this experiment is the same as in experiment 1.2, figure 5-10 on page 134. However, instead of adding two more predictors at a time, all predictors will be working together throughout this experiment.

The heart of this experiment is step 4, where the scoring system is built. Therefore, the details of this step will be explained here. The scoring system is one of the predictor-selection methods we are investigating, and its main idea is gradually building up a score that will be used to pick suitable predictors. The complete framework of step 4 is shown in chapter 4 in figure 4-7 on page 94. To obtain this complete scoring system, three experiments were conducted. Before providing the details of each experiment, a brief explanation of each experiment will be provided to give readers an overview. In the complete framework of the scoring system, we have highlighted the different parts with different colours according to each experiment in figure 5-25.

- **1st experiment (Only Score):** This experiment relates to the red highlight in figure 5-25. Since this is the first design of the scoring system, it is less complicated than the later ones. In this model, the score assigning process will give points to the predictors if they provide correct predictions. On the other hand, points are taken away for wrong predictions. The predictor selection criteria in this experiment is the score only. At this stage, the intermediate-model, called the Max Score model, acts as the final model.

  In the final model, the predictor(s) with the highest score will be chosen to predict for the next day. This method works well in some cases. However, it has the problem of the ability to switch the predictors quickly enough to cope with changes (examples will be given later). Therefore, we move on to the next experiment.

- **2nd experiment (Multiple Feature):** This experiment added more features to the model, which can be seen in the blue highlighted boxes in figure 5-25. As mentioned earlier, the cumulative score is not good enough. We included the other features in order to find a better feature to replace the score. The features we have considered are the exponential moving average, derivative and slope of

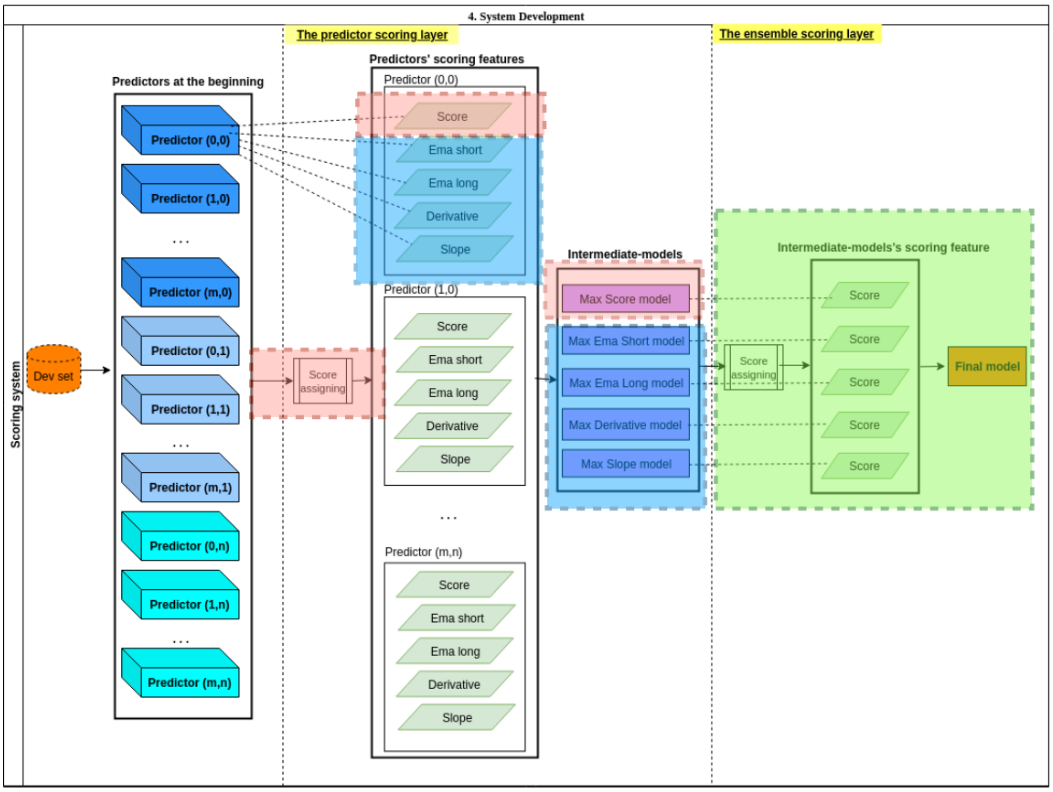


Figure 5-25: Scoring system separating into different experiments

the scores. The reasons for selecting these features can be seen on page 98. As in the previous experiment, at this stage we did not have the ensemble scoring layer yet. The intermediate models are acting as the final model. Therefore, there are five different final models all together. And we compare these models to decide which is the best to select suitable predictors.

However, the results from this experiment did not show clearly which model is the best. The performance of each keeps changing over time and across different stocks (examples will be given later). Therefore, we concluded that there is no best feature to use as the predictor-selection criteria. The best way is to put them to work together. This leads to the next experiment, where we applied the same technique as that which made the predictors work together. Therefore, the a second layer of scoring system was created, called the ensemble scoring layer.

- **3re experiment (Two-layer Scoring system):** This experiment produces the final part of the scoring system design. The additional design is shown in a green highlighted box. As mentioned in the second experiment, there are five final models (Max Score, Max Ema Short, Max Ema Long, Max Derivative, Max Slope models) and we cannot conclude which one is the best. Therefore, we designed a second layer of the scoring system, called the ensemble scoring layer. In this added layer, we create another score-assigning process. This process works similarly to the first score-assigning process (in the red highlighted box), except it will give or take points for the final models. Now we are adding a new layer to manage these models, they have become intermediate models.

Since the results from this experiment seem to be the best out of these three experiments, we have summarised the final design of our scoring system in figure 5-25. More details and some examples of the results of these experiments will be discussed in the following sections.

5.3.2.2 Scoring experiment 1: Only score

This experiment was designed as mentioned above. We started investigating the scoring system with a simple idea: Predictors gain higher scores for correct predictions and lose points for incorrect ones. Then, the predictor with the highest score will be selected to predict for the next round. higher scores will be given or taken away in cases where predictors provide correct or incorrect predictions on big change days (leading to a large profit or loss).

There are four different score levels that will be given to our predictors. The best predictors are given a full score of 1. These are predictors that give buy signals correctly on days that return a large profit and sell signals on days which a large loss would have been incurred had the shares been held. This is a very important feature of our system, since avoiding large losses is at least as important as making a profit in financial markets. This is especially important as it requires a greater percentage gain to return to breakeven after a loss than the original percentage loss itself, which can greatly diminish compound returns. Accurate predictors which give correct answers but on days with small profits or losses will be given 0.5. Conversely, predictors that work very poorly will be punished with -1. These are predictors whose signals lead to large losses or significant missed profits. The latter is an example of how we have incorporated the economic concept of opportunity cost into our model, opportunity cost being the idea that a missed profit opportunity is theoretically equivalent to a loss. Such costs can significantly affect compound returns in the long run. Inaccurate predictions on days with less price action will be punished with only -0.5. To set the thresholds of this scoring system, we calculated the different quartiles from the distribution of the return from our training set. Details of this point threshold can be seen on page 96.

- **Experiment Design**

  The framework of this experiment can be seen in figure 5-26.

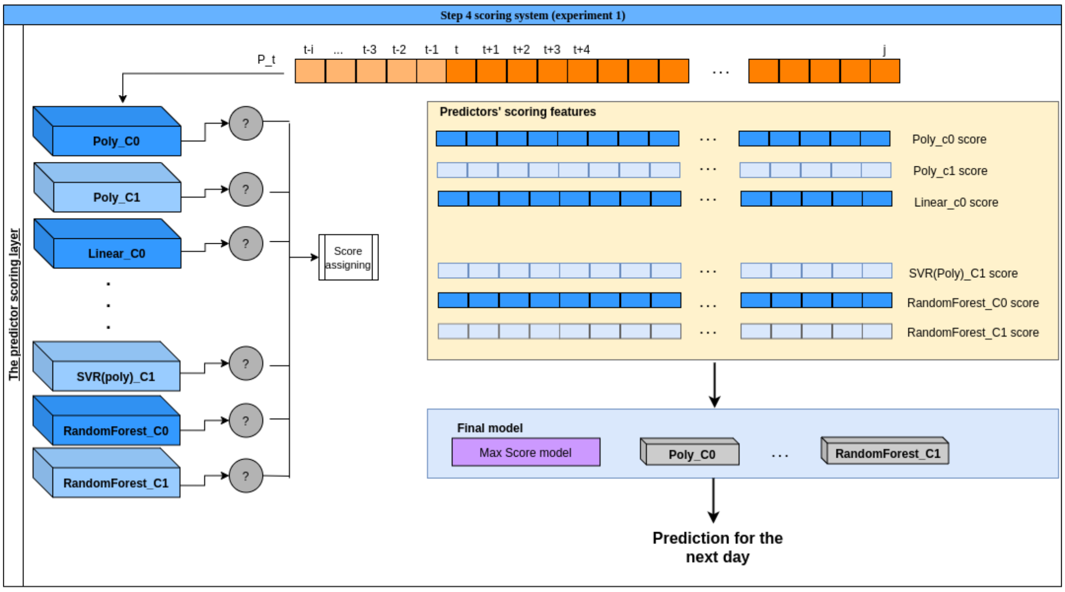


Figure 5-26: Scoring system design in scoring system experiment 1 (Only Score)

Figure 5-26 shows the framework of the first scoring system experiment, which only investigates the score. The light and dark blue boxes on the left of the framework represent the 12 predictors we have in this experiment. Ever predictor will start predicting from day t (after the look-back period). At the end of the day, all predictors are given the closing price of the stock. The predictors use that information together with the previous information they had and predict the price direction for the next day $(t+1)$. The answer from a predictor is shown in the grey circle next to it. The answer or predicted value is a real number: the expected return. This predicted return will be changed to -1, 0 or 1. A result of -1 means that a predictor has predicted that the price will go down. On the other hand, a result of 1 means the future price is expected to increase. If a predictor predicts no change is to be expected, the predicted value will be 0.

After the return is predicted on day $t+1$, the scoring system will wait until the end of day $t+1$ to start working. It will compare the actual return and predicted return, then predictors will be rewarded or punished based on their performance.

In the framework, scores are collected in the predictors' scoring feature section (the yellow rectangle). The criteria of the score given can be found on page 96. After assigning points to the predictors, the cumulative scores are computed. Then, the predictor(s) with the highest score will be selected to be in the final model, as shown in the blue rectangle at the bottom-right of framework. The final model, or max score model, is able to have more than one predictor. If multiple predictors have been selected, the scoring system will use a majority vote in order to give the final signal (buy, sell or hold). Then, when the market opens again on the next day, the trading system will perform an order according to the final signal given by the scoring system. These are the details of the initial scoring design. The next section shows some examples of the results from this experiment.

- **Example of results**

  The results of testing this scoring system with our 11 stocks did not turn out as we expected. Most of the results were not good, meaning the scoring system performed worse than the buy & hold strategy. There were three stocks (COST, FERG and SXS) where the scoring system worked better than the buy & hold occasionally, and there is only one stock (TSCO) for which it worked significantly better than the buy & hold. In this section, we will show some examples of the results. Then, all the results will be shown in the comparison section after finishing scoring-system experiment 3.

  The first stock we will discuss is TSCO. As mentioned, the scoring system worked much better than the buy & hold for TSCO stock. The result is shown in figure 5-27.

  Figure 5-27 shows the result of the experiment on TSCO stock. There are two sub-figures: the top sub-figure compares the cumulative return of each strategy. The X and Y axis show trading days and cumulative return, respectively. There are 24 graphs in this sub-figure. Each graph shows the cumulative return of each strategy. The black thick graph is the buy & hold result which, can be seen going down overall, especially from 2014-07 when the price decreased sharply. Even though the buy & hold benchmark performed badly, the scoring system worked better here, as shown by the thick green line. While the buy & hold ended up with a -44.86% loss, the scoring system provided a 50.55% profit. As for the Sharpe ratio, the scoring system increased the Sharpe ratio from -1.35 (for buy & hold) to 0.73. The standard deviation of the return from the scoring system

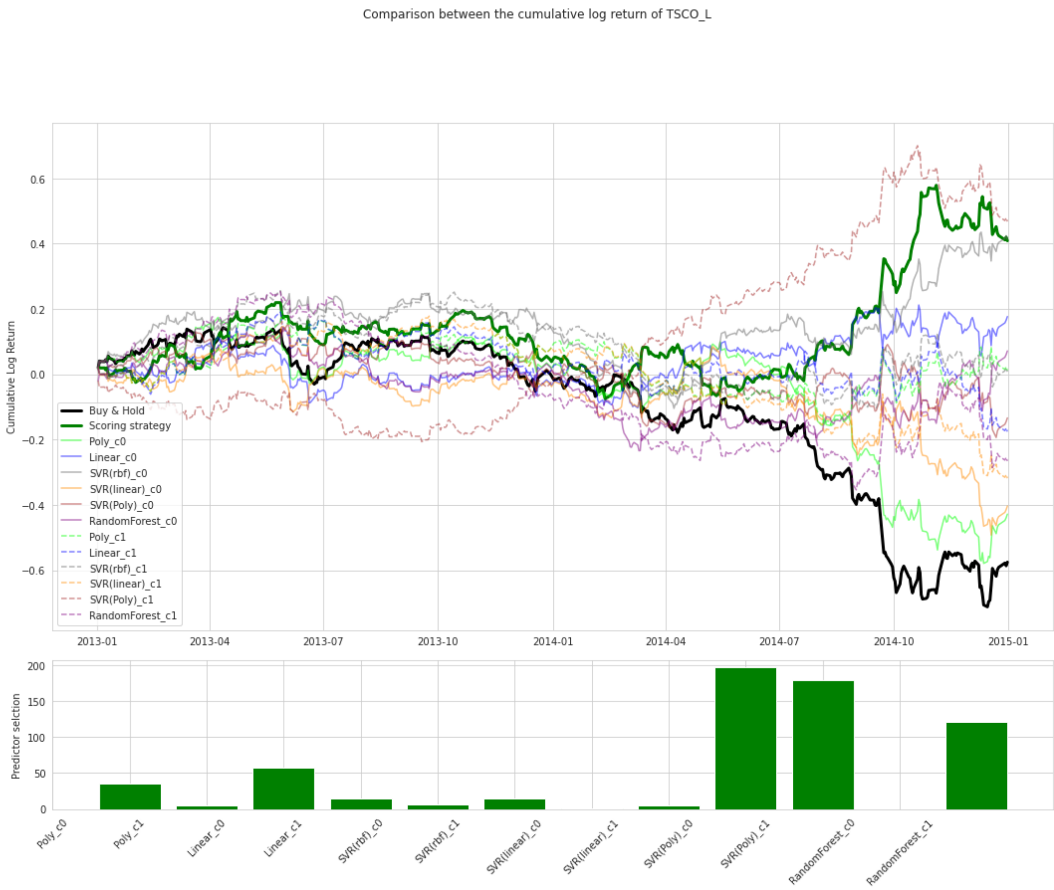


Figure 5-27: Scoring system design in scoring system experiment 1 (Only Score)

also decreased from 0.2444 (buy & hold) to 0.239.

Apart from those two thick lines we discussed above, there are many faint lines in this sub-figure which represent the results from each individual predictor. Dashed lines are the results from cluster 1 predictors (high volatility), and the rest are the results from the cluster 0 predictors. From the beginning of this period until after the beginning of 2014, most of the predictors seemed to work similarly. This is because the stock's price during this period was fairly flat. However, the situation changed dramatically after the 2014-04. When the price dropped to its lowest, most of the predictors seem to give different answers. Most of them tried to stay flat as the price had been quite flat until 2014-07, for example Linear_C0 and Poly_C1. Some of them lost money along with the buy & hold benchmark, such as Poly_C0 and SVR(linear)_C0. However, some predictors seem to cope with the changes and stay profitable, such as SVR(rfb)_C0 and SVR(Poly)_C1.

Our scoring system seemed to work really well in this situation. It managed to pick the correct predictors to follow, here SVR(rfb)_C0 or SVR(Poly)_C1 or both. There was quite a big lag when switching, but when it finally followed the correct predictors it ended up profitable.

The bottom sub-figure shows how often each predictor was selected. The X-axis shows the number of times predictors were selected, while the Y-axis shows the name of each predictor. As expected, predictor SVR(Poly)_C1 is the most popular. It was selected almost 200 times meaning it was the most often-used predictor. As expected, this predictor started to be selected after 2014-04 which made the scoring system profitable.

The first result (TSCO), saw our system work very well. Some contrasting results will be presented in figure 5-28.

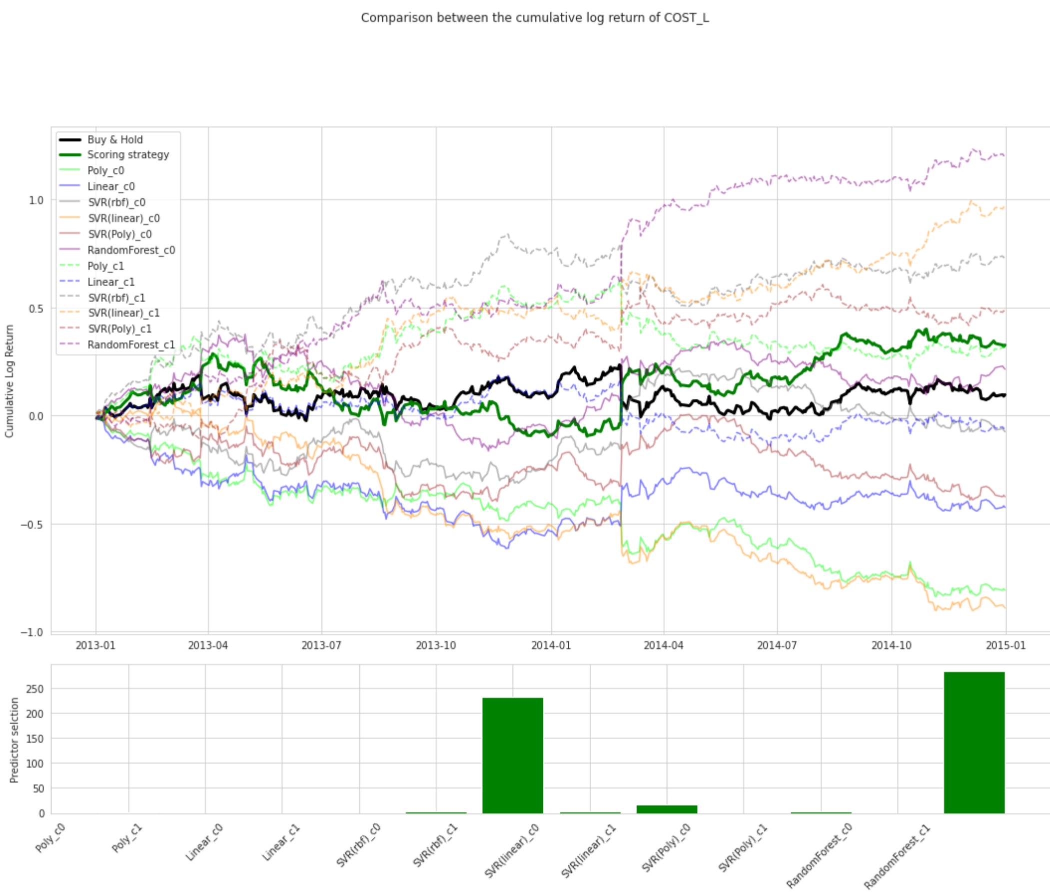


Figure 5-28: Scoring system design in scoring system experiment 1 (Only Score)

Figure 5-28 shows the result from COST, which is very different from TSCO. As can be seen, predictors seem to have different answers at all times for this stock. The faint lines perform differently. It looks like none of them have any idea how to cope with this stock, which confused the scoring system so it was unable to pick the best predictor. Up until about 2014-03, the scoring system selected the wrong predictor, SVR(linear)_C0. Unfortunately, SVR(linear)_C0 was one of the worst predictors for this stock. In fact, it ends up with the largest loss at -59.49% and a Sharpe Ratio of -1.72, which is also the worst Sharpe Ratio.

- Discussion

This experiment tested the first design of our scoring system. The design is simple. The scoring system gives or takes points from each predictor based on their recent performance. Then, for the next round (or next day), the predictor(s) with highest cumulative score will be selected to provide the next day's signal to buy, sell or hold. In this experiment, there was only one final model, called the max score model. In this max score model, there can be more than one predictor. For example, in the case that multiple predictors share the same highest cumulative score. If there is more than one predictor in the final model, the final answer (or signal) can be obtained by performing a majority vote. If a majority vote does not work - because the number of buy votes and sell votes are equal - the final signal will be to hold any open positions or not buy anything if no positions are open. Please note that it is also possible that no predictor is selected for the final model if all of them have negative cumulative scores. In this case the system will not open any new positions, thus the signal is to hold.

As the results of this initial version of the scoring system were not very good, we cannot rely on this criterion alone for predictor selection. We believe the reward and punishment principle is valid, but basing the signal on the cumulative score alone results in poor performance because the predictors in the final model cannot change quickly enough; we have to wait for their cumulative score to gradually decrease. So, the result will turn out badly as shown in figure 5-28.. In order to respond to this, we investigated further other features that could improve the scoring system. This will be discussed in the next experiment.

5.3.2.3 Scoring experiment 2: Multiple features

As the simple scoring system did not work well, this section introduces more features to be considered for incorporation into the scoring system. We concluded in the last section that the problem of using only the score might be that the score alone is not a good enough feature to identify the best predictors. For example, some predictors might have done well recently and have achieved a high cumulative score, but now their performance is going downhill sharply. But the cumulative scores take time to decrease. This makes the scoring system unable to switch to better predictors in time. Therefore, this section introduces new features to identify the best predictors, instead of only the score. The features which will be considered in this experiment are the exponential moving average, derivative, and slope of the scores. The reasons for selecting these features can be seen on page 98.

- **Experiment Design**

  The new features we added to the scoring system are the exponential moving average (EMA), derivative and slope of the cumulative score. The reasons for selecting these features can be found on page 98. The framework of this experiment is shown in the following figure.

  Figure 5-29 shows the framework of the scoring system for experiment two. As can be seen, it looks similar to the previous experiment's framework (5-26). Therefore, we will only describe the additional features we are testing. Firstly, there are additional features in the yellow box, which is no longer composed of only the cumulative score any more. Each predictor has extra features (EMA short, EMA long, derivative and slope). At the end of the day, the scoring system will compare the actual return and the predicted return, then add or deduct points for every predictor as before. However, in this experiment it will also calculate the other feature values of EMA short, EMA long, derivative and slope. Each value will be calculated daily. The scoring system will use these different features individually to select the predictors which provide a signal for the following day, as opposed to the first experiment which only chose the highest cumulative score. Now, the system will select the predictor(s) with the maximum EMA short value to create the max EMA short model. Likewise with EMA long to create the max EMA long model, and so on. Finally, instead of having only one final model - the max score model in the previous experiment - there will be four more final models which can be seen in the blue rectangle box.

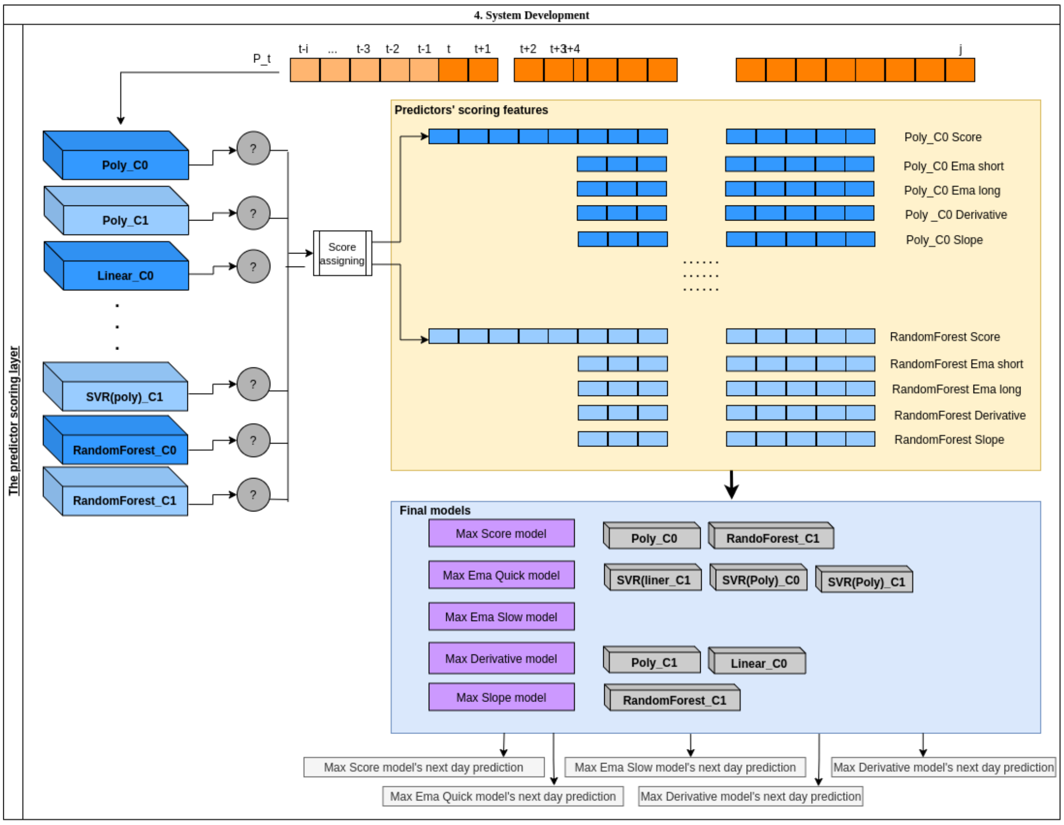


Figure 5-29: Scoring system design in scoring system experiment 2 (Multiple features)

Each final model is able to contain many predictors or it can have none. If it has more than one predictor inside, the final signal will be obtained by majority vote from all predictors contained. Finally, each final model will provide their signal, either sell, buy or hold. Therefore, each final model is one strategy. At the end of the experiment, the performance of each strategy will be calculated and compared. When we started this experiment, we were hoping to discover the best final model to use in our trading system. However, the results did not turn out as we expected (some examples of the results are shown in the next section). We are unable to draw a conclusion as to which is the best feature, which leads us to the next experiment and adding a second layer to the scoring system.

- **Example of results**

By adding more features into our scoring system, we produced five different final models (one model for one feature) from which we hoped to select the best model (feature) to select suitable predictors for our system. However, the results were inconclusive. We tested these versions of the scoring system with 11 stocks, as in the other experiments. The results show that the best model kept changing over time or was different between stocks. Here, some examples of the results will be presented. The first result is TSCO, which can be seen in figure 5-30.

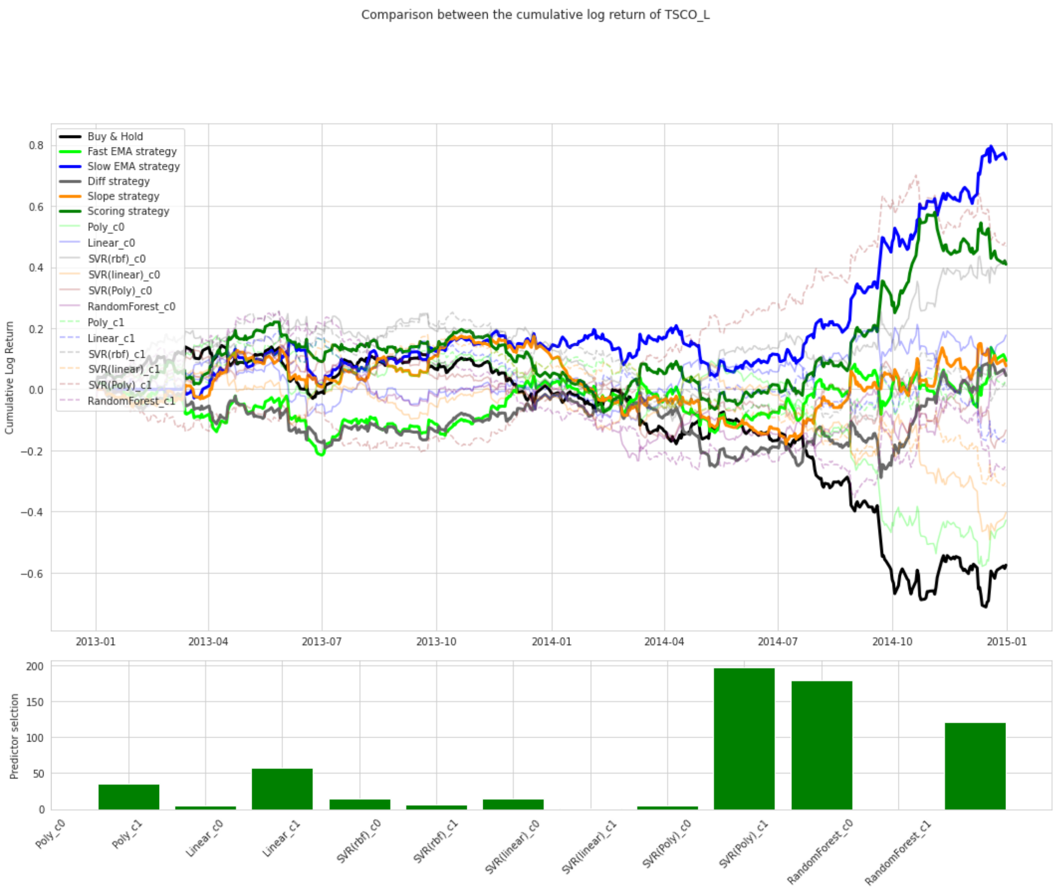


Figure 5-30: Result of TSCO from scoring system experiment 2

Figure 5-30 shows results from five different final models (max score, max EMA short, max EMA long, max derivation and max slope). As in the previous experiment, the top and bottom sub-figures represent the cumulative return and the predictor selection. In the top sub-figure, there are six thick lines representing results from the buy & hold (black), EMA short (light green), EMA long (blue), derivative (grey), slope (orange) and scoring (dark green) strategies. As for the faint lines, they represent the results from individual predictors as mentioned in the previous experiment. In this figure, we have to make them more faint for the reader to see the results from each added feature with ease. As can be seen, the scoring system itself works very well on this stock (as mentioned earlier). However, when compared to other features, the EMA long model seems to outperform the scoring system from 2014-01 onward.

After 2014-01, the EMA long strategy (blue) switched to a better predictor much quicker than the scoring system, as it started to follow the best predictor (SVR(Poly)_C1) shortly after its performance increased. Moreover, when the SVR(Poly)_C1 started to decrease, the EMA long model switched to following the new leading strategy (SVR(reb)_C0). This meant the it was able to remain profitable through to the end of the testing period. Finally, the EMA long model ended up with the most profit, at 112.39% over the period of two years, while the scoring and buy & hold strategies only attained 50.55% and -44.86%, respectively.

From the TSCO result above, it might look like the best strategy is the long EMA. However, this was not always the case. For example, see the results from COST in figure 5-31.

It can be seen from result of COST in figure 5-31 that the EMA long (blue) is not the best strategy any more. In fact, it is almost the worst. It performs badly throughout the period of testing. Clearly the best strategy can change dramatically, and even become the worst. It is not easy to identify any single suitable feature to use in the scoring system. In the following figure, we show another two results, from OXIG and MCD respectively.

Figure 5-32 shows results from OXIG and MCD respectively. It can be seen that we cannot clearly identify the best strategy (or feature). From sub-figure (a), the result of OXIG, the derivative strategy (grey) seems to provide the best result overall. However, in sub-figure (b), the result of MCD, the derivative turns out to be one of the worst-performing strategies compared to the others, including the buy & hold strategy. We show these results to demonstrate how the best feature seems to change over the period of time or over different datasets. Therefore, we cannot identify the best feature to include in our system from the results of this experiment.

- **Discussion**

In this experiment we tested more features in the scoring system in order to identify the best feature for selecting appropriate predictors. We tried five features, all of which following the results of the previous experiment, which showed that having a high cumulative score is not adequate to use as a prediction selection

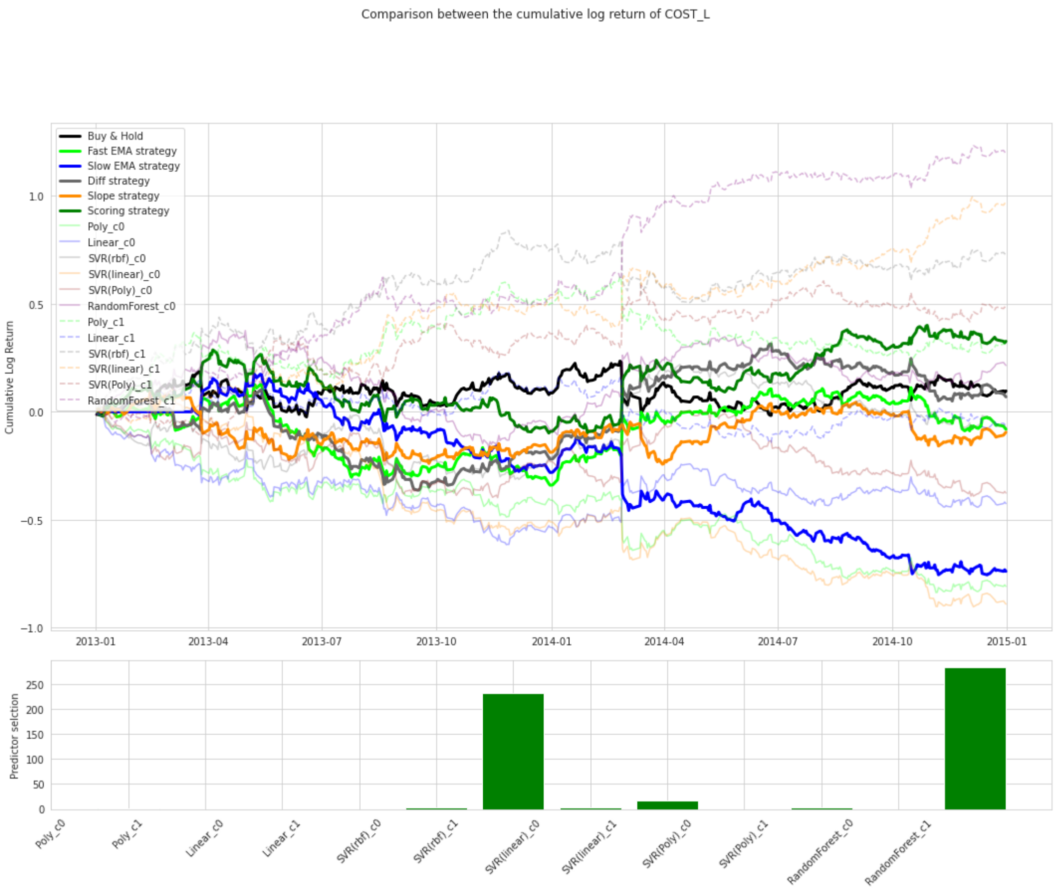
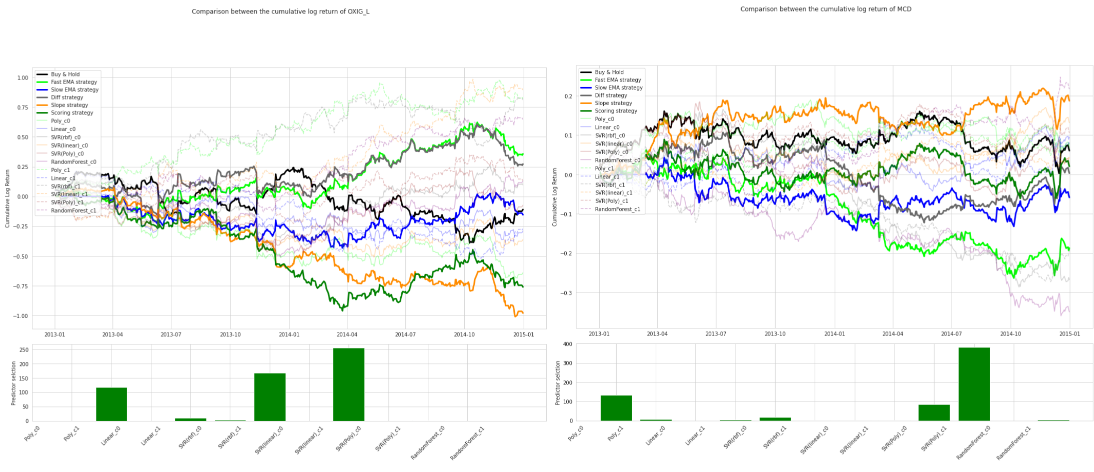


Figure 5-31: Result of COST from scoring system experiment 2

criteria. The details of the five features can be found on page 98.

We expected that the results from this experiment will help us to identify the best feature to use for switching our predictors over time. However, the results did not turn out as we expected. The best feature changes over time and over different datasets, like the predictors do. We conclude that there is not a best feature that can be used as the main criteria to select the best predictors. Therefore, we decided to create a second layer of scoring system to control the use of the features. This second layer will select a suitable feature to choose predictors. For example, when the price changes rapidly, selecting predictors based on their cumulative score is too slow and harms performance, but the long EMA may work very well. Ideally, this second layer of scoring will manage this and hopefully improve the trading system.



(a) OXIG (b) MCD

Figure 5-32: Results of scoring system experiment 2 of OXIG and MCD

5.3.2.4 Scoring experiment 3: The second layer scoring system

- **Experiment Design** In the previous experiment we were unable to identify the best feature (or final model), and to solve this problem we added another scoring system, called the ensemble scoring layer. This new layer is shown in a green highlight rectangle in figure 5-25 on page 169. This model works in the same way as the first score assigning model that we used for the predictor scoring layer (shown in the red highlight rectangle in the same figure). It gives and takes points using the same criteria (details of this point threshold can be seen on page 96). However, this new score assigning layer will allocate points for the final models (each model is created by an individual feature) instead of the machine learning predictors. Therefore, the final models in this experiment will henceforth be called intermediate models. The final model going forward is obtained from the maximum intermediate-model score.

  We are aware that the scoring system itself did not work very well to select predictors, as shown in scoring experiment 1 (section 5.3.2.2 on page 170). However, in scoring experiment 2 (section 5.3.2.3 on page 176), the results showed that other features besides score could be useful. If we are able to select useful features at suitable times, this would increase our trading performance. Therefore, we designed this experiment to develop the feature-scoring layer. The framework of this experiment is shown in figure 5-33.

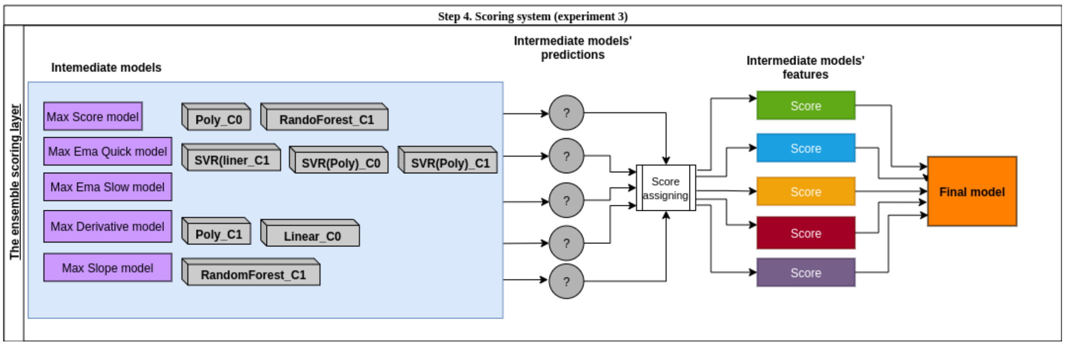


Figure 5-33: Framework of scoring system experiment 3 (only the added module)

Figure 5-33 shows the framework of scoring system experiment 3. The blue highlight rectangle shows the intermediate models (formerly called final models in the previous experiment) obtained from the previous experiment. In scoring experiment three, each intermediate model will be assessed by a new score-assigning module. The answer from each model will be compared with the actual return the next day and the models will be given or points or have them taken away, so each feature score is updated every day. Then, the intermediate model with the maximum cumulative score will be selected and used as the signal for the next day. This process is the same as the first layer of the scoring system (predictor scoring) so we will not go in to the detail here. If needed, details can be seen in the beginning of section 5.3.2 on page 167.

After running this experiment with our 11 stocks, the signal from the final model often turned out to be the best one. For some datasets, it did not provide outstanding performance, but it was able to overcome other models. The next section demonstrates some examples of the results, while all the results can be seen in the result comparison section on page 187.

- **Example of results**

The first example of the results shown in this section are for TSCO. The results are composed of two sub-figures, representing the cumulative return and number of times each predictor was selected. These results are shown in figure 5.3.2.

We discussed in the previous experiment how the EMA long strategy (blue line) worked better than the predictor-scoring system (in figure 5-30 on page 178) and seemed to be a better option. However, when we performed this experiment using the second layer scoring system to select the feature for the final model,

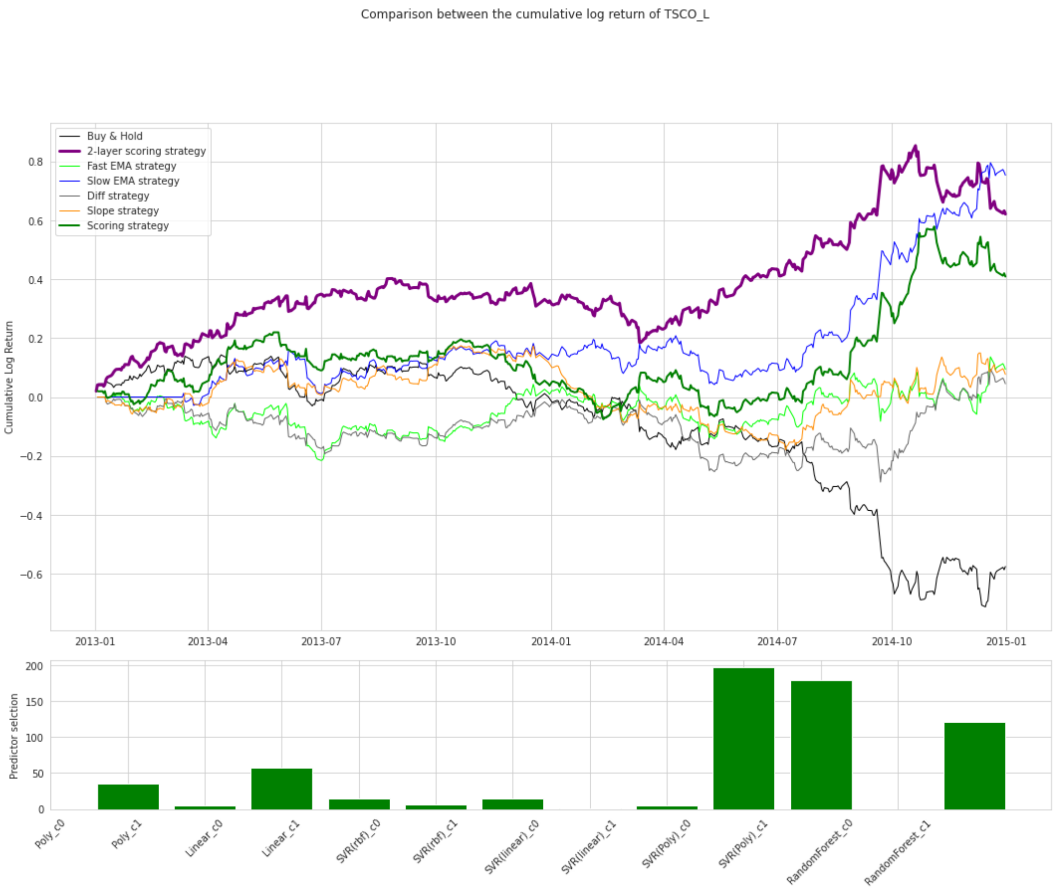


Figure 5-34: Result of TSCO from scoring system experiment 3

we outperformed the EMA long strategy for most of the testing period, although the second layer scoring system ended the period with a smaller profit. From this result it appears the second layer scoring system could be a beneficial addition to our trading system. In this case it was able to pick the best features to use to select the best predictors throughout the testing period, and it ended up with an 82.45% profit. Meanwhile, the buy & hold performed badly, losing -44.86%. The second layer scoring system also worked better than the predictor-scoring system, which did not do too bad by itself, achieving 50.55% profit.

It can be seen that the second layer scoring system outperformed the long EMA strategy for most of the test period, but finally ended with less profit. The long EMA strategy had the best profit (112.39%) and best Sharpe Ratio (1.46). As for the accuracy, the long EMA achieved 57.09% while the second layer scoring system got a little better at 58.88%. However, accuracy is not the main criteria

(a) CARR (b) MCD

Figure 5-35: Results of scoring system experiment 3 of CARR and MCD

we consider as we have seen that greater accuracy cannot guarantee higher profit. To be able to obtain higher profit, the system needs to be correct on days with high profit. For example, a system X provides the correct prediction 8 times out of 10, but on those days it only earns about 1 percent profit. Meanwhile system Y provides only 5 times correct answers, thus having lower accuracy, but there are two days where the prices increase rapidly, say 5% on each day. If system Y was correct on those days, while system X was not, system Y with lower accuracy will outperform system X on profit. The result from TSCO provides further evidence to support this argument. The long EMA strategy ends up with higher profit but lower accuracy than the second layer scoring strategy. This suggests highly profitable days were missed, but this result could be unique to TSCO, so we will discuss how the second layer performed for other stocks.

Besides TSCO, the second layer scoring system works well for other stocks too, for example CARR and MCD in figure 5-35.

Figure 5-35 shows results from CARR and MCD. It can be seen that the second layer scoring outperforms the other strategies. We tested our second layer scoring system on the other stocks too. The results were consistent and all of the results can be seen in section 5.4 on page 186.

- **Discussion**

This was the final experiment on our scoring system. The scoring system in this experiment is the complete version that will be used in our trading model, and the details of this model can be seen in chapter 4 on page 88. In this section, we discussed the design of our scoring system from its beginning to its final iteration. Our scoring system started with the basic idea of rewarding and punishing predictors according to their recent performances. The details and results of this application of the scoring system are shown in section 5.3.2.2 on page 170. In summary, the results from this first version of the scoring system were poor, and we concluded that the cumulative score of points allocated to the best-performing predictors is not consistent enough to be the sole feature used to select the predictors.

To solve the problem above, scoring system experiment 2 was created to try more features besides the scoring system. The objective of experiment 2 was to find the best feature to select predictors. We expected the best feature need be the only criteria to select the best predictors throughout the trading period. The four features we added were short and long EMA, derivative and slope. Details of these features and the design of this experiment can be found in section 5.3.2.3 on page 176. Unfortunately, the results were extremely varied. Identifying a single best feature was not possible. Therefore, we decided to try applying the scoring system - the same as we originally used to manage the predictors' scores in scoring experiment one - to the features instead. Since each feature created one intermediate model, the added module in the next experiment was called the ensemble scoring layer.

The ensemble scoring layer was added in scoring experiment three in order to be able to switch between intermediate models when the market situation changes. For example, when the market has a clear trend, the intermediate model related to the cumulative score is the best option, as the predictors which have been doing well should continue working well. On the other hand, when the market changes rapidly, the cumulative score does not work any more and is too slow to change predictors as their cumulative scores need time to decrease. This would result in the system losing money. The ensemble scoring layer should help solve this problem. When the market behaviour changes, it is able to switch effectively and quickly from one intermediate model to another, keeping the trading system profitable.

From the reasons above and the results from the three experiments, we conclude that the double scoring layer (features to select predictors, and the ensemble scoring layer to select features) design is suitable for our trading system, and we will use this in the final design. The details of the final design can be found in chapter 4 on page 88. The complete comparison of the results will be shown in section 5.4 on page 186.

5.4 Results Comparison

We performed many experiments in order to design an effective scoring system, which is the central focus of this research. Throughout this chapter, we have discussed how an ensemble model can outperform an individual predictor model, although we have only showed some examples of the results so far. In this section, we will provide all results and compare the strategies we have investigated in one table.

This comparison table aims to compare and summarise the performance of an individual machine learning algorithm and our ensemble-based scoring system. There are 11 strategies to be compared here. The first strategy is buy & hold, which we used as a benchmark. The next six strategies are individual machine learning algorithms, comprising polynomial regression, linear regression, support vector machines with 3 different kernels, and random forest. Following these machine learning algorithms, three ensemble-based strategies were designed to make multiple machine learning algorithms work together effectively. The first ensemble-based strategy was the simplest, which is the averaging strategy. The second ensemble strategy was the upper confidence bound, and in the penultimate column in the table below is the single-layer scoring system. Finally, the last strategy - shown in the last column of this table - is our complete scoring system, which is composed of two layers of selection: predictors chosen by features and features chosen by the ensemble scoring system. A comparison of all these results is shown in table 5.3.

Table 5.3: The comparison between Buy & Hold, Single predictor, Averaging and Scoring System methods, evaluated over two years (2013-2014)

Data/Model	B&H	Poly	Linear	SVR (rbf)	SVR (linear)	SVR (poly)	Random Forest	Average	UCB	1L Scoring	2L Scoring
CARR											
Profit	62.38	52.92	52.92	-28.57	214.15	24.8	35.53	5.55	-23.31	20.72	264.52
SP	0.77	0.66	0.66	-0.7	1.96	0.3	0.44	0.01	-0.57	0.24	2.33
Max DD	-26.32	-29.33	-29.33	-41.02	-19.35	-24.08	-30.59	-12.79	-37.9	-22.39	-12.16
ACC	52.89	52.46	52.46	46.9	55.25	49.89	49.89	57.39	48.39	53.53	60.39
Stdev	0.302	0.302	0.302	0.302	0.3	0.302	0.302	0.117	0.302	0.296	0.286
COST											
Profit	11.57	81.97	89.36	34.32	100.57	-13.0	15.53	11.43	42.91	38.62	192.69
SP	0.09	0.97	1.04	0.42	1.14	-0.35	0.16	0.4	0.53	0.49	1.84
Max DD	-22.26	-15.72	-15.72	-24.01	-18.06	-26.01	-35.95	-8.03	-33.21	-31.97	-22.90
ACC	51.26	51.46	51.88	50.42	52.51	47.91	48.95	64.23	51.88	52.51	59.62
Stdev	0.296	0.295	0.295	0.296	0.295	0.296	0.269	0.067	0.296	0.291	0.291
D4T4											
Profit	39.1	-27.22	-27.22	102.03	-1.64	9.78	60.81	19.26	21.37	3.27	196.44
SP	1.19	-1.25	-1.25	2.63	-0.11	0.3	1.75	1.56	0.68	0.07	4.29
Max DD	-37.76	-46.08	-46.08	-21.17	-39.99	-39.94	-24.22	-5.89	-32.53	-25.86	-17.96
ACC	51.75	49.12	49.12	50.88	50.88	53.51	54.39	56.14	50.88	53.51	66.67
Stdev	0.586	0.586	0.586	0.58	0.588	0.585	0.584	0.23	0.587	0.573	0.552
FERG											

Continued on next page

Table 5.3 – *Continued from previous page*

Data/Model	**B&H**	**Poly**	**Linear**	**SVR (rbf)**	**SVR (linear)**	**SVR (poly)**	**Random Forest**	**Average**	**UCB**	**1L Scoring**	**2L Scoring**
Profit	18.59	20.85	36.45	29.2	37.62	-15.31	7.68	-7.93	-1.05	10.3	77.95
SP	0.26	0.31	0.6	0.47	0.62	-0.54	0.03	-1.37	-0.17	0.09	1.26
Max DD	-16.09	-32.67	-26.21	-16.36	-22.86	-25.37	-25.16	-10.33	-40.9	-23.42	-12.54
ACC	49.1	48.70	49.3	50.1	49.5	49.9	51.7	61.48	51.1	50.5	57.49
Stdev	0.211	0.211	0.211	0.211	0.211	0.211	0.211	0.052	0.211	0.206	0.207
GOOG											
Profit	45.71	48.67	38.18	62.6	-38.66	-6.19	58.52	-5.08	13.8	-20.41	73.32
SP	0.74	0.79	0.62	1.0	-1.29	-0.29	0.94	-0.88	0.16	-0.7	1.20
Max DD	-18.72	-15.49	-18.06	-17.33	-43.13	-27.03	-20.18	-11.89	-21.17	-36.43	-12.70
ACC	50.1	50.50	50.1	55.27	43.34	52.49	54.08	58.25	50.89	49.11	58.25
Stdev	0.213	0.213	0.123	0.213	0.213	0.213	0.213	0.063	0.213	0.208	0.204
MACF											
Profit	26.96	-1.83	-4.17	27.04	-3.2	-22.95	-9.32	20.35	30.5	-15.24	103.61
SP	0.65	-0.15	-0.23	0.66	-0.2	-0.91	-0.4	1.41	0.74	-0.63	2.19
Max DD	-28.06	-36.28	-35.7	-25.01	-31.19	-45.66	-32.04	-4.71	-21.72	-27.98	-14.92
ACC	49.81	52.92	52.92	53.31	52.14	50.19	49.81	68.09	52.53	52.92	62.26
Stdev	0.312	0.313	0.313	0.312	0.313	0.312	0.313	0.108	0.312	0.307	0.304
MCD											
Profit	3.97	1.78	1.78	-8.68	13.71	7.71	-1.06	-0.49	-14.11	2.06	19.70
SP	0.08	0.17	0.17	-0.61	0.28	0.06	-0.29	-0.61	-0.86	0.16	0.49
Max DD	-14.61	-14.61	-14.61	-19.99	-15.09	-12.72	-21.31	-7.9	-22.84	-13.93	-12.02

Continued on next page

Table 5.3 – *Continued from previous page*

Data/Model	B&H	Poly	Linear	SVR (rbf)	SVR (linear)	SVR (poly)	Random Forest	Average	UCB	1L Scoring	2L Scoring
ACC	54.6	54.2	54.2	51.6	54.4	54.8	48.6	63.0	49.2	54.8	57.8
Stdev	0.124	0.124	0.124	0.124	0.124	0.124	0.124	0.053	0.124	0.123	0.123
MKS											
Profit	25.11	-18.61	-34.81	27.8	-23.07	16.1	57.13	0.1	-38.46	-32.89	19.42
SP	0.35	-0.56	-1.04	0.39	-0.68	0.19	0.83	0.46	-1.16	-0.99	0.26
Max DD	-25.24	-29.86	-43.69	-18.99	-31.52	-30.7	-25.2	-8.44	-41.19	-44.08	-25.15
ACC	48.8	48.6	47.8	51.01	47.4	50.4	52.2	63.4	48.2	52.6	58.0
Stdev	0.237	0.237	0.237	0.237	0.237	0.237	0.237	0.065	0.237	0.233	0.231
OXIG											
Profit	-15.0	21.61	19.87	-9.08	-11.01	24.96	27.57	19.14	70.51	-53.27	320.38
SP	-0.33	0.20	0.18	-0.23	-0.26	0.24	0.27	0.38	0.71	-1.24	2.13
Max DD	-46.91	-27.29	-28.09	-49.02	-39.94	-34.86	-30.3	-16.83	-24.84	-64.66	-17.76
ACC	48.07	50.10	49.69	51.12	-47.86	52.75	49.69	56.42	52.55	50.71	58.04
Stdev	0.346	0.346	0.346	0.346	0.346	0.346	0.346	0.158	0.346	0.339	0.332
SXS											
Profit	0.67	-45.88	-48.22	-21.66	-38.85	-12.51	-25.86	1.43	49.65	13.24	83.04
SP	0.1	-1.30	-1.38	-0.58	-1.06	-0.37	-0.69	0.24	0.66	0.13	1.09
Max DD	-35.85	-52.22	-54.28	-38.7	-47.7	-35.28	-39.48	-12.55	-19.41	-24.17	-21.91
ACC	51.7	47.29	46.69	48.7	48.1	46.49	49.1	58.72	53.31	59.91	56.91
Stdev	0.263	0.262	0.263	0.263	0.262	0.263	0.263	0.093	0.263	0.256	0.253
TSCO											

Continued on next page

Table 5.3 – *Continued from previous page*

Data/Model	**B&H**	**Poly**	**Linear**	**SVR (rbf)**	**SVR (linear)**	**SVR (poly)**	**Random Forest**	**Average**	**UCB**	**1L Scoring**	**2L Scoring**
Profit	-44.86	-17.42	-21.5	-14.4	-22.92	-19.75	-4.71	-5.93	-32.43	50.55	82.45
SP	-1.35	-0.52	-0.62	-0.44	-0.66	-0.58	-0.22	-0.58	-0.93	0.73	1.13
Max DD	-57.5	-29.66	-33.14	-30.61	-34.56	-39.32	-32.85	-17.54	-42.32	-25.57	-20.74
ACC	46.91	46.91	46.71	46.91	47.5	49.7	46.91	56.49	49.1	54.89	58.88
Stdev	0.244	0.244	0.244	0.244	0.244	0.244	0.244	0.104	0.244	0.239	0.242

Table 5.3 shows the comparison of the 11 trading strategies' performances, tested on 11 stocks. These performances were evaluated using the profit, Sharpe Ratio, maximum drawdown, accuracy and standard deviation (more detail can be found in section 2.3.4 on page 42). The profit, maximum drawdown and accuracy are shown as percentage values. The following is an example result.

For the stock CARR, the maximum profit of 264.52% was obtained by the two-layer (2L) scoring system. Note that this profit was computed over the two years. Therefore, it is about 91.04% yearly after taking the compounding return into consideration (calculating from $(1+2.65)^{1/2}-1$). The maximum Sharpe Ratio for CARR, 2.33, also came from the 2L scoring system. As this stock trended up during the testing period (as can be noticed by the 62.38% profit of the buy & hold), the individual machine learning predictors seemed to perform well too. Support Vector Regression (SVR) with linear kernel is the best algorithm out of all six individual algorithms, providing 214% profit with a Sharpe Ratio of 1.96. The other predictors also performed well on this stock, except for SVR with rbf kernel. The SVR with rbf kernel performed really badly, ending up with -28.57% loss. It also ends up with the worst maximum drawdown at -41.02%, meaning an investor would have to bear with losing almost 50% during this trading period. Note that 100% profit is needed to recover to break-even from a 50% loss. Results such as this justify our use of multiple predictors, since they demonstrate that if the wrong predictor is used in the case of a single-predictor system, large losses could result or traders may abandon the system altogether.

As for the ensemble-based strategies (average, UCB, 1L and 2L scoring strategies), the average and 1L scoring strategies also ended up profitable in addition to the 2L scoring system, but their profits were small. The average model returned 5.55% profit while the 1L scoring strategy returned 20.72%. However, the average strategy has a negative Sharpe Ratio of -0.01 in spite of it making a profit. This indicates that the profit did not justify the risk taken to earn it.

As for the accuracy, the maximum accuracy was 60.39%, which was obtained by the 2L scoring strategy. The second highest accuracy was 57.39%, from the averaging model. As we discussed, ending with only 5.55% profit and a -0.01 Sharpe Ratio indicates that the averaging strategy is not profitable enough to be worth investing. But it came second place in terms of accuracy. This is not surprising since we already recognised that greater accuracy does not translate into greater profits, or only does so above a high threshold.

We ran a paired T-test on the Sharpe ratios of these eleven datasets to compare between our scoring system and the other individual strategies. We obtained the p-values between our scoring system (2L scoring system) verses the buy & hold benchmark, polynomial regression, linear regression, svr (rbf), svr(linear), svr(polynomial), random forest, Averaging, UCB and the 1L scoring system, which all are very significant, p-values are less below 0.005. With only eleven pairs, each individual result may not tell us much, although the values are very promising, all sitting above the 95% confidence level. Taken together, these support that our ensemble approach leads to a meaningful improvement in the Sharpe Ratio above the 95% confidence level over the benchmark and individual machine learning algorithms. The details of these tests can be seen in appendix E on page 336.

5.4.1 Discussion

What follows are some conclusions we have drawn from our results:

1. **No best individual predictor**: From these results, it is not possible to select an individual predictor which could perform well with all stocks at all times. For example, polynomial regression performs very well, beating the buy & hold strategy for COST, FERG, GOOG and OXIG stocks. However, for D4T4, MACF, MKS and SXS it did not perform well at all. SVR with linear kernel works really well on COST, giving more than 100% profit, but did very badly on GOOG, ending up with -38.66% while the buy & hold profited 45.71%. Similar situations applied to all the individual machine learning predictors tested (polynomial regression, linear regression, SVR with rbf, linear and polynomial kernel, and random forest). Each predictor seemed to do well on specific stocks or at specific times, the reasons for which are hard to identify. Nonetheless, such results validate our theory that the characteristics of financial market data changes through time and across stocks, which underpins this research and our belief that an adaptive trading system is needed to be consistently profitable. However, changes in the characteristics are more complex than switching between high and low volatility, necessitating the inclusion of multiple machine learning algorithms to increase our chance of noticing changes and making better predictions.

   Another point to consider here is the similarity of the results from polynomial and linear regressions. It can be seen that for most stocks, they provide the same or very similar results. This raises the question of whether one of them is redundant and can be removed. The answer here is no, because in this research we performed clustering on the training data and created multiples of these predictors specialising in different characteristics of the data. Although their results in the testing period used in this research were similar, it is possible for the results to diverge when the system is used in the future or on different stocks, and therefore it is worth keeping them in our system.

2. **Performance of 2L scoring strategy**: The best strategy here is the 2L scoring system. It performed better than the buy & hold in every dataset except MKS. This stock seemed to be difficult for most of the predictors, as the price fluctuated a lot. There were also several big drawdown periods which most predictors failed to cope with. Apart from MKS, the 2L scoring system strategy works really well, ending up with a relatively high profit and Sharpe Ratio.

   Additionally, the 2L scoring system also performed well in terms of drawdown. For most of the stocks, it provided relatively small drawdowns compared to the other models, except for the averaging model, for example in D4T4, FERG, GOOG, MACF, MCD, MKS, OXIG and TSCO. This means that not only does it offer investors a high profit, but it is also safer, as a smaller drawdown means investors do not need to endure big losses during the trading period. It is important when developing a trading system that investors can stick with it and do not abandon it, which large drawdowns may cause them to do. It is well-known that losing money is more psychologically painful than gaining the same amount, and that earning a profit then losing most of it back can be difficult even when the investor still ends up with more than they started with.

3. **Accuracy does not guarantee profit**: We discussed that when the averaging strategy was applied in the case of CARR it ended up with higher accuracy but much lower profit than SVR with rbf kernel predictor. This was not the only case. There were many times when this situation happened in our experiment. For example, in the results from MACF, the averaging strategy ended up with much higher accuracy (68.09%) than that of SVR with rbf kernel (53.31%), UBC (52.53%) and the 2L scoring (62.26%). However, the profit it made was much lower than the others. It only provided 20.35% profit while the SVR, UBC and 2L scoring strategies gave 27.04%, 30.5% and 103.61%, respectively. Another example of this situation is SXS. The accuracy of the 1L strategy (59.91%) is greater than the accuracy of the 2L scoring strategy (56.91%), but the profit of the 2L strategy (83.04%) is much greater than the 1L strategy (13.24%). There are many results where this happened besides the aforementioned examples.

   In financial trading, it is not enough to get the direction correct only, but to get it correct on high-profit days. For example, if we get the direction correct on a day which only gives us £1 profit at the end, it is not even worth the trading fee that the investor needs to pay. Therefore, it is not about the number of times you predict the direction correctly, it is about whether you get it correct on the right days too. Getting the direction correct 10 times and each time only making £(1) profit is worse than getting it correct only one time and making £20. The distribution of financial market returns is such that missing high-profit periods can have a severe impact on results, since just a few such periods can account for most of the available profit. This is the so-called opportunity cost. For this reason and what we have observed in our results, we are not going to take accuracy to be the main criteria to evaluate our results. In this research, we consider Sharpe Ratio to be the superior evaluation metric as it takes both profit and risk into account (more details of the Sharpe Ratio can be seen in section 2.3.4 on page 42).

4. **Averaging strategy with high accuracy but low profit**: Another interesting point that can be noticed is the high accuracy of the averaging strategy (but low profit, as mentioned earlier). There are two further observations worth addressing here. Firstly, it provided the lowest standard deviation among all strategies. This makes sense as it averages the answers from all predictors, leading to low variance. In finance, this means less risk to invest with this strategy as the chance of large negative deviations is small. This is also demonstrated by its having the smallest maximum drawdown. However, saving an investor from big losses does not mean a strategy is worth investing with (since big losses are guaranteed to be avoided if one does not invest at all). In this case, usually ended up making only a small profit compared to the other strategies, typically between 5 to 20 percent. Sometimes it was not profitable at all; there are four stocks out of 11 for which it ended up with losses.

   Another point to consider is that this method acted very similarly to majority voting. Because this strategy averaged the answer from every predictor, which all predicted either 1, -1 or 0, the average answer and the majority vote are pretty much the same. Although we did not perform an experiment using majority voting to select features, we can still confirm that this method would not be better than the 2L scoring system because the results would closely resemble the averaging strategy.

   Finally, the fact that the averaging model achieved mostly higher accuracy than the other models - in fact the highest for 6 stocks out of 11: COST, FERG, GOOG, MACF, MCD and MKS - means that the majority of our predictors provided correct answers most of time. If we were working on other problem spaces, this would be very good. However, as we are working in the financial area, we primarily want to achieve high profit and low risk, not high accuracy. As discussed earlier, high accuracy and profit are not the same. Even though the averaging model was able to provide high accuracy, it did not make much profit and performed poorly for some stocks in terms of profit and Sharpe Ratio, such as FERG, GOOG, MCD, MKS and TSCO.

From the experiments and discussion presented, we conclude the 2L scoring strategy is the best model we evaluated as it provided superior and consistent results in terms of Sharpe Ratio, profit and drawdown. We will take this design to use in our trading system.

Experiments one to three developed the essential parts of our scoring system, which is central to this research. These experiments are presented in order and it is recommended readers follow them from the beginning of experiment one to the end of experiment three in order to understand the rationale and design of our scoring system. All datasets used in these experiments spanned the same period of time to allow for comparison and to identify the best solution for our system. The next experiment, experiment four, is not as essential as the previous three experiments. Experiment four discusses a few different system adjustments and comprises a number of small experiments which focus on different parts of our trading system. These experiments are not in any order and the datasets used in each are not always the same. However, these issues do not affect the heart of our system, which is the scoring mechanism.

5.5 Experiment 4: Other System Adjustments

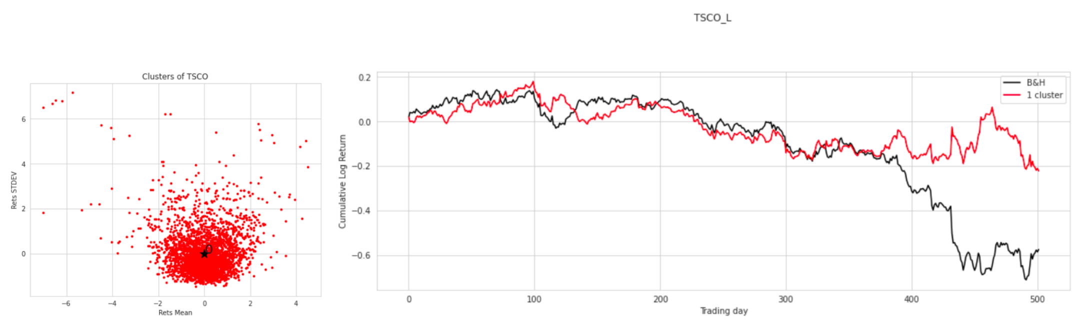
This experiment is separate from the three main experiments. It describes some short experiments conducted to test adjustments to the system, such as how the final signal is created or how signals can be optimised. These experiments aimed to improve the performance of the system without affecting the scoring mechanism of our trading system (the scoring system). We also don't expect their improvements to be as substantial.

5.5.1 Number of Clusters

This section investigates how the number of clusters affects the system. Although satisfied with the results in table 5.3 on page 187, we would like to investigate further the effects of the number of clusters. Throughout this chapter, we used two clusters, representing low and high volatility, and the results we achieved were good. Therefore, this experiment only aims to improve upon this result if possible.

This experiment increases the number of clusters one by one to see whether the result is affected. In this section, we will only demonstrate some of the results. At the end of experiment 4, all the results will be shown and compared, followed by a discussion. Firstly, we take TSCO as an example and perform the experiment using an individual machine learning algorithm: polynomial regression. The reason we decided to show the result from an individual predictor is firstly, that it is easier to understand the effect of the number of clusters and, secondly, we would like to see whether the effects will hold for different machine learning predictors. By making individual predictors work better, the ensemble result will be better as well, given we have already established the benefit of our scoring system in the first three experiments.

The number of clusters starts from one, which is equivalent to not performing clustering and using only one single predictor. The TSCO result for the single predictor (polynomial regression) strategy was shown in table 5.2 on page 153. However, the reader can see the results again here. The performance of the single predictor and cluster for TSCO stock, tested over a period of two years (2013 - 2014) can be seen in figure 5-36.



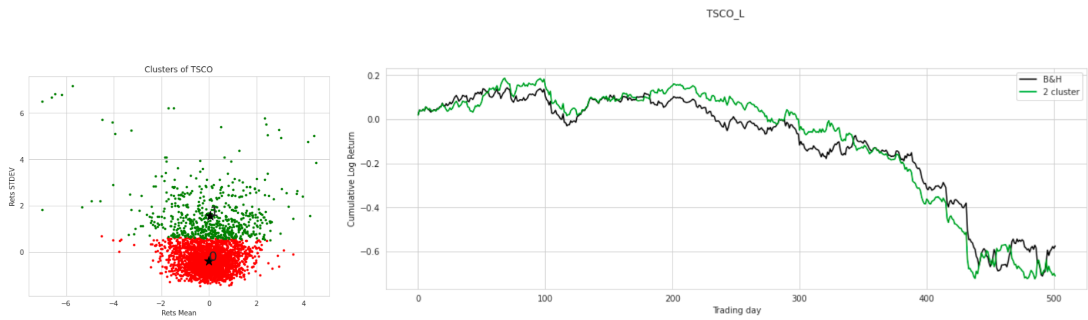
(a) One cluster (b) Result of one cluster

Figure 5-36: Example of TSCO with one cluster

Figure 5-36 shows the result on TSCO stock with only 1 cluster. This result is the same as shown in 5.2 on page 147, because it is the result from having only one predictor.

This result comprises two sub-figures, sub-figure (a) shows the cluster result and (b) shows the trading performance according to the cluster in (a). The black and red graphs in (b) represent the cumulative profit from the buy & hold and single cluster strategies, respectively. With only one cluster, even though the result (red) is better than the buy & hold (black) benchmark strategy, TSCO still ends up at a -17.42% loss. The Sharpe Ratio is a poor -0.52. Not only are these two key values bad, but the drawdown is also very large at -29.66%. These figures suggest that trading this stock with only one cluster does not lead to a good outcome.

This is a problem that we found out in experiment 2 (5.2 on page 147), and we tried to improve the result by using 2 clusters. However, it did not help much (as can be seen in 5.2 on page 153). For the reader's convenience, we show the result with two clusters again in figure 5-37.

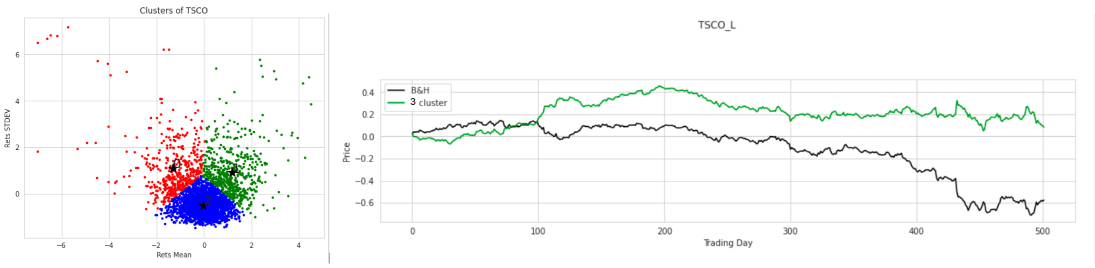


(a) Two cluster (b) Result of two clusters

Figure 5-37: Example of TSCO with two clusters

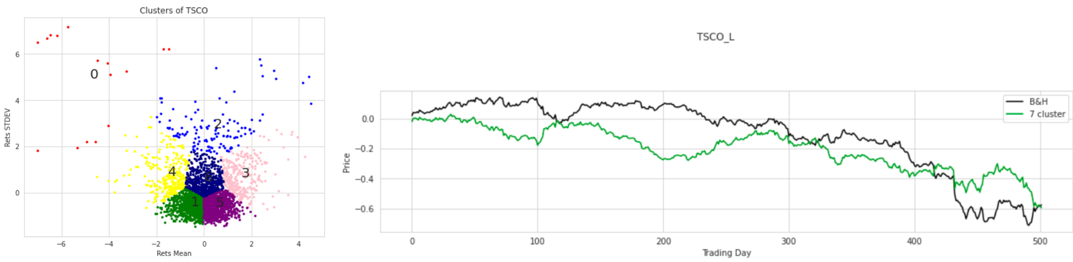
In figure 5-37, the black and green graphs in sub-figure (b) represent the results from the buy & hold and two cluster strategies respectively. As can be seen, TSCO performs even worse overall, finishing with a -51.88% loss. The Sharpe Ratio is bad at -1.64. Once again, the drawdown is very low at -59.82%. This means that trading this stock with only two clusters is also far from optimal. Investors not only end up with a big loss, but have to endure losing more than half of their capital during the trading period. Therefore, for these results, it can be said that increasing the number of clusters to two does not help.

We also identified this problem in experiment 2, where we found two stocks, TSCO and FERG, for which having two clusters did not help when using the polynomial regression predictor. In that experiment, we concluded that we could solve the problem by adding more machine learning predictors and having them work together. The solution works well once the right method to manage the cooperation of multiple machine learning predictors had been identified, as shown in the results after finishing experiment 3 (5.3 on page 187). Although satisfied with the results of multiple predictors and the scoring system, we acknowledge that the increasing the number of clusters may also have improved the performance, which is why we have included this additional experiment.



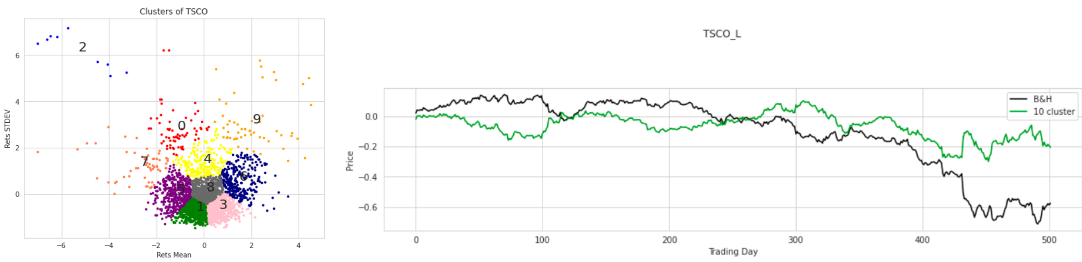
(a) Three cluster (b) Result of three clusters

Figure 5-38: Example of TSCO with three clusters



(a) Seven cluster (b) Result of seven clusters

Figure 5-39: Example of TSCO with seven clusters



(a) Ten cluster (b) Result of ten clusters

Figure 5-40: Example of TSCO with ten clusters

Figure 5-38 to Figure 5-40 represent the results from TSCO using 3, 7 and 10 clusters, respectively. In each figure, sub-figure (a) and (b) shows the clusters of data and the result. From these three different clusters' results in all sub-figures (b), it can be seen that once the number of clusters increases to three, the result is better than with one and two clusters. The loss is gets smaller, from -51.88% (at cluster = 2) to

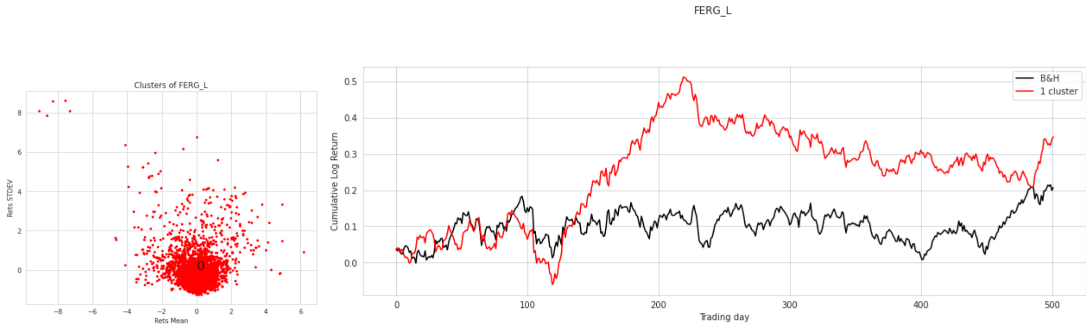
-2.93%. The Sharpe Ratio also increases from -1.64 to -0.18. The drawdown is also much better, shrinking from -59.82% to -35.72%. This means that once the number of clusters increases to three (therefore the number of predictors is equal to 3), polynomial regression works much better for TSCO stock. Not only would an investor lose less money, the investor would also encounter less stress from the maximum loss of about 35% of their funds rather than 59.82% during the trading period.

The result above matches our expectation that as the number of predictors increases, the more likely a predictor will be good at the specific characteristics of that data will emerge. One might think of the analogy that more specialist doctors might provide a more accurate diagnosis and better course of treatment. With this rationale, we experimented further with even more predictors.

However, the result did not show a linear relationship between a greater number of clusters and performance, as can be seen from figure 5-39. Once the number of clusters reaches seven, the results get worse, even worse than buy & hold. The cumulative profit of having 7 predictors (clusters) is -43.74%, while the buy & hold is -44.86%. As can be seen from sub-figure (b), most of the time the cumulative profit of the 7-predictor strategy is lower than the buy & hold. Another example is when the number of clusters is 10 in figure 5-40. Even though the cumulative profit of the 10-predictor strategy (-16.96%) ends up much higher than the 7-predictor strategy (-43.74%), its result overall is still much worse than the 3-predictor strategy.

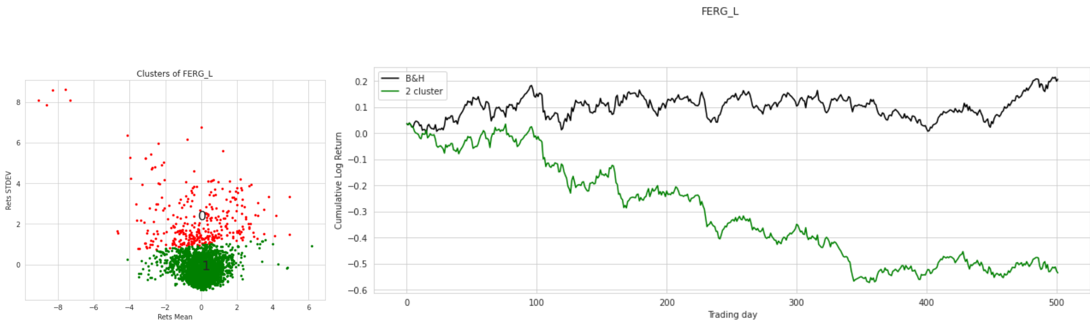
From the results above, it can be concluded that trading performance can be improved by a higher number of clusters, but only to a point. The results will get better when there are enough predictors that some can specialise in the characteristics of the data, confirming our original idea as discussed in experiment 1 and 2. For some stocks, two predictors are enough for the results to get better (the results can be seen in table 5.2 on page 153). On the other hand, two clusters or predictors are not enough to deal with all different characteristics in the data for some stocks, such as TSCO, which need at least three clusters. The fact that results are worse at seven and ten clusters follows the intuitive expectation that too many clusters results in overfitting - predictors are too specialised for the training data and unable to work on unseen data that does not resemble the past. However, we cannot draw too general conclusions after having presented only one stock's result. Below we present the result for FERG, because it is another stock for which the results from using two predictors were not good.

The results of FERG with 1, 2, 3, 7 and 10 clusters can be seen in figure 5-41 to 5-45, respectively.



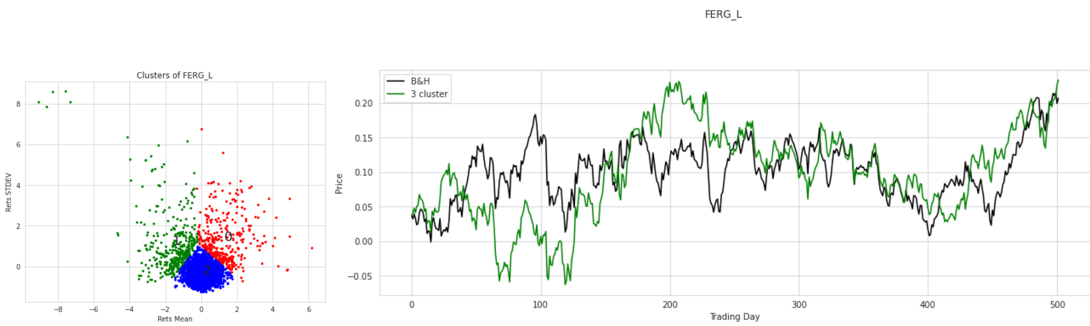
(a) One cluster (b) Result of One clusters

Figure 5-41: Example of FERG with one clusters



(a) Two cluster (b) Result of two clusters

Figure 5-42: Example of FERG with two clusters



(a) Three cluster (b) Result of three clusters

Figure 5-43: Example of FERG with three clusters

The results from FERG are similar to TSCO. The single cluster (single predictor) strategy performed much better than two clusters. While the single predictor provides 20.85% profit, the two cluster strategy ends up with a big loss of -43.56%. Consequently,

(a) Seven cluster (b) Result of Seven clusters

Figure 5-44: Example of FERG with seven clusters

(a) Ten cluster (b) Result of ten clusters

Figure 5-45: Example of FERG with ten clusters

the Sharpe Ratio of the single cluster (0.31) is much higher than the two clusters (-1.51). When the number of clusters is increased to 3, as shown in figure 5-43, it can be seen that the clusters help improve the cumulative return from -43.56% to 54.34%. The Sharpe Ratio also increases from -1.51 to 0.89. While it appears the conclusion can be drawn that more clusters lead to more profit, this is premature as the results are similar to TSCO and the profit and Sharpe ratio decline once the number of clusters increases to 7 and 10, as shown in figure 5-44 and 5-45. The profit and Sharpe Ratio of the 10-cluster strategy drop to -26.14% and -0.86, respectively.

It can be seen from the results of TSCO and FERG above that the performance of our ensemble predictor system can be improved by increasing the number of clusters but only up to a specific point. Once the number of clusters is too high, the performance starts to drop sharply, which we would expect as the final predictions become too fit to the past data. In the next section, we will experiment further to find the optimum number of clusters.

5.5.1.1 Experimental Results

In this section, we experiment further on the effect of clusters on our system's performance. We selected four stocks to be investigated. These four stocks are classified into two groups, as follows:

- Group 1: Stocks for which the two-cluster strategy did not work very well in experiment 2 (results in table 5.2 on page 153). We select these stocks in order to investigate whether a different number of clusters will improve the predictors' performances. Stocks selected for this group are FERG and TSCO.

- Group 2: Stocks for which the two-cluster strategy performed well in experiment 2 (results in table 5.2 on page 153). These stocks are selected to make sure that increasing the number of predictors will not decrease the performance. The stocks selected for this group are MACF and CARR.

The visualisations above result from polynomial regression only. In this section, we experiment further with all 6 machine learning models used in our system to ensure that the effect of the number of clusters holds for the other algorithms. Firstly, we will present visualisations of some results, as can be seen in figure 5-46.

Figure 5-46 shows the result for FERG. This experiment is set up the same as experiment 2, but instead of clustering the training set into two groups, we experimented with different numbers of clusters between 1 to 10. The objective of this experiment was to observe the effect of the number of clusters on our trading performance. The key value we used to evaluate the performance here was the Sharpe Ratio, which has been discussed in section 5.4.1 on page 192. This visualisation is composed of three graphs. The top graph shows the comparison of Sharpe Ratios when using different numbers of clusters and different algorithms, while the middle and bottom graphs show the average Sharpe Ratio and the standard deviation related to each number of clusters.

The Y-axis of the top graph shows the Sharpe Ratio and the X-axis shows the number of clusters from 1 to 10. Please note that when there is only one cluster, this is actually the single predictor strategy. Each line in the top graph shows the Sharpe Ratio from a machine learning model. There are six lines representing the six algorithms we have used. It can be seen that when the number of clusters is two, all the algorithms perform badly, which we expected from the results of experiment 2 (seen in table 5.2 on page 153). In this table, the trading performance of polynomial regression with two clusters drops to -43.56% and the results support that two clusters is inadequate for prediction of this stock's price. However, although all the predictors seemed to perform badly

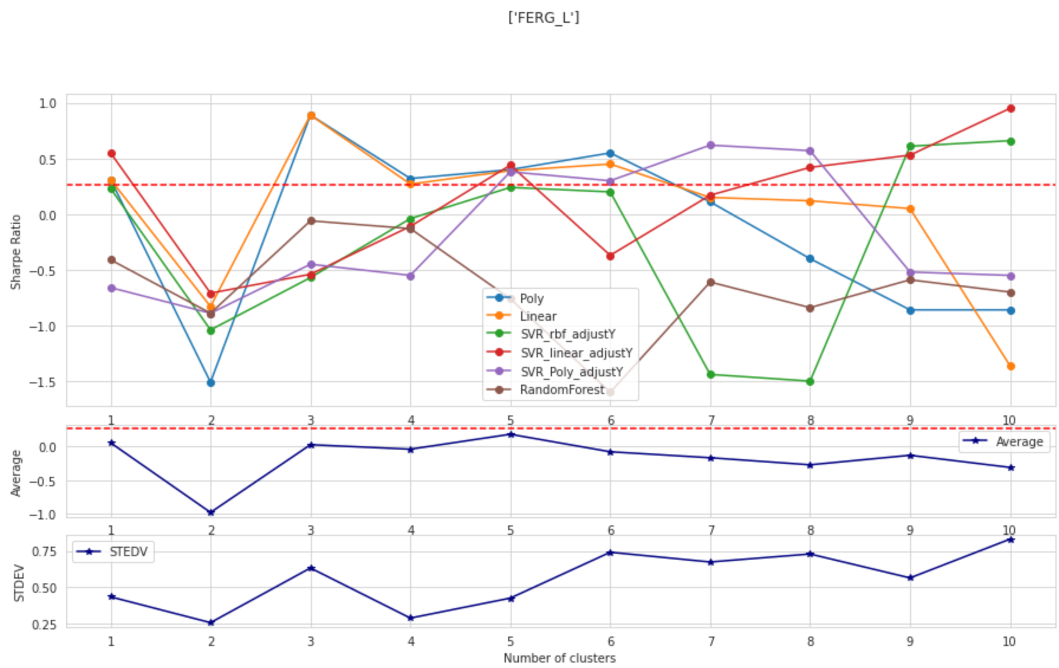


Figure 5-46: The comparison of Sharpe Ratio of six machine learning algorithms testing on FERG

individually, our scoring system managed very well and selected the correct predictors at the right times, resulting in 77.95% profit at the end (this can be seen in ref 5.3 on page 187).

Although the scoring system worked well on this stock with just two clusters, we decided to run the experiment with different numbers of clusters in order to investigate whether we can improve the system's performance by increasing clusters. From results in figure 5-46, it can be seen that with 3, 4 or 5 clusters the Sharpe Ratio is higher and does not change much among different algorithms. With 7, 8 and 10 clusters, the Sharpe Ratio from some algorithms - such as SVR with linear kernel - was high, but the differences between the Sharpe Ratios of different algorithms was very high too. It is difficult to decide from the top graph which cluster number should be best, but three clusters seems to be suggested.

In order to make a better decision, the average and standard deviation should be taken into account. The middle graph shows that the maximum average Sharpe Ratio (0.18) is obtained when the number of clusters is five. The second highest Sharpe Ratio is with three clusters. But when looking at the standard deviation in the bottom graph, five clusters results in a lower standard deviation. Therefore, five clusters seems to be the best number for this stock. But this may not be the case for other stocks, so we need to look at other results, for example the result from TSCO in figure 5-47.

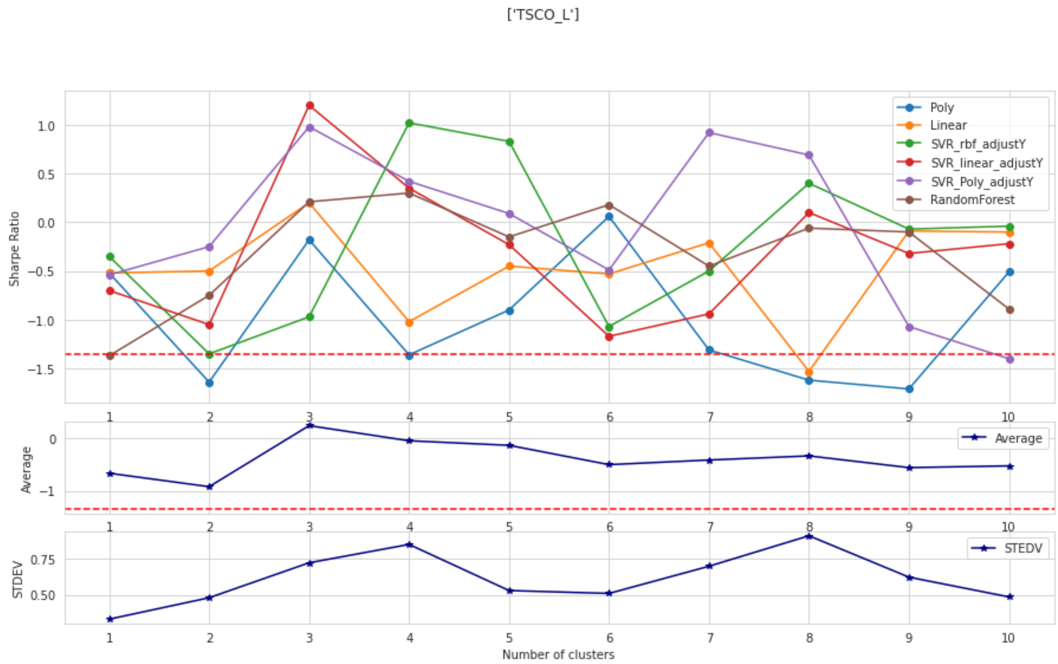


Figure 5-47: The comparison of Sharpe Ratio of six machine learning algorithms testing on TSCO

The results from TSCO in figure 5-47 have a few differences from FERG. Firstly, the variance of the Sharpe Ratio at three and four clusters is not as low as for FERG. The differences among predictors' Sharpe Ratios are bigger. However, this should not matter much because once the scoring system is used, the best predictors should be selected. Therefore, a high Sharpe Ratio is more important than low variance. Secondly, five clusters does not provide the maximum average Sharpe Ratio any more. The highest average Sharpe Ratio (0.24) is obtained by three clusters, with five providing the second highest. However, three clusters (with the maximum average Sharpe Ratio) result in a higher standard deviation than five clusters (the second highest average Sharpe Ratio). However, a high Sharpe Ratio is more important for our scoring system. For this reason, three clusters seems to be the best option for TSCO.

FERG and TSCO were selected to demonstrate the results of the experiment on stocks from group 1 (stocks for which two clusters did not perform well). Now, we will present an example from a stock in group 2 (for which 2 clusters resulted in good performance).

The example stock from this group is MACF, and the results are shown in figure 5-48.

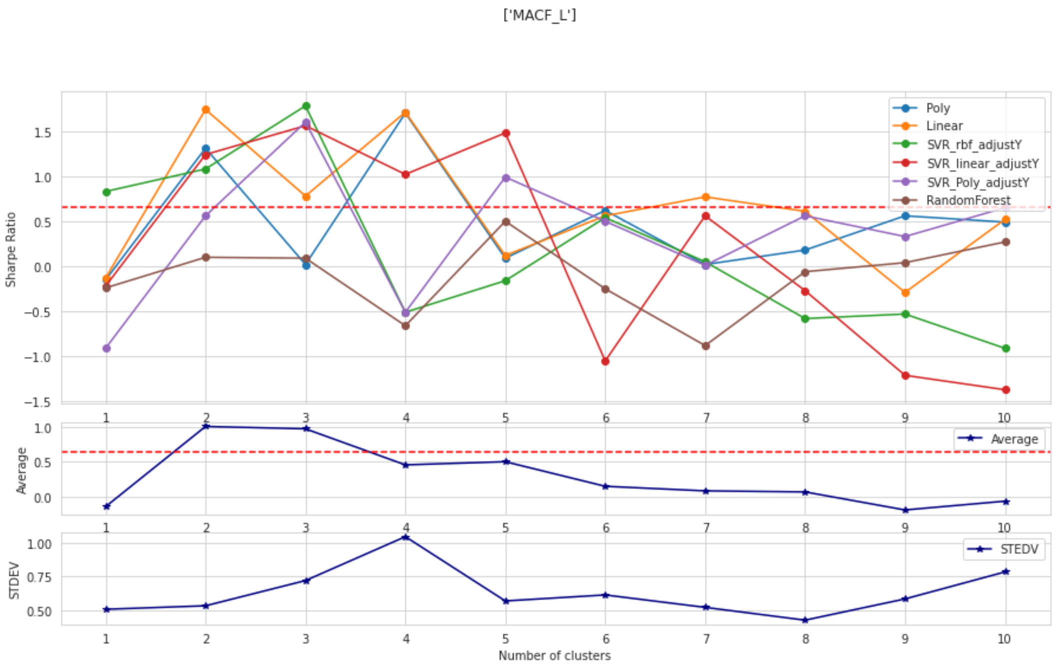


Figure 5-48: The comparison of Sharpe Ratio of six machine learning algorithms testing on MACF

The effect of increasing the number of clusters is clearer from the MACF results than FERG and TSCO. From five clusters upwards, the Sharpe Ratio drops. From experiment 2 (seen in table 5.2 on page 153) we know two clusters already works well for MACF when using polynomial regression, resulting in a high Sharpe Ratio of 1.31. After experimenting with other machine learning algorithms, it can be seen that most of them also work well with two clusters. Polynomial and linear regressions, as well as SVR (with rbf and linear kernel), provide good Sharpe Ratios higher than 1, especially linear regression which provides the maximum Sharpe Ratio of 1.74. The remaining algorithms, SVR with polynomial kernel and randomforest, did not produce a Sharpe Ratio as high as the aforementioned algorithms but their Sharpe Ratios were still greater than 0. Therefore, the average Sharpe Ratio for this share with two clusters is a good 1.005. Now, let us look further at the effect of more clusters.

With three clusters, the top graph shows that the performance was consistent for most of the algorithms that performed well with two clusters, except polynomial regression. Its performance drops sharply with three clusters. The Sharpe Ratio falls from 1.74 to 0.02. However, the average Sharpe Ratio is still high at 0.97, as the SVR with polynomial kernel works very well with three clusters. The middle graph shows that two and three clusters give the best Sharpe Ratios, with two performing a little better. Taken with the other results discussed, the results here further support the general idea that between three and five clusters is optimal.

From the results of the three stocks above, it seems between three and five clusters could improve the performance of our trading system. Therefore, we ran experiments on a further stock, CARR, and calculated the average and standard deviation for every algorithm which should help us decide how many clusters is most suitable. The results are shown in 5-49.

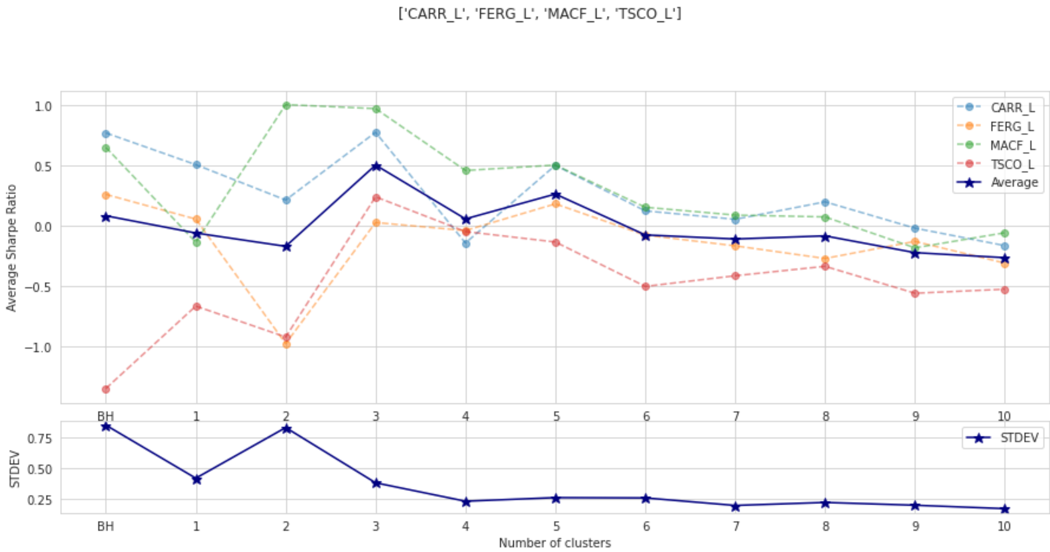


Figure 5-49: The average Sharpe Ratio from six machine learning algorithms of CARR, FERG, MACF and TSCO

The result from figure 5-49 is very helpful. It is much clearer to analyse the results from the average values for every algorithms. The top graph shows the average Sharpe Ratio for each stock. The four dashed lines represent the average Sharpe Ratio for every algorithm on four stocks: CARR (blue), FERG (yellow), MACF (green) and TSCO (orange). The bold navy line shows the average value from all stocks (all four dashed lines). The maximum Sharpe Ratio is obtained by three clusters (0.50), followed by five clusters (0.26).

The standard deviation seems to go down after three clusters. Accordingly, the Sharpe Ratios start to converge from four clusters upwards. This convergence could not be noticed earlier. Although three clusters provided the highest Sharpe Ratio, the standard deviation of this cluster is higher than with five clusters, and we must decide between including either three or five clusters or using both together. Finally, we decided to go three clusters for the following reasons: 1) We will put all the algorithms to work together using the scoring system, which is able to identify the best predictors. Therefore, we will focus on having predictors with high Sharpe Ratios ready for the scoring system to select. The variance should not be a problem for the scoring system, especially as the standard deviation of three and five clusters is similar like this. 2) We must consider time consumption. More clusters means more predictors for each machine learning algorithm. For now, we only have six algorithms and none of them take a lot of time to process. However, our system is designed to be able to add new algorithms. The processing time can increase vastly, especially if time-consuming algorithms are added such as neural networks. More clusters would only increase time consumption further, with little to gain.

On the subject of time consumption, we will also discuss the time our system takes at the moment. Since the machine learning algorithms used in our current model are not time-consuming ones, they work very quickly. The average processing times of our model at each step can be seen in table 5.4.

Table 5.4: The average processing times (in second) of our system at different designs

Cluster	Step 1 Preparation	Step 2 Clustering	Step 3 Training	Step 4 Development	Step 5 Simulation	Total Total
1 Cluster						
1 algorithm	0.0208	0.0000	0.0202	0.1024	52.8232	52.9666
2 algorithms	0.0208	0.0000	0.0266	0.2014	52.8232	53.0720
3 algorithms	0.0208	0.0000	0.2521	0.1157	52.8232	53.2117
4 algorithms	0.0208	0.0000	0.4402	0.3072	52.8232	53.5914
5 algorithms	0.0208	0.0000	0.8443	0.2980	52.8232	53.9863
6 algorithms	0.0208	0.0000	1.7259	0.3120	52.8232	54.8819
2 Clusters						
1 algorithm	0.0208	0.7868	0.0404	10.7147	52.8232	64.3858
2 algorithms	0.0208	0.7868	0.0533	21.4293	52.8232	75.1134
3 algorithms	0.0208	0.7868	0.5041	32.1440	52.8232	86.2789
4 algorithms	0.0208	0.7868	0.8805	42.8587	52.8232	97.3699
5 algorithms	0.0208	0.7868	1.6886	53.5733	52.8232	108.8927
6 algorithms	0.0208	0.7868	3.4518	64.2880	52.8232	121.3706
3 Clusters						
1 algorithm	0.0208	0.9925	0.0606	16.0720	52.8232	69.9691
2 algorithms	0.0208	0.9925	0.0799	32.1440	52.8232	86.0604
3 algorithms	0.0208	0.9925	0.7562	48.2160	52.8232	102.8086
4 algorithms	0.0208	0.9925	1.3207	64.2880	52.8232	119.4451
5 algorithms	0.0208	0.9925	2.5329	80.3600	52.8232	136.7293
6 algorithms	0.0208	0.9925	5.1777	96.4320	52.8232	155.4462
3 Clusters						
1 algorithm	0.0208	1.2409	0.1010	26.7867	52.8232	80.9725
2 algorithms	0.0208	1.2409	0.1332	53.5733	52.8232	107.7914
3 algorithms	0.0208	1.2409	1.2603	80.3600	52.8232	135.7052
4 algorithms	0.0208	1.2409	2.2012	107.1467	52.8232	163.4327
5 algorithms	0.0208	1.2409	4.2214	133.9333	52.8232	192.2396
6 algorithms	0.0208	1.2409	10.3555	160.7200	52.8232	225.1604

Table 5.4 shows the time usage for different conditions in our model. All times are shown in seconds and separated into steps 1 through 5, according to the design in figure 4-1 on page 89. Time usage was measured from a single computer with Core

i7-8750H and CPU 2.20GHz * 12. Time usage is shown for different numbers of clusters and numbers of machine learning algorithms used in the experiment. The numbers of clusters used were 1, 2, 3 and 5. For each case, we measured the time with different numbers of machine learning algorithms involved. The number of algorithms ranged from 1 to 6.

The time usage in step 1, data preparation, does not depend on the number of clusters or machine learning algorithms. Therefore, this is almost the same in every case and only varies a little between datasets. In this table, we measure the data preparation times for three different stocks (TSCO, CARR and MKS). Each stock was tested 5 times. The average times for step 1 for TSCO, CARR and MKS were 0.0185, 0.0230 and 0.0207, respectively. The standard deviations are very low: 0.0009, 0.0095 and 0.0047. All time usage values in this table are averaged over these three stocks, 5 runs on each stock. The standard deviations are very low. Consequently, we are not going to go into detail about all the values. Instead, we will focus on the average time usage shown in table 5.4.

Please note that steps 1, 2, 3, 4 and 5 are data preparation, clustering, training, system development and trading simulation & evaluation, respectively. For 1 cluster, we don't need step 2 as there is no clustering to perform. As for step 3, training, it can be seen that each machine learning algorithm works quickly. For example, a polynomial regression only takes 0.0202 seconds to train. When adding another algorithm (linear regression) into the model, the time only increased a little to 0.0266 seconds. The maximum time taken for this step was only 1.7259 seconds when all six algorithm were involved. Time usage in step 4 is very quick too. The longest time taken is by step 5, trading simulation & evaluation. However, there is only one process within step 5 that takes long to run, which is the signal optimisation. Signal optimisation performs grid search in to order find the optimal point to filter the trading signals, which is time consuming. The average time taken for optimisation is still only under a minute, and the time taken is not affected by the number of clusters or predictors. Therefore, for 1 cluster, the maximum possible time usage is only 54.8819 seconds when all 6 algorithms are included.

The time usage for other numbers of clusters can be seen in the table. The processing time is not long at all as we have deliberately chosen machine learning models which are not time consuming. The maximum time, when using 5 clusters and 6 algorithms, is under 4 minutes (225.1604 seconds). However, our system is designed to be able to add new algorithms of interest. Therefore, the time could increase depending on the choice of algorithms and the number of clusters. Therefore, it would be better not to have too many clusters. On balance, we selected three clusters, which allows the model to provide good results - as can be seen in table 5.5 - while running quickly and with scope to add more algorithms.

Table 5.5: The comparison between results from the buy & hold strategy and our system with two, three and five clusters, evaluated over two years (2013-2014)

Data/Model	**B&H**	**2 clusters**	**3 clusters**	**5 clusters**
CARR				
Profit	62.38	264.52	243.43	211.20
SP	0.77	2.33	2.18	1.99
Max DD	-26.32	-12.16	-25.75	-15.83
ACC	52.89	60.39	59.10	58.67
Stdev	0.302	0.286	0.292	0.293
FERG				
Profit	18.59	77.95	110.78	47.27
SP	0.26	1.26	1.66	0.81
Max DD	-16.09	-12.54	-13.54	-19.75
ACC	49.1	57.49	56.69	59.28
Stdev	0.211	0.207	0.208	0.203
MACF				
Profit	26.96	103.61	74.13	44.35
SP	0.65	2.19	1.65	1.06
Max DD	-28.06	-14.92	-19.20	-21.43
ACC	49.81	62.26	59.5	56.03
Stdev	0.312	0.304	0.310	0.312
TSCO				
Profit	-44.86	82.45	127.60	88.66
SP	-1.35	1.13	1.60	1.19
Max DD	-57.5	-20.74	-16.11	-22.38
ACC	46.91	58.88	59.48	57.09
Stdev	0.244	0.242	0.240	0.243

Table 5.5 compares the results from the buy & hold strategy and our system with two, three and five clusters. It can be seen clearly that our system performs better than buy & hold with any number of clusters. The best cluster number for CARR, FERG

and TSCO seems to be three. With three clusters (18 predictors in total), our system provides the highest Sharpe Ratio of 2.18, 1.66 and 1.60 for CARR, FERG and TSCO, respectively. However, three is not the best number of clusters for MACF. It provides a good Sharpe Ratio of 1.65, but two clusters resulted in a Sharpe Ratio of 2.19. As this is still considered good, and three clusters seemed to work better for the other stocks, we decided to use three clusters in our model. Please note that this is the result before adding other factors, such as the brokerage fee. Once other factors are taken into account, these results may change, which we will discuss in the next experiment.

5.5.1.2 Discussion

This section investigated the number of clusters with a view to improving the performance of our system. For some stocks such as FERG and TSCO, having two clusters is not enough, and so we designed an experiment to add more clusters. Please note that even though two clusters did not work very well for TSCO and FERG, our scoring system was still able to cope well with them, as can be seen in experiment 3. Therefore, without increasing number of clusters, our model was still good overall. However, researching different numbers of clusters might be helpful and improve our system's performance. To perform this experiment, we set every condition up in the same way as experiment 3, except for the number of clusters. The number of clusters in this experiment ranged from 1 to 10. The key metric to evaluate our system's performance was the Sharpe Ratio. After running the experiment with different numbers of clusters, we found that the Sharpe Ratio tends to increase with the number of clusters, but only to a certain point after which it starts to decline or varies more widely between stocks. This is partly because when the number of clusters is high, a lot more predictors are involved. For example, if we have 10 clusters and 6 machine learning algorithms, there will be 60 different predictors in total. This could be very confusing for both the scoring system and the predictors, as can be seen in our results. We have also discussed how creating too many clusters may result in overfitting - predictors become too specialised to the past data on which they have been trained. After careful consideration of our results, we decided to go with three clusters, which seems to provide the best result in terms of having a high Sharpe Ratio, low standard deviation, and low time consumption.

The results obtained from this experiment are very good but we have not yet concluded the development of our trading model. The results up until this point lack a degree of realism since we are yet to consider other factors, such as trading fees. Therefore, in order to simulate our trades as realistically as possible, the next section will include such real-world factors in the investigation.

5.5.2 Signal Optimisation

In all previous experiments, we followed every signal the trading system provided without considering factors such as the starting capital (initial funds) an investor has to trade with, and trading fees which must be paid. Therefore, the results for some stocks may show unrealistically high profits, so we expect some changes to the results when we account for these factors. For example, our system provided a very good profit of 243% over two years (2013-2014) for CARR stock when using three clusters. The Sharpe Ratio was also good (2.18). These results can be seen in table 5.5 on page 210. The details of these trades compared to the buy & hold strategy can be seen in table 5.6 below:

Performance	Buy & Hold strategy	Our model
Sharpe Ratio	0.77	2.18
Profit (%)	62.38	243.43
Maximum Drawdown	-26.32	-25.75
Stdev	0.302	0.292
Number of Trades	2	202

Table 5.6: Performances summary of CARR comparing between our model and the buy & hold strategy during two years testing period (2013-2014)

From table 5.6, the number of trades made by our model is much higher than the buy & hold, which simply buys at the beginning of the period and sells at the end. If the investors followed every signal our model provided, there would be 202 trades (which means positions are opened or closed every other day), which is a high number over two years considering our focus is on retail traders. The details of these trades are shown in figure 5-50.

Figure 5-50 shows all trades our model made for CARR during 2013-2014. This figure relates to the results of CARR in table 5.6. Please note that in this figure, the cumulative return is shown in logarithmic scale but the profit in table 5.6 is scaled normally. The top graph compares the cumulative logarithmic return of our model (green) and the buy & hold strategy (black). Our model shows a much better result than the buy & hold. Details can be seen in table 5.6. However, the number of trades our model made was much higher than the buy & hold. The signals from our model are shown in the bottom graph. It provides and switches between trading signals (-1, 1 and 0) very

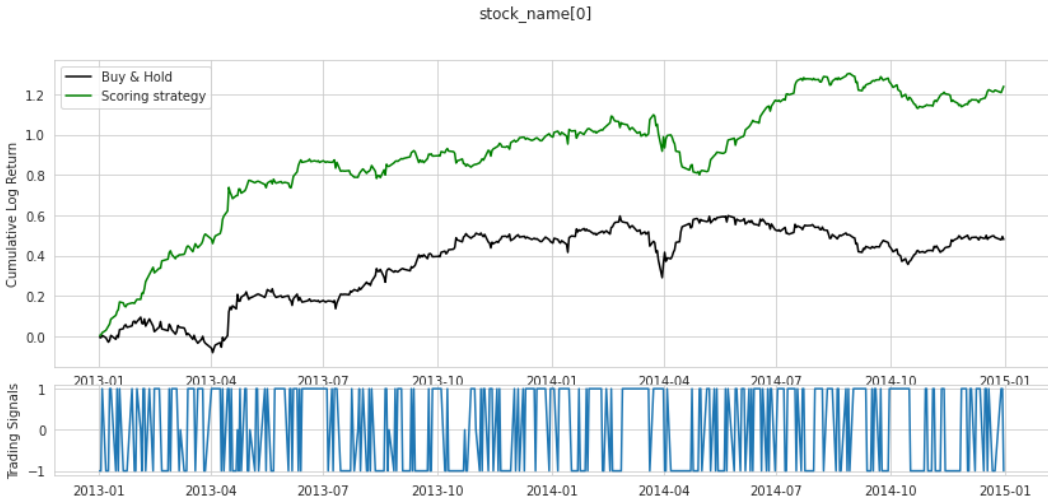


Figure 5-50: Results of all trades for CARR from our model during 2013-2014

often. Following all these signals, an investor will change position 202 times within two years, which in reality may lead to trading fees having a large negative effect on the investor's return, especially if they started with only a small amount of funds.

In this research, the representative brokerage fee (trading fee) is set to a flat rate of £3.79 per transaction (both buying and selling), calculated by averaging the brokerage fees of 16 well-known retail brokerages (details show in section 4.5.2 on page 107). Therefore, the total brokerage fee over the test period for our system, which made 202 transactions over two years, is £765.58. Trading this often is only feasible if with a large amount of starting capital, otherwise this amount in fees may be too expensive and will greatly diminish an investor's compound return over their investment lifetime. Some examples of how trading with different amounts of initial funds affects the result can be seen in the next section.

5.5.2.1 Comparison of different initial funds

According to the results in table 5.6 on page 212, if an investor follows all 202 trades the system suggests, the fees that the investor ends up paying amounts to £765.58. This is probably fine if they start with a large amount of capital. For example, the visualisation of results with £10000 starting capital can be seen in figure 5-51.

Figure 5-51 shows the result when a trader has initial funds of £10000. In the following example, we will show the results of trading under the same conditions (strategy and

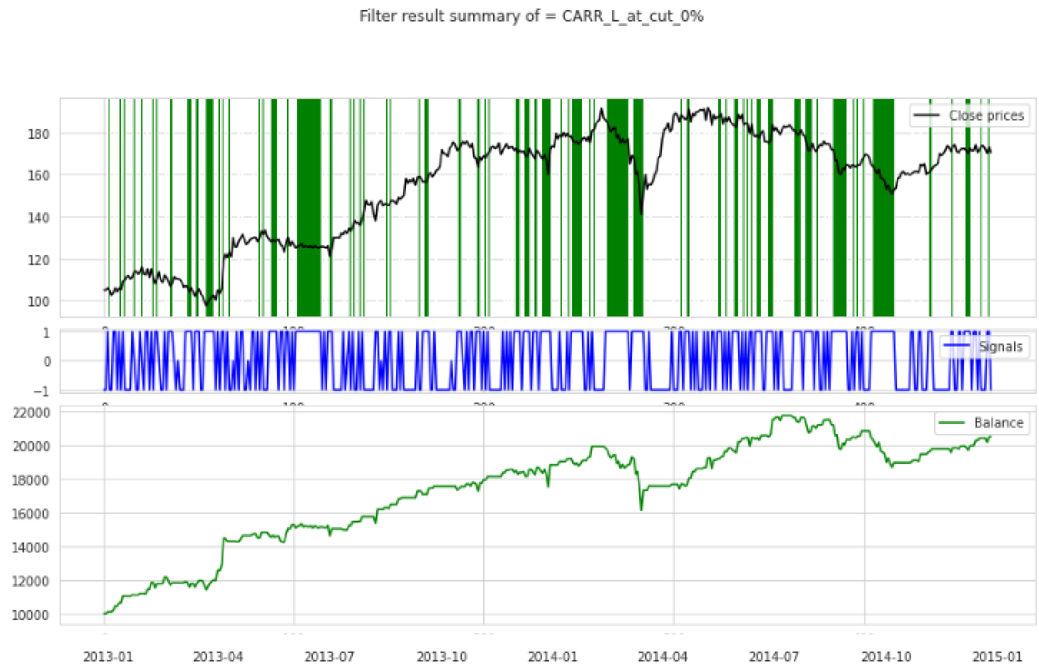


Figure 5-51: Result from signal creation of CARR when the initial fund is £10,000

dataset) with a smaller amount of initial funds.

Figure 5-52 shows the result of CARR with starting capital of £1000, which is far less than the initial funds in the previous example (£10000). This result is much worse than the previous result in Figure 5-51 on page 214. When starting with only £1000, following all the trading signals results in a decreased profit of 17.28% (instead of 105.13% as in the previous example starting with £10000). The Sharpe Ratio therefore also decreases to 0.38 (from 1.73 previously). These two values clearly indicate that trading performance decreases as a direct result of trading fees when starting with less capital.

These two examples were tested under the exact same conditions, the only changed variable being the amount of initial funds. The only reason that the second experiment provided a worse result than the first is because of the brokerage fee effect. As the total number of trades of these two experiments was the same, the amount paid in brokerage fees was also the same for each, £765.58. The second experiment initially had only £1000 but still had to pay £754.21 in fees, while the percentage profit and the volatility of the trades stayed the same. The result was that the profit the investor starting with the lower amount made was not enough to compensate for the brokerage

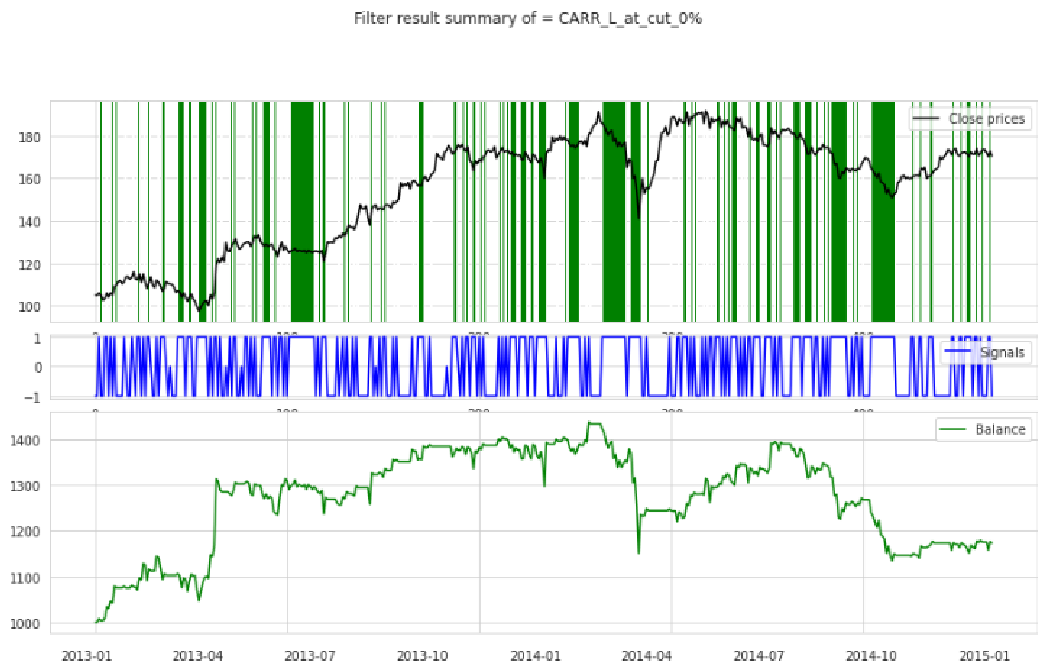


Figure 5-52: Result from signal creation of CARR when the initial fund is £1000

fees, which cannot be avoided. Even though at the end of the trading period the balance increased, the gain was only slight as it was greatly diminished by the fees. We have only presented the results here over the relatively short testing period, but over their lifetime an investor expects to compound their trading returns by reinvesting profits. Over a long time horizon, profit unable to be compounded because it is paid back in brokerage fees can make a huge difference to an investor's returns.

A detailed comparison of the trading results with £10000 and £1000 can be seen in table 5.7.

Table 5.7 shows all of the results from the Buy & Hold strategy and from our strategy with initial funds of £10000 and £1000. The Buy & Hold strategy has a positive Sharpe Ratio of 0.77, indicating that this strategy ends up with a profit after taking risk into account. The exact profit made was 62.38%. This number is the profit calculated from the accumulated logarithmic return over the whole trading period without taking brokerage fees into account.

The results from our model with initial funds of £10000 and £1000 are shown in the

Performance	Buy & Hold	Our model (no fee/fund)	Our model fund (£10000)	Our model (£1000)
Sharpe Ratio	0.77	2.18	1.73	0.38
Profit (%)	62.38	243.43	105.13	17.28
Maximum Drawdown	-26.32	-25.75	-19.12	-21.1
Stdev	0.302	0.292	0.226	0.227
Number of Trades	2	202	202	202
Brokerage fee	£7.58	£765.58	£765.58	£765.58

Table 5.7: Performances summary of CARR comparing between our model and the buy & hold strategy

third and fourth columns. With initial funds of £10000, our model performs very well, with a Sharpe Ratio of 1.73 and profit of 105.13%. This profit is calculated after taking brokerage fees into account. However, even though our model works very well with a high amount of money to start with, it does not perform well with lower initial amounts. With only £1000 to start with, this model ended up with a smaller profit of 17.28% over the test period of two years. The Sharpe Ratio decreases to 0.38. The performance of our system before including these factors in the experiment is shown in the second column (shown as our model (no fee/fund)).

To solve the problem of trading too many times without sufficient initial funds, signal optimisation is proposed in the following section.

5.5.2.2 Signal Filtering

Having observed the problem of brokerage fees when starting with a small amount of funds in the previous section, this section proposes a method for signal optimisation. The signal optimisation process will improve the quality of trading by filtering out weak signals and will decrease the effect of brokerage fees, since fewer trades will be placed. This signal optimisation module takes into account the initial funds that an investor is starting with, then performs grid search to find the optimal point to filter the signals. The details of this method can be found in chapter 4, section 4.5.4, on page 111.

In order to perform the signal optimisation, we need a development set (or dev set). We took the first half of the test set to be our dev set, starting from the beginning of 2013 until the year's end. The test set therefore starts from the beginning of 2014 through to the end of the year. The dev set is used for signal optimisation by running grid search in order to obtain the parameters for filtering out signals for the test set.

Before optimising, we present our system's and the buy & hold performances on the test set for CARR in figure 5-53.

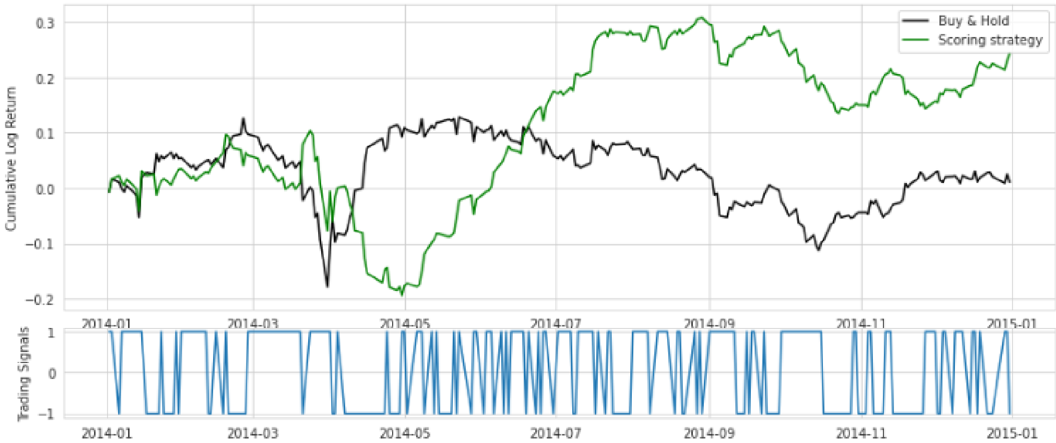


Figure 5-53: The comparison of our model's result and the buy & hold performance of CARR for the year 2014

For the year 2014, our test set in this experiment, our scoring system performed better than the buy & hold, which can be seen from the top graph of figure 5-53. The green graph (scoring system) is much better than the black graph (buy & hold), especially for the second half of the year. Our model ended up with 28.61% profit and a Sharpe Ratio of 0.79, while the buy & hold only obtained 1.07% profit with a 0.04 Sharpe Ratio. However, this result is before taking the fee and initial funds into account. The question remains whether our model's performance is still good once the brokerage fee and limited initial funds of £1000 and £10000 are accounted for.

Figure 5-54 and 5-55 show the results from our model after adding a brokerage fee of £3.79 to every trade and setting the initial funds to £1000 and £10000, respectively.

Figure 5-54 clearly shows that with initial funds of £1000, our model cannot make a profit on CARR during this test period (the year 2014). Four graphs are shown in the results. The top graph shows the closing price of CARR during the year 2014. The green highlighted areas represent long positions (from when a stock is bought until it is sold) which correspond to the second graph below. The second graph shows the signals from our system. The third graph shows the balance in the account, starting with £1000 and ending with below £800. Therefore, at the end of the testing period, the investor would have lost -23.43%. The bottom graph compares the cumulative log return of our model and the buy & hold.

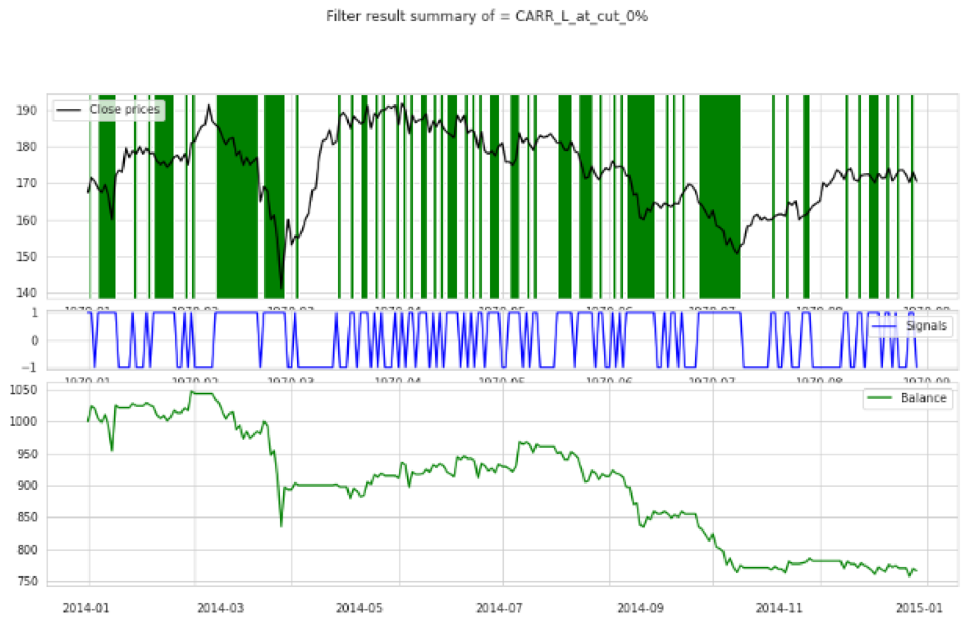


Figure 5-54: Result from our model with fee and an initial fund of £1000 (without signal optimisation)

Figure 5-55 shows results from the same stock, CARR, over the same period and with the all same conditions as the previous figure except greater initial funds. This iteration starts with £10000, not £1000 as earlier. The result from the third graph of this figure clearly shows that our model is able to make a profit (10.63%) when starting with more money, as we expected. Because the trading fee is flat, it now shows a smaller percentage of the amount invested in each trade, so we expected the model to achieve a higher profit. Details of the comparison between buy & hold and our model before and after accounting for brokerage fees can be seen in table 5.8.

Table 5.8 shows the comparison between the buy & hold strategy and our model under three different conditions. 1) Our model without taking any fees or initial funds into account, 2) our model with the brokerage fee included in every trade and initial funds of £1000. Finally, 3) our model with brokerage fees and initial funds of £10000. From the result of buy & hold, we see that this stock does not perform very well over this period, only ending up with a tiny profit of 1.07%, which is a very low return considering the stock had to be held for the whole year. Such a return is comparable to that of far less-risky cash in bank account. However, our model is able to obtain a much higher

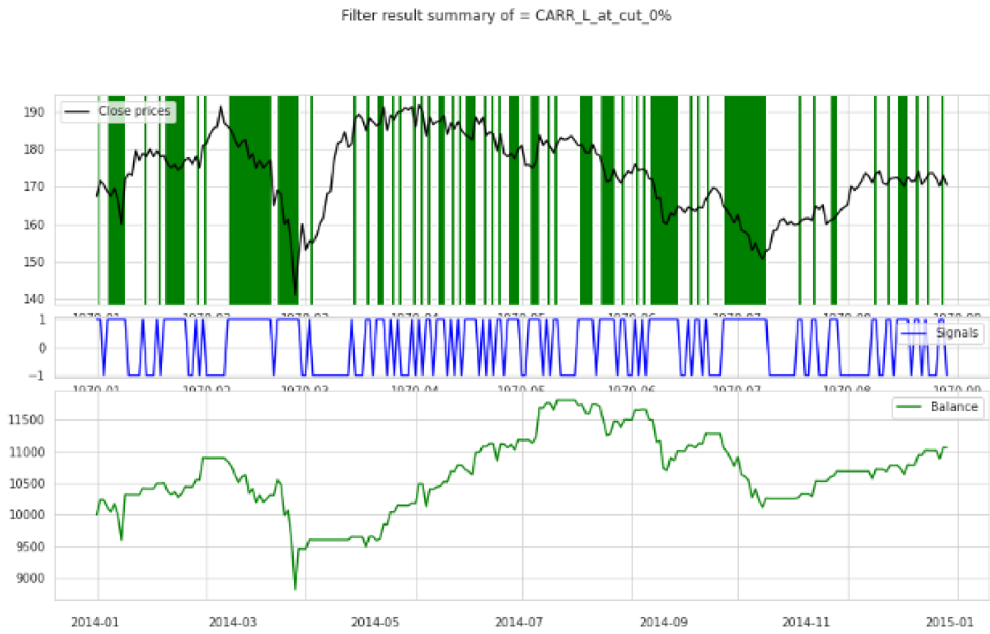


Figure 5-55: Result from our model with fee and an initial fund of £10000 (without signal optimisation)

Performance	Buy & Hold	Our model (no fee/fund)	Our model (£1000)	Our model (£10000)
Sharpe Ratio	0.04	0.79	-1.21	0.46
Profit (%)	1.07	28.61	-23.43	10.63
Maximum Drawdown	-26.32	-25.75	-27.71	-19.17
Stdev	0.296	0.295	0.232	0.23
Accuracy	52.5	57.45	57.45	57.45
Number of Trades	2	96	96	96
Brokerage fee	£7.58	£363.84	£363.84	£363.84

Table 5.8: Performances summary of CARR comparing between our model and the buy & hold strategy for the test period of year 2014

profit of 28.61%, although this is the profit before taking the fees and initial funds into account and is therefore unrealistic.

Once we try to simulate these trades closer to the real world by adding the fees and initial funds, the results changed completely. In the case of only having £1000 to start with, the model ends up with -23.43% loss, instead of almost 30% profit as mentioned earlier. This is because of the brokerage fee effect. Every time the model produces a trading signal that is acted upon, there is a brokerage fee the investor has to pay.

During this trading year, there are 96 transactions, which means that £363.84 were paid in fees. That's almost 40% of the money already gone. There are not enough opportunities to make a profit with the leftover money. Therefore, the model results in a loss. However, with £10000 to start with, our model is able to make money and end up with 10.63% profit. This is not considered to be a big gain, but is considerably better than the buy & hold strategy. Still, our model did not manage very well with this stock, as can be seen from the top graph of figure 5-54 or 5-55. There were five occasions during this testing period that our model bought the stock and its price subsequently went down (represented by the big green highlighted areas). The reason that our model could not cope very well and gave out the wrong signals is because this testing period was different from and more fluctuated than before, as can be seen in figure 5-56.

Figure 5-56: The closing price of CARR from 2000 to 2014

From figure 5-56, it can be seen the testing period of 2014 has more fluctuation than in previous years. Since our predictors were trained on a period with less fluctuation, it is possible that the model experienced some difficulty dealing with the fluctuations, although with enough initial funds it did still provide a reasonable profit. When there is a problem with the fluctuation of the data, the signals can be all over the place and there may be more wrong signals than usual. If we follow every signal, the model's performance may be poor, especially when starting with less money to trade. Therefore, performing signal optimisation can be helpful in this situation, since it will ideally filter unprofitable signals or those that result in little or no profit after fees. By doing this, we expect our model to be able to make more profit, even with less initial funds.

Examples of the results from the signal optimisation process with initial funds of £1000 and £10000 are shown in Figure 5-57 and 5-58.

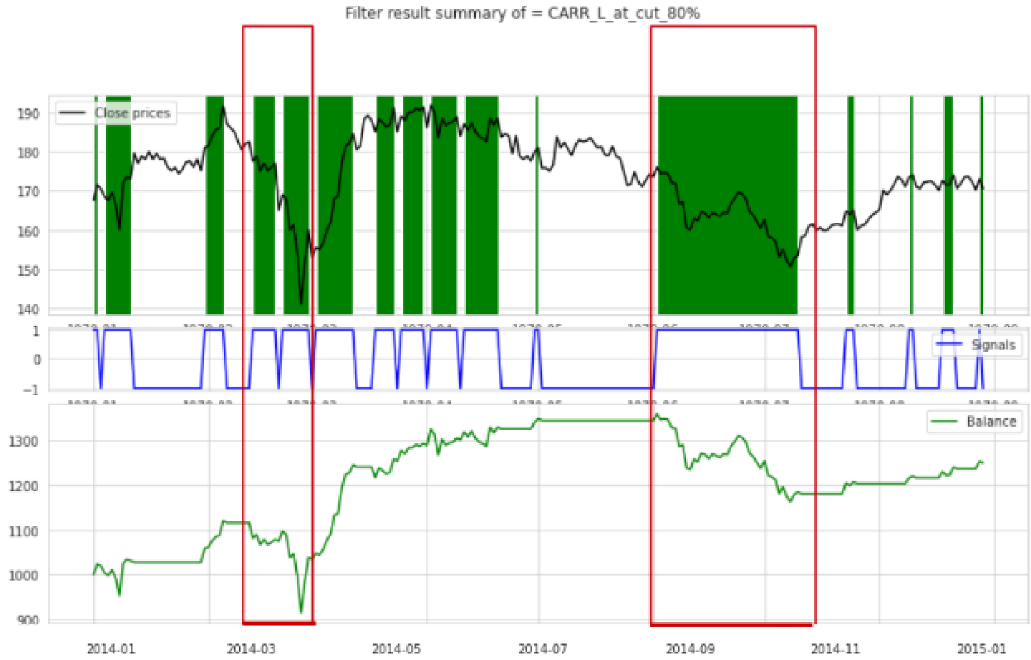


Figure 5-57: Trading's performance after optimisation of CARR with an initial fund of £1000

We have shown results from our system before and after the signal optimisation in two cases, with initial funds of £1000 and £10000.

Firstly, the results with £1000 initial funds. Without signal optimisation, our model performs badly when starting with £1000. This can be seen in figure 5-54 on page 218. Our model had to take all the trading signals, resulting in 96 trades in total. Finally, it ended up with a negative Sharpe Ratio of -1.21 and a -12.43% loss. The top graph of this figure shows there were wrong trades, such as after 2013-03, when two long positions should not have been opened since the price subsequently went down during those periods. Another two incorrect long positions were taken during 2014-09 and 2014-11. These incorrect signals lead to big drops as can be seen in the third graph of the same figure. Next, we will see if signal optimisation would help filtering these kind of signals and bring us back to profit.

Figure 5-57 shows results from the same conditions as figure 5-54 on page 218 but with signal optimisation included. It can be seen that at the same periods in the two red

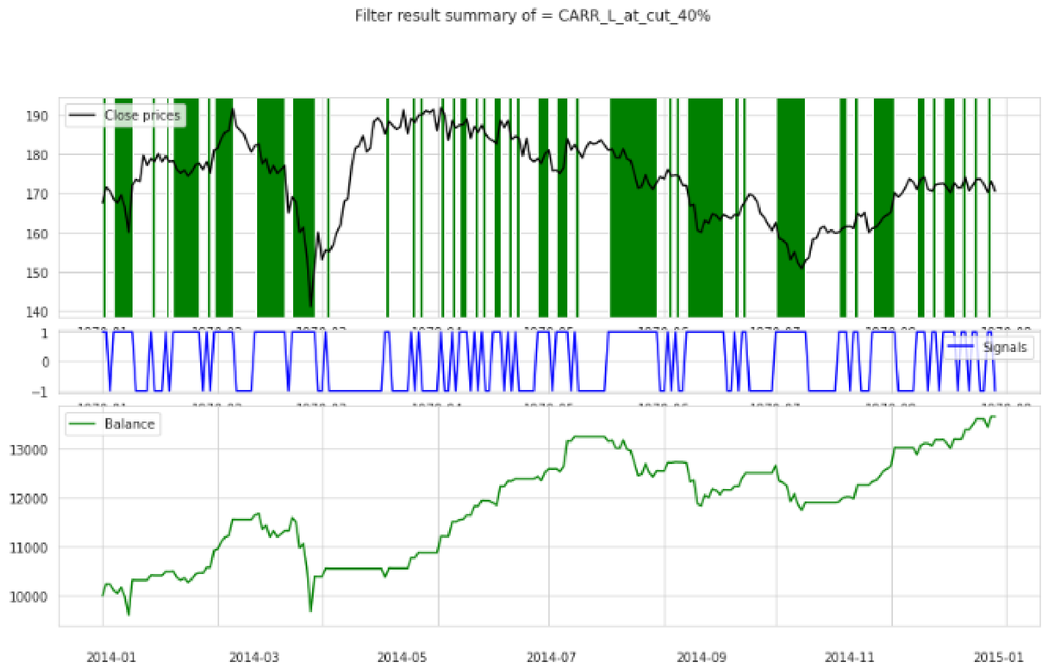


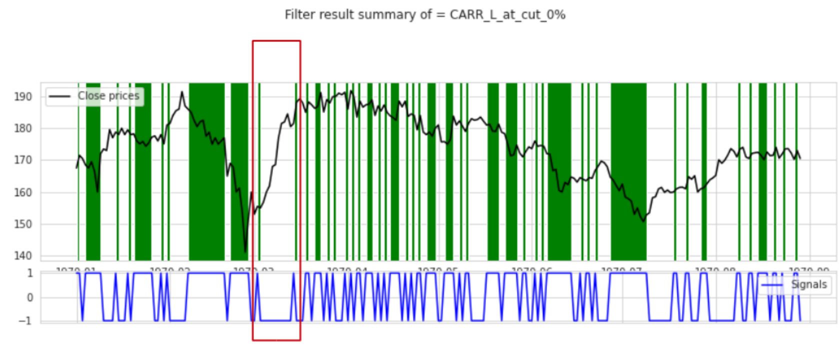
Figure 5-58: Trading's performance after optimisation of CARR with an initial fund of £10000

boxes, the number of trades is reduced when signal optimisation is performed. There will be fewer changes of signals, as seen by the longer green areas in this result compared to the previous result.

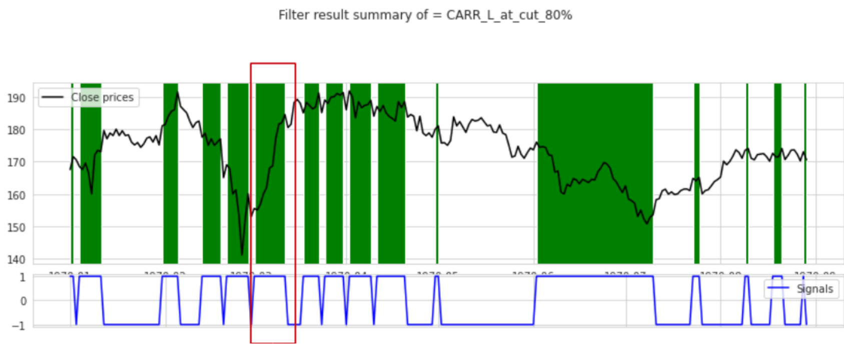
Figure 5-58 shows the result after the signal optimisation for the same conditions, for example starting with £1000. Unfortunately, the signal filtering method was unable to get rid of the bad signals we mentioned above, as can be seen in the two red rectangle boxes. These signals were strong signals (signals with high expected returns) so the filtering model did not filter them out. However, the filtering helped a little by making the first wrong position shorter (the first long position in the first red box). This means that it actually got rid of the initial buy signal so this incorrect position started a little bit later, resulting in losing less money.

Even though the filtering module could not get rid of the wrong signals mentioned above, it still ended up with a much bigger profit of 24.81% (instead of losing -23.43%). This is because of two reasons. Firstly, the system is able to get rid of bad small trades. As can be seen from the top graphs, the small and noisy signals are filtered out. The number of times the system has to change position decreases from 96 to only 32 times, incurring far less trading fees as well as avoiding some losses. This is the first reason that the optimisation worked well. The second reason is that the optimisation module is able to make profitable trades be held for longer. For example, when the stock price is trending generally up, it is often best for our system to buy on day 1 and hold the position while the trend continues. However, there are a few sell signals that come during these times and without filtering our system will follow these and close the long position. Then, after the turbulence, it will buy again in order to ride the trend again, but having lost some of the profit in the intervening period when it closed the position. We found that signal optimisation can be helpful with this situation by filtering out the small noisy signals. If the sell signals are not too strong, they will be filtered out and our profitable trade will continue (thereby not incurring more fees).

The results of the second situation are very positive, so we will show the results before and after optimisation in figure 5-59 for ease of comparison.



(a) Before signal filtering



(b) After signal filtering

Figure 5-59: The signals for CARR before and after performing signal optimisation

Figure 5-59 shows the result of our system before and after performing signal optcomparison withimisation. The top graph (a) shows the result before signal filtering. The red rectangle shows a situation where the system started to catch an upward trend and correctly bought the stock. However, not long after opening the position, a sell signal was received. Without optimisation, it only takes one sell signal for the system to close the buy position, after which it will hold cash until another buy signal is received. In this case, the second buy signal came too late, at the end of the upward trend. Therefore, the system missed a good profit opportunity.

The result after signal optimisation is shown in the bottom graph. Over the same period of time (shown in the red box), the system was able to catch the trend at the very beginning, opening a long position by buying the stock. As we saw from the top graph, a sell signal comes shortly after this. However, in this case our signal filtering got rid of that signal. So, the stock was held until the end of the upward trend. This led to a large profit in this case.

From the results shown above, we see that signal filtering is very helpful, especially when the investor has limited funds such as only £1000 to start with. It helped results change from losing over 20% to gaining 24.81%. The optimisation process also improved the trading performance of our system when starting with initial funds of £10000. The comparison between the result before and after signal optimisation with £10000 can be seen in figure 5-55 on page 219 and figure 5-58 on page 222, respectively.

Comparing between these two figures, it can be seen clearly that the results after optimisation in figure 5-58 are significantly better than before. The two bottom graphs clearly show more profit after optimisation. A comparison of the details of these two cases can be seen in table /reftab: Ex evaluation CARR realTest after optimisation.

Performance	£1000		£10000	
	(Before)	(After)	(Before)	(After)
Sharpe Ratio	-1.21	0.99	0.46	1.47
Profit (%)	-23.43	24.81	10.63	36.63
Maximum Drawdown	-27.71	-18.2	-19.17	-17.12
Stdev	0232	0.236	0.232	0.225
Accuracy	52.5	80	57.45	69
Number of Trades	96	32	96	80
Brokerage fee	£363.84	£121.28	£363.84	£303.2

Table 5.9: Performance summary of CARR comparing our model and the buy & hold strategy after optimisation for the test period of the year 2014

Table 5.9 shows all of the improvements of our system after performing signal optimisation. For both cases, starting with initial funds of £1000 and £10000, profits improve after optimisation. For £1000, the profit increases from -23.43% to 24.81%, as well as the Sharpe Ratio which improves from -1.21 to 0.99. This is similar to the result from starting with £10000. Profit increases from 10.63% to 36.63%.

When starting with less money, our signal filtering is helpful in that it decreases the number of trades and therefore fees. In the case of having initial funds of £1000, the number of trades after optimisation decreases from 96 to only 32. This makes the fees much less, decreasing from £363.84 to £121.28. However, the number of trades goes up when starting with more money. With £10000 initial funds, the number of trades increases to 80 even after signal filtering. This is because our system performs filtering by taking the amount of money an investor has into account. Having more money means investors suffer less from the impact of fees and make more trades so as to have more opportunities to profit. The signal filtering module tries to find the optimum point where investors have the maximum Sharpe Ratio given their original money. However, please note that the module will not guarantee maximum accuracy. Taking more trades could cause a decrease or increase in accuracy as can be seen from the accuracy when trading starts with £10000. When the initial funds increase from £1000 to £10000, the signal filtering module will attempt more trades, increasing from 32 to 80 times. Even though this makes more profit and gives a higher Sharpe Ratio, the accuracy decreases from 80% to 69%. However, as we have mentioned in the discussion section (5.4.1 on page 192), accuracy is not our main interest and we do not consider the decrease in accuracy to be a problem.

Before moving to the discussion section of this experiment, we will show one more example of our system's performance before and after performing signal optimisation. We will show the result from FERG. The conditions of the test are the same as those we used to run the experiment on CARR. The dev set is the whole year of 2013 and the test set that we are going to use to compare results is the whole year of 2014. There will be two values of initial funds, £1000 and £10000. The results can be seen in table 5.10.

Table 5.10 shows the comparison of FERG's results for the buy & hold strategy and our model with different conditions, which are our model before adding fees and the initial funds, our model with initial funds of £1000 before & after optimising the signals and our model with initial funds of £10000 before & after optimising the signals.

Table 5.10: Performances summary of FERG comparing between our model and the buy & hold strategy for the test period of year 2014 after optimisation

Performance	Buy& Hold	Our model (No fee/fund)	£1000 (Before)	£1000 (After)	£10000 (Before)	£10000 (After)
SR	0.41	1.71	-0.75	1.59	1.33	1.59
Profit(%)	7.62	38.83	-8.63	19.48	17.11	19.48
Max DD	-14.47	-13.54	-11.39	-6.56	-7.4	-6.56
Stdev	0.178	0.175	0.121	0.113	0.119	0.113
Acc	49.6	57.37	57.37	60	57.37	60
Trades	2	68	68	40	68	40
Fee	£7.58	£257.72	£257.72	£151.6	£257.72	£151.6

The buy & hold strategy for FERG in this testing period made a profit of 7.62% with a Sharpe Ratio 0.41, which is not very much. When our model traded this stock, the model provided a much better result than the buy & hold, as shown in the third column. The Sharpe Ratio increased to 1.71 while the profit increased to 38.83%. However, this is the unrealistic result before taking the brokerage fees and the initial funds into account. To get more realistic results, we have included a £3.79 brokerage fee for every trade and set the initial funds to £1000 and £10000.

For each amount of initial funds, we have shown the results before and after the signal optimisation process. With £1000, the return goes from a -8.63% loss to a 19.48% profit. The Sharpe Ratio also increases from -0.75 to 1.59. These are very good results given the initial money is only £1000. The signal optimisation is able to filter out bad signals, resulting in the number of trades decreasing from 68 times to only 40.

However, when more initial funds are provided, such as £10000, the number of trades remains at 40. The accuracy and the standard deviation also did not change. This suggests that all the trades were the same even after adding more money. This will clearly lead to more profit and a greater Sharpe Ratio as it is just the same trades as before but with more money. Ideally, putting more money should allow an investor to trade more often. However, this is not the case here. The reason that the number of trades did not increase when starting with more money here is that the noisy signals are too weak (signals with very little expected return). The optimisation module did not find them worth trading even with more money. This suggests that the optimal number of trades for FERG in 2014 was 40 times with initial funds between £1000 and £10000.

5.5.2.3 Discussion

This experiment shows the importance of filtering signals out, especially for individual retail investors. Since this is the audience this research focuses on helping, this process is of great importance and is necessary to include in our trading system. The filtering process helps optimise the number of trades for each individual trader. The number of trades will be adjusted according to the initial fund that investors start with. More funds means the investor is able to take more trades. One the other hand, when an investor starts with limited funds, the number of trades will be low. In this section, we showed some examples of the results to demonstrate this conclusion. Besides those presented here, we performed the experiment on each stock with different initial funds, raging from £1000 to £10000, the results of which will be shown in the next chapter.

5.6 Summary

This chapter described the experiments we conducted as we developed our model step-by-step. There were two sets of experiments in this chapter. The first group comprised the three main experiments, related to the development of our central idea, the scoring system. Experiment one started by testing our initial hypothesis that multiple predictors would be better than a single predictor. Experiment two dealt with how multiple predictors could work together and removing look-ahead bias from the first experiment in order to make system more realistic. After confirming the advantage of having many predictors, experiment three investigated the most essential part of our trading system, which is the scoring mechanism. This experiment aimed to build an effective scoring system which could manage the cooperation of predictors effectively. The final design of the scoring system was obtained after this experiment. Finally, the second set of experiments tested some system adjustments: cluster selection and signal optimisation. We call these adjustments only since they improve the performance of our model but do not affect our central scoring system. The system would still work without performing these last experiments, but it would perform worse.

During these experiments, we encountered the look-ahead bias a few times. Initially, this was deliberately included to test the validity of our multiple predictors idea, but at one point it did appear by mistake. This type of bias can make the results very different. Therefore, it worth mentioning before moving to the next chapter. The intentional bias has been discussed at length in experiments one and two, especially in experiment 2 on page 147. Here, we will only discuss the potential effect of this bias that could have happened in another experiment, that of the number of clusters, 5.5.1 on page 196.

In this experiment, we investigated the best number of clusters for our system. We found that increasing the number of clusters will help our system but only up to a certain point. Once the number of clusters is too high, our results start to decrease and exhibit more fluctuation across different stocks. The results we have just described can be seen in 5.5.1 on page 196

However, had we accidentally included the look-ahead bias in this experiment by including the price of the next day in the calculation of the mean and standard deviation, the results would have turned out very differently. What follows are the results with look-ahead bias included when experimenting on the number of clusters from two to five. These results come from TSCO stock (the same stock as shown in 5.5.1). Every setting is the same as in section 5.5.1, except that in the process of selecting the predictor, the mean and standard deviation are calculated by including the value from the next day.

It can be seen from figure 5-60 that something has gone wrong. We realised right away once this kind of result happened; it was just too good to be true. We expected that the performance could only increase with the number of clusters up to a certain point because of overfitting, but here, as the number of clusters increases (in the left figure), the results from our trading model go up rapidly (the green line from the graphs on the right side). With the number of clusters increasing from two to five, the Sharpe Ratio has increased from about 1.5 to 4.0 and could keep on going thanks to the impact of look-ahead biases, but the results would be impossible to achieve in real life.

We have already seen that the look-ahead bias can have a big impact on the results, so below we briefly describe the measures we took to avoid this bias.

- Separate data: Our data was separated into three files: the training, dev and test sets. The training set was only used to train the predictors, while the dev and test sets were used for creating the system and testing our model.

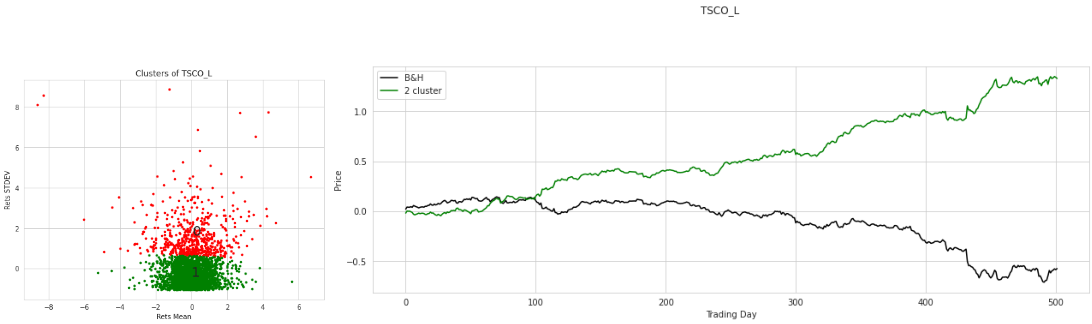
- Separate code: We separated our code into multiple files and functions. The code we used to train our predictors was separated from the others. The training code will read the training set and train predictors. Then, the trained predictors will be saved separately, one file for one predictor. Therefore, the predictors are 100% certain never to have seen other parts of the data. Once the system wants to predict, the predictors will be read from files by the testing code.

- Separate calculation: To prevent miscalculations that might include the look-ahead bias, in the testing code, we only read data up to the close of the day the signal is generated (the day before order execution) and calculate the mean and standard deviation daily to make sure that only data up to the present day has been included in the calculation.

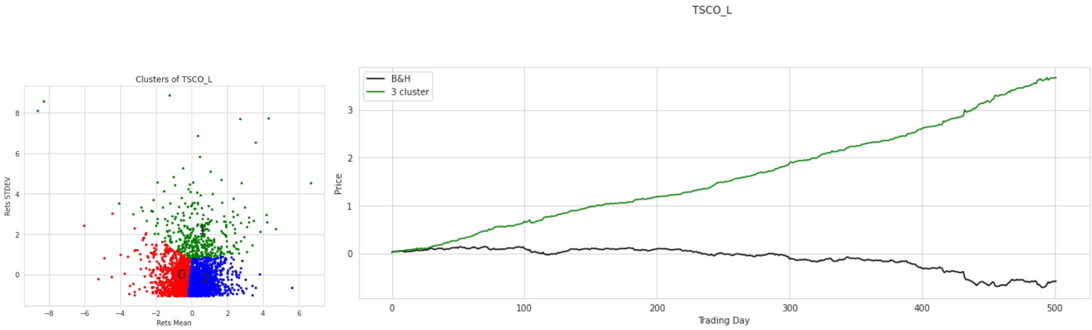
- Other settings: We were also careful about normalisation. As some of machine learning models are very sensitive to the scale of data, such as k-means clustering or SVR, we performed normalisation on the training set, which was separated from other parts of data. Then, we saved our normaliser into a file. When it came to the testing time, we read the normaliser and used it to adjust our testing data (only adjust, not recalculate: in other words, transform but not fit the model again). This method makes sure that the normalisation is calculated (or fits) only on the training data.

Given the methods we selected, we are certain that our model was not exposed to any look-ahead bias. Therefore, the results we have shown are correct. The only possible small difference that could affect our results is the simulation conditions, such as the fees. There might be other type of fees that an investor needs to pay - such as stamp duty, a tax paid when purchasing a share in the UK (excluded in this research because it may not apply in other countries, while trading fees are universal)- or the amount of fees may be different from our program since we only considered an average fee when running the simulation.

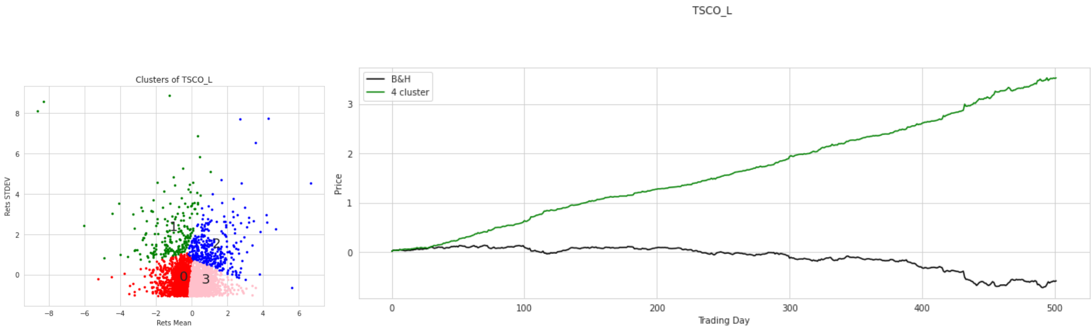
The result comparison section 5.4 on page 186 compares all the methods we investigated, which are six individual predictors and 4 ensemble-based strategies. The results from these strategies are presented in terms of profit, Sharpe Ratio, maximum drawdown, accuracy and standard deviation. The conclusions of our main experiments (one to three) can be found shortly after that in section 5.4.1 on page 192. After that, we moved onto experiment four, which looked at other system configurations to improve performance and simulate real-world trading systems. From all experiments, we concluded the final design of our trading system, which can be seen in chapter 4 on page 88. This design will be used in the next chapter where we evaluate our system. The system will be tested with a lot more stocks to ensure that the design works well for a variety of stocks.



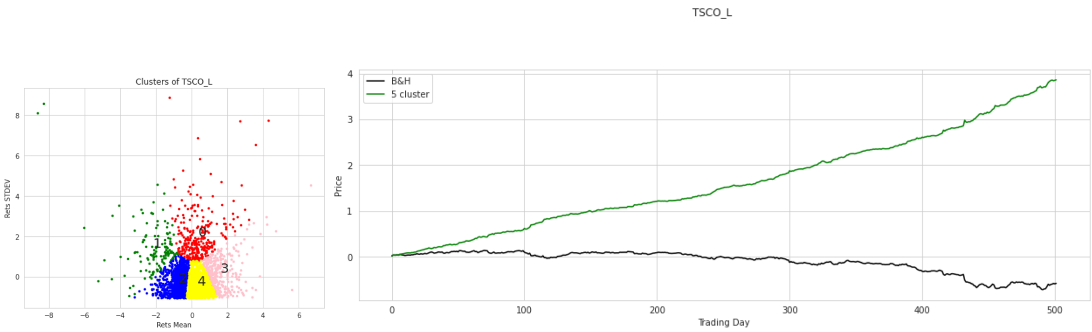
(a) Two cluster (b) Result of tow clusters



(c) Three cluster (d) Result of three clusters



(e) Four cluster (f) Result of four clusters



(g) Five cluster (h) Result of five clusters

Figure 5-60: Example of TSCO with 2-5 clusters once biases included

Chapter 6

System Evaluation

In this chapter, the performance of our model is shown on different stocks.

6.1 Key metrics for system evaluation

In this section, each stock from the previous section will be used to evaluate the performance of our trading system. There are several metrics for identifying model performance but the most important one is the Sharpe Ratio as this value already includes two main important aspects of the trading process: profit and risk. A system's performance cannot be judged only by how much profit it has made. One cannot simply say high profit, high performance because it does not matter large the profit your system can make if investors have to take too much risk. In such cases, it is more likely that the investors will run out of funds before getting rich. Therefore, the Sharpe Ratio, which is also known as the "Risk adjusted return", is considered to be the most important performance evaluation feature in this project and in the investment area. Besides the Sharpe Ratio, several other metrics will be measured. The details of all the metrics are as follows:

- Sharpe Ratio (SR): Also known as *Risk Adjusted Return*. This value is the most important for the system evaluation process. It adjusts the return by the risk that an investor has to accept to achieve it. The detail and equations of SR calculation can be seen in section 2.3.4 on page 42.
- Profit: This is the annual return or *yearly return*. This value is one of the basic metrics in system evaluation. However, profit taken on its own may not be reliable

as more profit does not always indicate a better system. The risk of the system needs to be taken into account as well. Annual profit calculation detail is shown 2.3.4 on page 42

- Drawdown (DD): This is the decline in equity between a successive peak and trough. In this project, we will calculate the *maximum drawdown*, which is the largest peak-to-trough decline over a given period of time. Therefore, drawdown is the loss at a specific time and maximum drawdown is the biggest loss of the whole trading period. This value is important because a bigger drawdown means that the system will need more time and money to recover from the loss. More details of drawdown can be seen in Figure 2-4 on page 48.

- Volatility: This is the key metric to evaluate risk within the system. The volatility is calculated from the standard deviation of the return. This value will identify the stability of your system. In this project, the annual volatility will be taken in to account.

- Fee: This is one of the most important considerations for traders, especially traders with less funds to start with. THe fee is the price that every trader has to pay to their broker. Therefore, it can be called a *brokerage fee* as well. Every time a trader opens or closes his/her position, a fee is paid to the broker for the execution of the order. There are two types of brokerage fee which are flat-rate or percentage-rate fees. A percentage fee is normally applied to the institutional investors or the investors with a very large amount of funds, while a flat-rate fee is the normal fee for an individual investor. As this research aims to be used by individual investors, we will only use a flat-rate fee. Throughout this research, a fee of 3.79 will be paid for every opening or closing a position. This fee is taken from the average of 16 popular brokers [92], as mentioned in chapter 5 section 4.5.2 on page 107.

6.2 Experimental results on 11 selected stocks

6.2.1 Datasets

In order to evaluate our model, we have selected variety of stocks from different sectors, ranging from food producers, retailers, general industrial, construction and electronics to computer and technology compamies. These stocks are mainly from the UK (London stock Exchange) and some of them come from the well-known companies in America

(NASDAQ and NYSE markets). The list of stocks that have been selected can be seen in table 6.1 on page 233.

Symbol	Name	Detail	Market
CARR.L	Carr's Group plc.	Food producers	LON
TSCO.L	Tesco plc.	Multinational groceries	LON
MKS.L	Marks & Spencer Group plc.	Multinational retailer	LON
MACF.L	Macfarlane group plc.	General Industrials	LON
FERG.L	Ferguson plc.	Industrial suppliers	LON
COST.L	Costain Group plc.	Construction and Materials	LON
OXIG.L	Oxford Instruments plc.	Electronic and Electrical Equipment	LON
SXS.L	Spectris plc.	Electronic and Electrical Equipment	LON
D4T4.L	D4T4 solutions plc.	Software and computer service	LON
GOOG	Alphabet Inc.	Multinational conglomerate	NASDAQ
MCD	McDonalds	Fast food company	NYSE

Table 6.1: Stock list

The reasons we selected stocks from a wide range of business sectors is to be able to evaluate our model against a variety of stock characteristics. Figure 6-1 shows the closing prices from the year 2000 to 2018 of all the stocks we mentioned in table 6.1 above.

As can be seen from Figure 6-1, the stocks can be grouped based on three different characteristics, and they were selected specifically because of this. The top group, sub figures (a) - (c), contains stocks have generally shown an upward trend over the given period. MCD and GOOG have very clearly shown an increasing trend over almost all of the period, while SXS.L exhibited more fluctuations but in general it also shows an increasing trend.

Sub figures (d) - (g) shows the closing prices of stocks with up and down characteristics. The first two stocks, MKS.L and TSCO.L, are more fluctuated, which can be seen from the prices going up and down all across the period. The remaining two stocks, COST.L and FERG.L, also have fluctuating prices but less so. These two stocks mainly go up and down for two big cycles.

The final group of stocks are shown in sub figures (h) - (k). These are stocks which have an uncommon characteristic. They have a long period of stability before the prices start to change. The first three stocks, CARR.L, MACF.L and OXIG.L, were stable for more than half the time, then the prices started shooting up and down. Therefore,

(a) MCD (b) GOOG (c) SXS.L

(d) MKS.L (e) TSCO.L (f) COST.L

(g) FERG.L (h) CARR.L (i) MACF.L

(j) OXIG.L (k) D4T4.L

Figure 6-1: Stock's closing price data

the predictors will be fitted with a lot of stable prices and have to perform predictions when the prices have different characteristics (as we train our predictors with the data from years 2000-2014). The last stock, D4T4, is a little different as it starts with a big increase and decrease in price. However, after the first year, it goes into a long period of stable pricing, as do the other stocks in this group.

In this section, the results from the 11 stocks mentioned in table 6.1 will be illustrated in order to show the trading system's performance. Firstly, we select a few stocks to show in detail and present graphs of their results to offer a better understanding. Secondly, all of the results from every stock are summarised to show the overall performance of our system.

6.2.2 Example of Results

6.2.2.1 Datasets

The results of three different stocks will be shown in this section. To demonstrate the results across a variety characteristics, we have selected one stock out of every group in Figure 6-1 on page 234.

Firstly, we selected McDonalds which is a company from the first group for which the price increases overall throughout the period of the dataset (2000-2018). McDonalds is one of the biggest fast food companies from the United States and has branches all around the world. This company's symbol in the market is *MCD*. Secondly, we selected Tesco from the second group. Stocks in this group have both upward and downward trends and are more fluctuated than those in the first group. Tesco is a well-known multinational grocery retailer from the United Kingdom which has almost 7,000 branches around the world. This company's symbol in the market is *TSCO*. Finally, for the third group which are stocks with uncommon characteristics as they include some stable periods before the prices start to change and there are many more fluctuations in the test set than in the training set, we selected CARR's group plc. It is a company that focuses on agricultural activities in the UK. It is also known in the market as *CARR*.

These three selected stocks are divided into three sets, which are training, validation and test sets. The test set is to be used for the evaluation of our model's performance. Therefore, all of the results in this chapter come from testing our model on the test set only. Figure 6-2, 6-3 and 6-4 represent the closing price of McDonalds, Tesco and CARR's group plc, respectively. The training set is represented in blue, ranging from 2000 to 2014. The validation set is shown in yellow and starts from 2014 to 2016. The test set is green and encompasses the rest of the data ranging from 2017 to 2018.

The closing prices of MCD which comes from the first group show in Figure 6-2. Dataset has an upward trend in overall. However, it shows downward trends and ranging behaviour in the first six years of training, with the price still shooting up

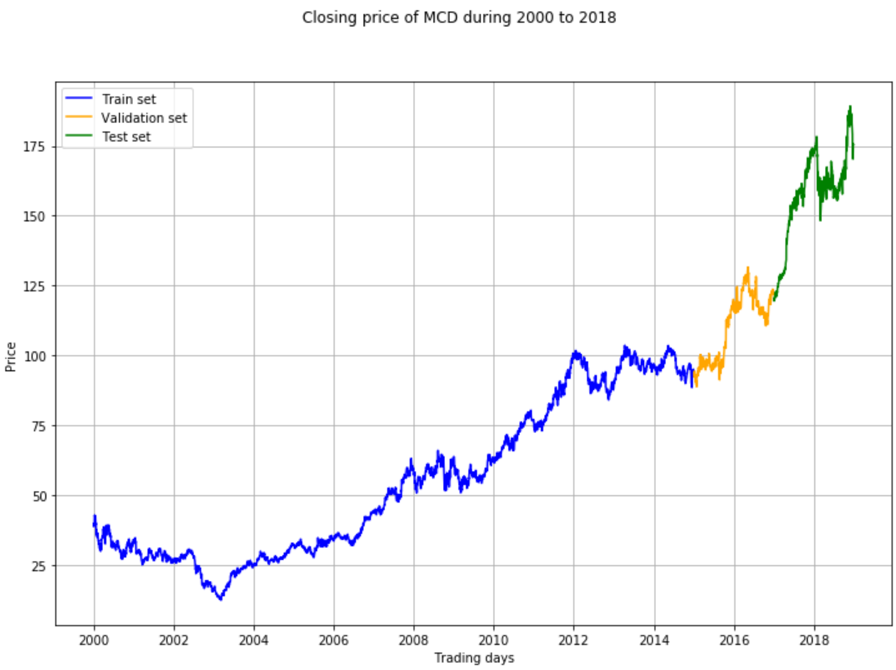


Figure 6-2: Closing price of MCD from 2000 to 2018

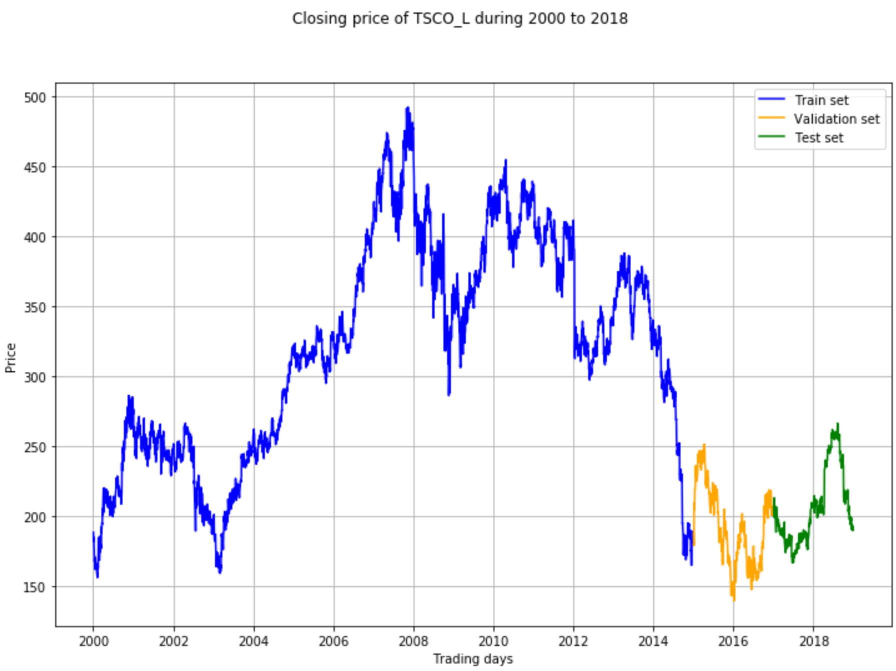


Figure 6-3: Closing price of TSCO from 2000 to 2018

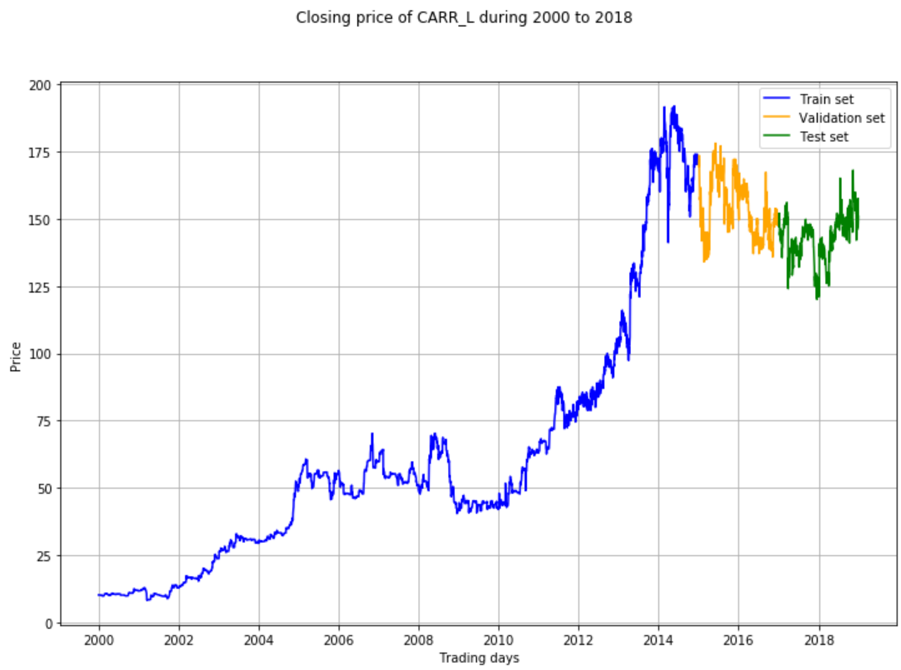


Figure 6-4: Closing price of CARR from 2000 to 2018

for the rest of this period. The characteristics of this data should be helpful for the predictors in the training period as they will be able to learn from different trends. As for the testing period, this displays a combination of upward and downward trends. At the beginning of 2018 there is a sharply decreasing period which never happened in the training period. This could be especially challenging for our model.

As for TSCO which represents the second group, closing price is shown in Figure 6-3. There are a variety of characteristics in the training period. This should benefit the learning process of our model as the predictors can learn different patterns. TSCO is very fluctuated overall, except for the obvious uptrend which lasts for about four years from 2003 to 2007. At the end of the training period, TSCO experiences a massive drop in the share price of about 58%. In the case that only one predictor is used, this characteristic of the training set would be very confusing for the predictor. However, as we have created multiple predictors and trained them with specific clusters of data representative of different market behaviours, it is not expected that sudden drops in price should be a problem. However, the validation and testing sets also had a combination of trends, and at the end of the test set the price drops by about 25%, so we will see how our model copes with the challenge of drastic price changes. Even though this drop is not biggest loss for this dataset, it is the biggest in the testing set.

It will be interesting to see whether our model is able to handle such sudden changes by selecting the correct predictor(s) to be used during this period.

Figure 6-4 shows the closing prices of CARR's group plc which is a stock from the third group. It can be seen that CARR has more different characteristics than the two datasets in the previous section. The difference between the characteristics of the training set and the other sets is very clear. The training set has two characteristics which are going up overall and having little fluctuation in the middle of the period, meanwhile the validation and test sets are very fluctuated. It is clear that these two sets are ranging and bullish or bearish periods cannot be identified. The overall characteristic of this dataset could be challenging for our model because the predictors will be fit for the overall uptrend data and a little bit of ranging. However, the validation and test sets are almost totally ranging. It would be interesting to see how well our model can work with this dataset.

6.2.2.2 Buy & Hold strategy

The buy & hold strategy is one of the most common benchmarks against which to measure performance for every stock. It measures how much an investor could have got if he/she had bought the stock at the beginning and sold it at the end of the testing period. Therefore, in this experiment, buy & hold strategy means an investor had bought this stock at the beginning of 2017 and sold at the end of 2018. The details of buy & hold performance of MCD, TSCO and CARR can be seen in table 6.2. Please note that this table shows result without taking the risk-free rate into account to make it easier when comparing to the results from graphs. However, we realise risk-free rate is important, therefore the 3% risk-free rate will be added in the comparison tables at the end of each experiment.

Stock	Sharpe Ratio	Profit (%)	STDEV	Max DD	Avg DD
MCD	1.07	44.23	0.174	-16.87	-4.18
TSCO	-0.19	-8.4	0.231	-28.79	-10.26
CARR	0.01	0.92	0.386	-23.08	-9.43

Table 6.2: Results from Buy & Hold strategy of MCD during the test period of 2017-2018 (not taking risk-free rate into account)

The buy & hold strategy of MCD shows that the buy & hold already works very well, with a Sharpe Ratio 1.07 indicating that this strategy achieves a good profit even after adjusting for the risk the investor has to take. This high Sharpe Ratio corresponds with the graph in Figure 6-2 on page 236, which shows that for the testing period, if an investor bought at the beginning (at a price of around 119) and sold at the end of the period (at a price around 175), that investor would have made a profit of over 40%.

As for TSCO, the buy & hold for this stock performs poorly, much worse than the previous dataset (MCD). The resulting loss is able to seen from Figure 6-3 on page 236 which shows that for the testing period, if an investor bought this stock at the beginning at a price of around 210 pence, they would have sold at the end at a price around 190 pence. In this case, buy & hold also has a negative Sharpe Ratio of -0.19.

Finally, the buy & hold strategy of CARR show that it does not perform very well but still performs better than the previous dataset (TSCO) as the Sharpe Ratio and profit are positive. The Sharpe Ratio of the buy & hold strategy is 0.01, indicating that after taking risk and brokerage fees into account, buying this stock at the beginning of 2017 and selling at the end of 2018 does not really give an investor any significant return. The very low Sharpe Ratio (0.01) means an investor almost does not gain or lose after taking risk into account, even though there is a profit. Therefore, it is not a worthy investment. As can be seen in Figure 6-4 on page 237, the last price of the testing period is very similar to the price at the beginning.

These results from the buy & hold strategy in table 6.2 will be used for comparison with the results from our model in the following section.

6.2.2.3 Results from our model

- **The best results**

  Our model performances on MCD, TSCO and CARR are shown in Figure 6-5, 6-6 and 6-7, respectively.

  Figure 6-5 shows the results of MCD from our trading system. There are four different sub-graphs. The top graph shows the closing price, and the periods of buying are shown in green. The second graph represents the signals during the testing period. A signal of 1 means buy or continue to hold and -1 means sell the existing positions, while 0 means take no action. The buy signals in this graph correspond to the periods of having this stock in the portfolio on the top graph.

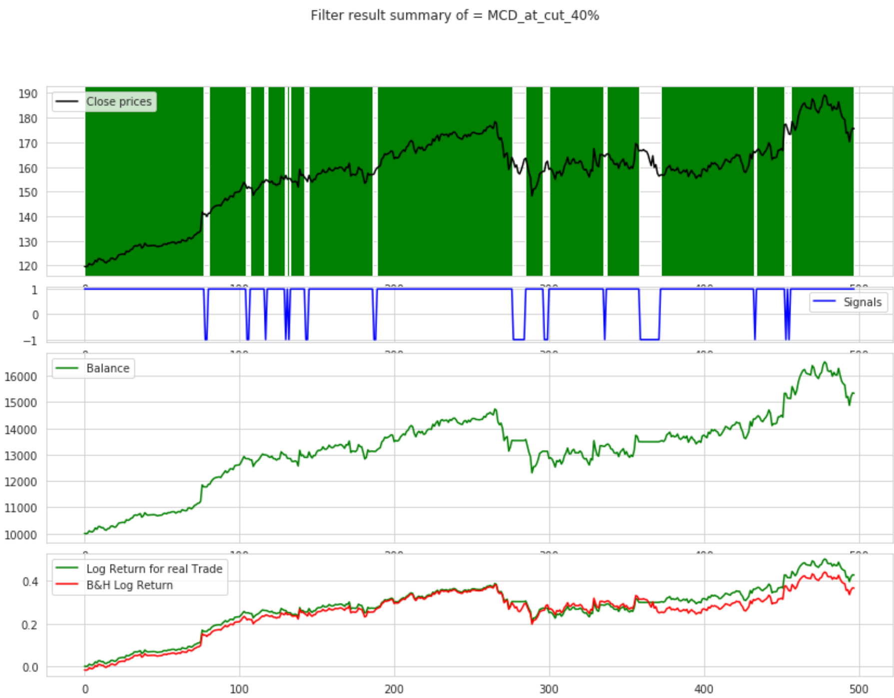


Figure 6-5: Final result of MCD from the testing period of 2017 to 2018

The third graph shows the balance throughout the testing period. Brokerage fees are applied. The balance starts from £10,000 and ends with £15,335 which means gaining over 50% profit. The bottom graph illustrates the comparison of logarithm returns from the buy & hold strategy (red) and our trading system (green). Brokerage fees are not applied in this graph. It can be seen that our system performs a little better than the buy & hold strategy almost throughout the testing period, and especially at the end of testing. It can be seen that even though this stock performs, very well which makes it hard for other strategies to overcome, our model is able to win over buy & hold.

As for TSCO, the result can be seen in Figure 6-6. The second graph shows the balance throughout the testing period. Brokerage fees are applied in this graph. The balance starts from £10,000 and ends with £11,669.7 which means about 16% profit is achieved. This is not bad given this dataset is very fluctuated and could have been difficult for many machine learning algorithms to predict. The bottom graph illustrates the comparison of logarithm returns from the buy & hold strategy (red) and our trading system (green). Brokerage fees are not applied in this graph. It can be seen that our system performs better than the

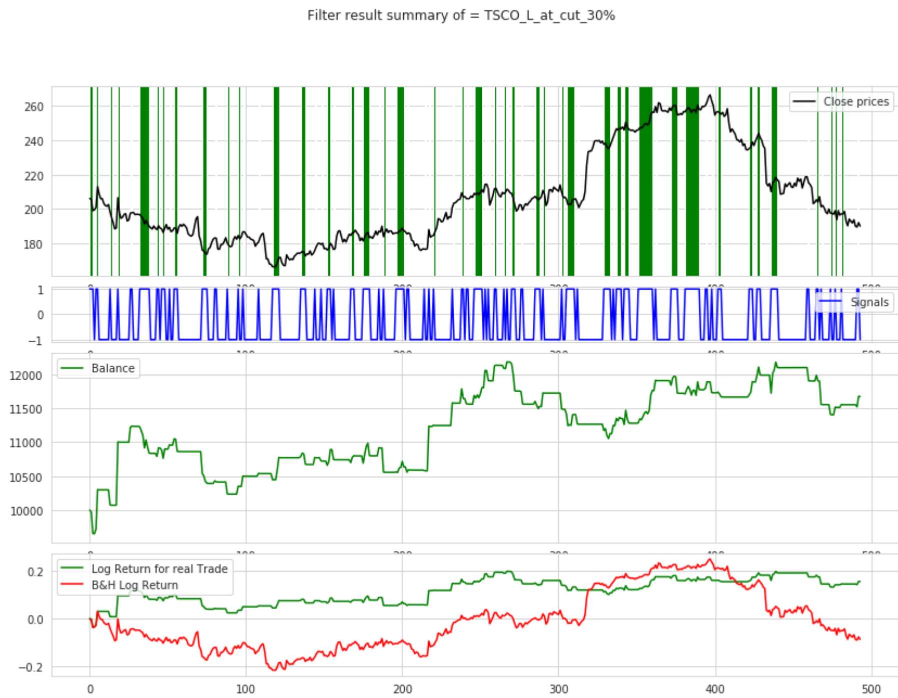


Figure 6-6: Final result of TSCO from the testing period of 2017 to 2018

buy & hold strategy almost throughout the testing period, except for the period near the end of the testing. As can be seen, from about the 310th day until the 410th day, our model's performance is worse than the buy & hold because it could not catch the big increase at the beginning of this period. However, even though it did not catch this initial rising opportunity, it seems to work well after that and is able to identify multiple increasing prices, then it starts to outperform the buy & hold again after day 410.

Even though the model does not perform as well for the dataset as it does for MCD, it does very well in terms of risk control. The bottom graph compares the return of our model and the buy & hold, and it can be seen that while the buy & hold's log return continues to drop throughout the first 220 days, our model is able to maintain profitability. This can also be seen in the Sharpe Ratio of our model, which is 0.59, much better than the buy & hold strategy (-0.19). This means that our model ends up with a profit after taking risk into account.

Finally, the result from CARR shows in Figure 6-7.The third graph shows the balance throughout the testing period after taking brokerage fees into account. The balance starts from £4,000 and ends with £6,848.63; a profit of about 70%.

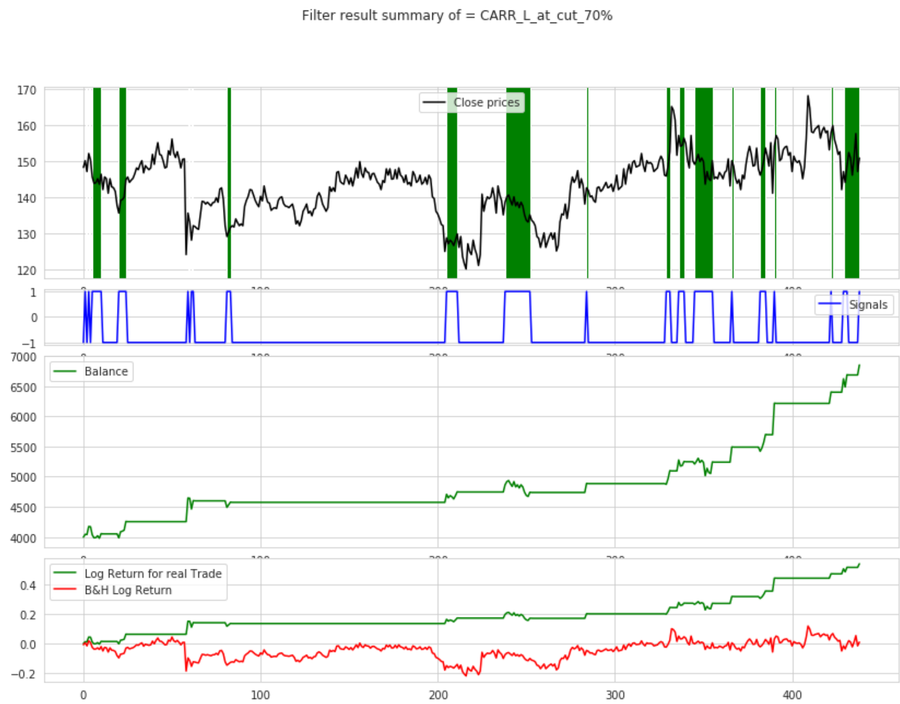


Figure 6-7: Final result of CARR from the testing period of 2017 to 2018

Gaining 70% within such few trades (37 times over 2 years) can be considered a very good performance. The bottom graph illustrates the comparison of logarithm returns from the buy & hold strategy (red) and our trading system (green). Brokerage fees are not applied in this graph. It can be seen clearly that our system performs much better than the buy & hold strategy throughout the period of testing and ends up with a much higher profit.

All result comparison between the performance of the buy & hold strategy and our model can be seen in table 6.3. All results in this table are computed after taking a 3% risk-free rate into account. In this comparison, we have selected the results when starting with the initial funds of £10,000, £10,000 and £4,000 for MSC, TSCO and CARR, respectively, as they provided the best performance. Then, in the next section, the results when starting with different amounts of capital will be shown.

The comparison result of MCD after taking 3% risk-free rate into account in Table 6.3 shows that the Sharpe Ratio from our model is 1.19, which is greater

Stock	Strategy	Sharpe Ratio	Profit (%) (%)	STEDV	Max DD (%)	Acc (%)	Fee £
MCD	Buy & Hold	0.95	38.14	0.174	-16.87	-	£7.58
	Our system	1.31	53.36	0.166	-16.49	47.27	£432.06
TSCO	Buy & Hold	-0.32	-14.49	0.231	-28.79	-	£7.58
	Our system	0.59	16.7	0.133	-9.27	53.73	£511.65
CARR	Buy & Hold	-0.06	-5.17	0.386	-23.08	-	£7.58
	Our system	1.98	71.22	0.157	-5.44	63.13	£140.23

Table 6.3: Comparison between buy & hold and our system testing on MCD, TSCO and CARR during 2017-2018 (not taking risk-free rate into account)

than the Sharpe Ratio of the buy & hold strategy (0.95). The profit from our model is 47.27%, which is also better than the buy & hold, which provided only 38.14% profit. The annual volatility represents the risk of the trading. Our model offers less risk and also a smaller drawdown. All of these results are calculated after taking the brokerage fee into account.

As for the result of TSCO, it can be seen clearly that our model outperforms the benchmark with a Sharpe Ratio of 0.37 and 10.61% profit, compared to the benchmark Sharpe Ratio of -0.32 and an -14.49% loss. This supports that our model performs well in terms of risk control, which is evident from the drawdown and volatility values. Our model has a much smaller maximum drawdown (-9.27) than the buy & hold (-28.79) which means that the furthest drop in the balance of our model is much less than the buy & hold. This means our model is able to recover from losses much easier and quicker. As for the volatility, our model also provides less volatility (0.133) than the buy & hold (0.231), which means that our model's profit is more stable and therefore less risk than the buy & hold strategy.

Finally, it can be seen clearly from the result of CARR that our model perform much better than the benchmark, achieving a Sharpe Ratio of 1.78 (compared to -0.05, the Sharpe Ratio of the benchmark). Moreover, the model also provides a much higher profit of 65.13%, compared to the benchmark's loss of -5.17%. Not only is our model better than the buy & hold strategy in terms of profit and Sharpe Ratio, it is also better at controlling the risk, as can be seen from the values of annual volatility and drawdown. The maximum drawdown of our model is -5.44, which is much smaller than the drawdown of buy & hold (-23.08). This means that the maximum drop of our model is only 5.44%, which makes it much easier for our model to recover from the loss and return to profit. As for the volatility, our model also provides less volatility (0.157) than the buy & hold (0.386) which means that our model's profit is more stable and less risky.

- **Results when having different initial funds**

As mentioned, our trading system aims to be used by the individual trader, therefore different starting amounts of funds should be taken into account. Therefore, we have experimented using different amounts of initial funds, ranging from £1,000 to £10,000. The results of MCD, TSCO and CARR can be seen in Figure 6-8, 6-9 and 6-10, respectively.

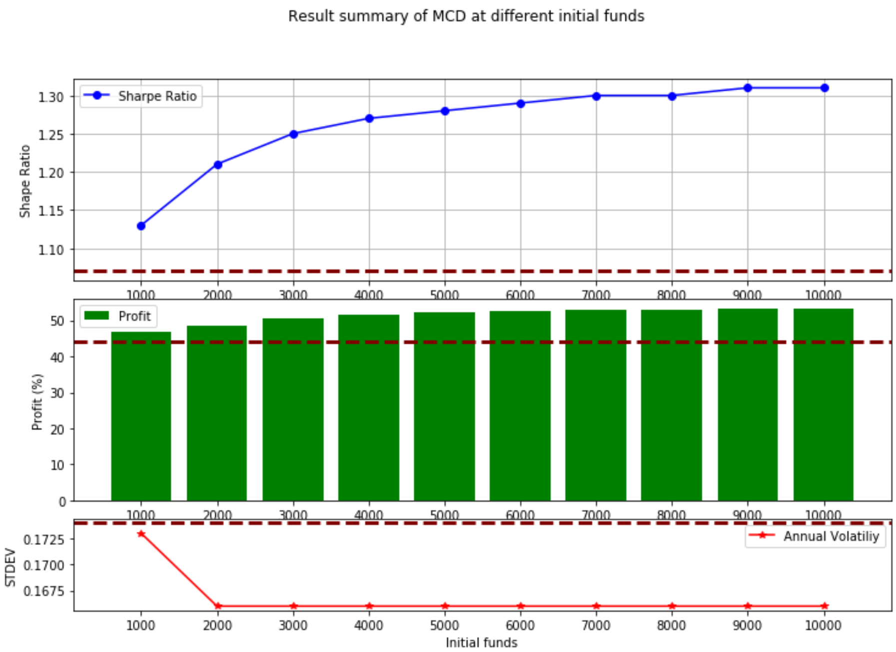


Figure 6-8: Result summary of MCD at different initial funds

In Figures 6-8, 6-9 and 6-10, the top graph shows the Sharpe Ratio from our system when starting with different amounts of funds. The Y-axis is the Sharpe Ration, while the x-axis shows the amount of funds, which ranges from £1,000 to £10,000. The red dashed-line shows the Sharpe Ratio of the buy & hold strategy, here considered as a benchmark. The middle graph shows the profit from our model starting with different amounts of funds. Finally, the volatility or risk can be seen in the bottom graph.

From the result of MCD in Figure 6-8, the top graph clearly shows that our model

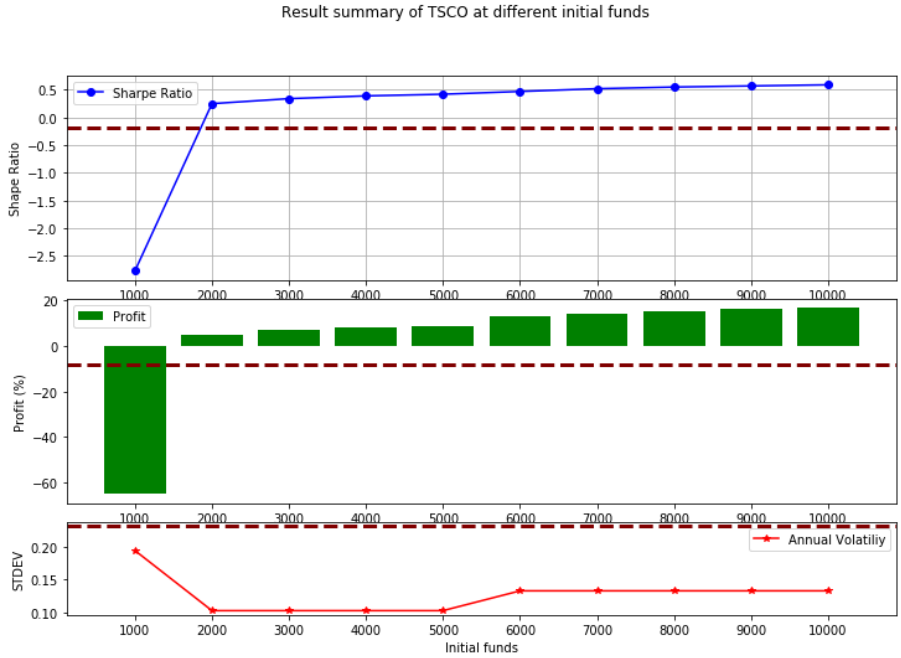


Figure 6-9: Result summary of TSCO at different initial funds

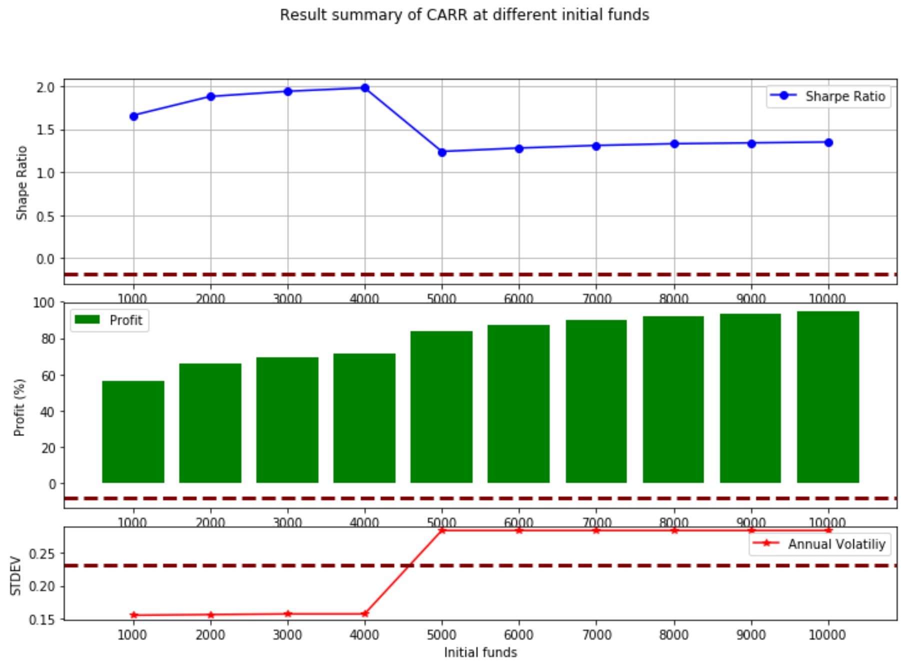


Figure 6-10: Result summary of CARR at different initial funds

provides a better Sharpe Ratio regardless of the initial amount of funds. As for the profit, it can be seen from the middle graph that our model also provides better profit than the buy & hold strategy. Finally, considering the risk from the bottom graph, our model also has less risk than the buy & hold strategy. In this graph, every value of volatility from our model is lower than the volatility of the buy & hold strategy, which is shown by the red dashed-line.

As for the result from TSCO, the top graph of Figure 6-9 shows that our model provides a better Sharpe Ratio on the condition of having more than £2,000 in initial funds. This graph clearly shows that when the starting amount of capital is only £1,000, the model is not able to profit as there are few opportunities and it is difficult to profit in highly-fluctuated situations. Also, with less money to start with, the brokerage fee will have a greater effect on the profit. However, in this case, adding another £1,000 into the initial funds leads to making a profit with the model, which can be seen from increase of the Sharpe Ratio. Considering the profit, it can be seen from the middle graph that our model also provides better profit than the buy & hold strategy, when starting with initial capital of £2,000 and upwards.

Our model also demonstrates less risk than the buy & hold strategy, as can be seen in the bottom graph. In this graph, every value of volatility is lower than the volatility of the buy & hold strategy, which is shown by the red dashed-line. However, increasing initial funds from £6,000 to £10,000 seems to increase the volatility of our model as the model tries to increase the number of trades in order to increase the opportunities to make a profit. In this case, it turns out that some of the trades our model made were incorrect. However, overall, most of them were correct, that is why our model still ends up making a profit.

The result of CARR shows in Figure 6-10 which can be seen from the top graph that our model provides a better Sharpe Ratio than the buy & hold strategy regardless of initial funds. Starting with only £1,000, the model already provides a very high Sharpe Ratio (1.66). The Sharpe Ratio continues to increase with funds up to £4,000, there achieving the maximum Sharpe Ratio of 1.98. In this case, increasing initial funds beyond this point does not increase the Sharpe Ratio further.

This is an unusual situation. Mostly, increasing money increases the Sharpe Ratio. However, our model tries to trade more often as initial funds increase. The number of trades increases from 37 to 161. The volatility also almost doubles (from 0.157 to 0.284). Also, increasing the initial funds from £4,000 to £5,000 also makes the drawdown bigger, from -1.23 to -5.23. This means that the additional trades that our model tries to make include at least one incorrect decision. That trade was unfortunately on a day which saw a big decrease in price. Therefore, even though the amount of funds is £5,000, and the profit increases to 84.07% (see the middle graph), the Sharpe Ratio still decreases, according to the risk from the drawdown of the incorrect decision.

As for the profit, it shows from the middle graph that our model also provides better profit than the buy & hold strategy regardless of the initial amount of funds. Considering the volatility in the bottom graph, our model also has less risk than the buy & hold strategy, as can be seen in the bottom graph. From £1,000 to £4,000 in starting capital, the volatility is lower than the volatility of the benchmark. However, after £5,000, our model provides worse volatility than the buy & hold strategy. This means that investors have to accept more risk when using our model.

Even having provided more risk from £5,000 onward, the profit also increases. It can be said that our model has more risk but also more profit. Therefore, at this point, in order to evaluate whether the risk is worth taking, we have to see the value of the Sharpe ratio. If the value of the Sharpe Ratio is still high, it is still worth investing. From £5,000 onward, the Sharpe Ratio values are still positive. Moreover, the values are more than 1 regardless of the amount of funds. Therefore, the model still performs well and the investment worth making for every amount of starting capital.

6.2.3 All Results and Comparisons

Having looked at a few examples of the results in the previous section, this section presents the results from all datasets in our experiment together with the benchmarks for comparison.

6.2.3.1 All Results

Table 6.4 shows the results from our model on every stock mentioned in section 6.2.1 on page 232 after taking a 3% risk-free rate into account. For each stock, the Sharpe Ratio and profit starting with initial funds from £1,000 to £10,000 will be shown. The first column lists the stock's symbol on the market. If the return of the buy & hold strategy was positive, the symbol will be shown in black. On the other hand, if the buy & hold is negative, the symbols are shown in red. There are five positive stocks (MCD, GOOG, CARR, MACF and OXIG) and six negative stocks (SXS, MKS, TSCO, COST, FERG, CARR and D4T4) in this table. As for the results from our model (Sharpe Ratio, profit and accuracy), columns are coloured in green if our model outperforms the buy & hold strategy (benchmark), otherwise columns are shown as normal. The performance from the buy & hold strategy, here considered as a benchmark. Please note that the accuracy of buy & hold strategy means the percentage of the days which the price increases.

Table 6.4: Results from our model after taking risk-free rate into account at different initial funds, evaluated over two years (2013-2014)

Stock	Metrics	Initial funds (£)										
		BH	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
MCD	Sharpe Ratio	0.95	0.95	1.03	1.07	1.09	1.1	1.11	1.12	1.12	1.12	1.13
	Profit	38.14	40.68	42.37	44.41	45.43	46.04	46.45	46.74	46.96	47.13	47.27
	Acc	-	100	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14
GOOG	Sharpe Ratio	0.52	0.2	0.3	0.33	0.35	0.36	0.36	0.37	0.37	0.37	0.39
	Profit	28.28	10	15.21	16.94	17.81	18.33	18.68	18.93	19.12	17.2	17.89
	Accuracy	-	61.54	61.54	61.54	61.54	61.54	61.54	61.54	61.54	53.33	53.33
SXS	Sharpe Ratio	-0.19	-0.05	0.18	0.25	0.29	0.31	0.32	0.33	0.34	0.1	0.11
	Profit	-9.71	-2.37	8.2	11.72	13.49	14.54	15.25	15.75	16.13	4.76	5.12
	Acc	-	50	50	50	50	50	50	50	50	50	50
MKS	Sharpe Ratio	-0.94	-4.77	0.25	0.56	0.43	0.53	0.6	0.64	0.68	0.71	0.73
	Profit	-36.97	-84.04	4.31	13.49	11.92	14.98	17.02	18.48	19.57	20.42	21.1
	Acc	-	51.72	48.28	44.64	42.47	42.47	42.47	42.47	42.47	42.47	42.47
TSCO	Sharpe Ratio	-0.32	-2.92	-0.04	0.05	0.1	0.12	0.25	0.29	0.32	0.35	0.37
	Profit	-14.49	-71.09	-0.92	1.06	2.05	2.65	6.97	8.27	9.24	10	10.61
	Acc	-	49.48	33.33	33.33	33.33	33.33	53.73	53.73	53.73	53.73	53.73
COST	Sharpe Ratio	-0.36	-0.55	-0.05	0.17	0.27	0.34	0.38	0.41	0.43	0.44	0.46
	Profit	-16.9	-24.28	-1.68	4.49	7.58	9.43	10.66	11.55	12.21	12.72	13.13
	Acc	-	0	54.17	54.17	54.17	54.17	54.17	54.17	54.17	54.17	54.17
FERG	Sharpe Ratio	-0.26	-5.09	-0.16	-0.25	-0.16	-0.07	0.01	0.07	0.11	0.14	0.16

Continued on next page

Table 6.4 – *Continued from previous page*

Stock	Metrics	Initial funds (£)										
		BH	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Profit	-10.3	-85.68	-6.31	-9.58	-6.21	-2.43	0.32	2.29	3.76	4.91	5.83
	Acc	-	44.07	40	43.4	43.4	48.04	48.04	48.04	48.04	48.04	48.04
CARR	Sharpe Ratio	-0.06	1.47	1.68	1.75	1.78	1.13	1.17	1.2	1.22	1.23	1.25
	Profit	-5.17	50.22	60.16	63.47	65.13	77.98	81.48	83.99	85.87	87.33	88.5
	Profit	-	44.44	44.44	44.44	44.44	57.5	57.5	57.5	57.5	57.5	57.5
MACF	Sharpe Ratio	0.27	-0.65	0.2	0.33	0.39	0.43	0.45	0.47	0.48	0.49	0.5
	Profit	10.47	-23.22	5.83	11.23	13.93	15.55	16.63	17.41	17.99	18.44	18.8
	Acc	-	58.7	63.64	63.64	63.64	63.64	63.64	63.64	63.64	63.64	63.64
OXIG	Sharpe Ratio	0.21	-0.43	0.36	0.56	0.65	0.7	0.74	0.76	0.78	0.79	0.8
	Profit	18.23	-17.41	24.76	40.77	48.78	53.58	56.78	59.07	60.78	62.12	63.18
	Acc	-	40.43	51.96	51.96	51.96	51.96	51.96	51.96	51.96	51.96	51.96
D4T4	Sharpe Ratio	-0.1	-0.47	0.23	0.41	0.5	0.54	0.58	0.6	0.62	0.63	0.64
	Profit	-10.25	-27.6	13.65	27.4	34.28	38.41	41.16	43.12	44.59	45.74	46.66
	Acc	-	50.67	50.67	50.67	50.67	50.67	50.67	50.67	50.67	50.67	50.67

In table 6.4, the buy & hold strategy for every stock is used as benchmark. The symbols are shown in black or red in the first column. If the Sharpe Ratio of a stock is negative, the symbol is red. Otherwise, symbols are shown in black. The first two stocks, MCD and GOOG, perform very well as shown by their respective Sharpe Ratios of 0.95 and 0.52. This good performance also can be seen in the high profit of 38.14% and 28.28%. Another two stocks, MACF and OXIG, also have positive Sharpe Ratios which means that these stocks achieved positive returns from the buy & hold strategy. The rest of the stocks which are shown in red did not perform very well and ended up with a loss and negative Sharpe Ratio.

Based on this comparison, we separate stocks into three groups, as follows:

- **Excellent (3 stocks)** : Stocks that outperformed their buy & hold regardless of the amount of initial funds. There are three stocks in this group, which are MCD, SXS and CARR.

- **Good (7 stocks)** : Stocks that performed better than buy & hold under the condition of having at least £2,000 initial funds. There are seven stocks in this group, which are MKS, TSCO, COST, FERG, MACF, OXIG and D4T4.

- **Fair (1 stocks)** : The stock that could not overcome the buy & hold strategy. There is only GOOG in this group. We call this group *Fair* as even though our model could not outperform the benchmark, the results are still good. All Sharpe Ratios are positive as well as the profit at every amount of initial funds. This situation can happen sometimes for stocks with an upward trend as the buy & hold already performs very well. Therefore, it is not easy to overcome with other models.

6.3 Experiments on FTSE 100

In this section, we experiment further on more stocks from a wider range of economic sectors. The objective of this experiment is to ensure that our system works effectively on stocks with many different characteristics. The group of stocks we have selected are stocks in the Financial Times Stock Exchange 100 index, also called the FTSE 100 index, as it is a well-known index which comprises a variety of stocks from different sectors.

6.3.1 FTSE 100 index

The FTSE 100 index is composed the 100 stocks with the highest market capitalisations listed on the London Stock Exchange. Stocks in the FTSE 100 operate in many different sectors which is very good for our experiments. The predictors within our model are primarily based on movements in the share price, so-called price action or technical indicators, as opposed to fundamental data to do with the operations of the underlying businesses. However, a large influence on the behaviour of the share price, trending or ranging and the extent to which it does so, is the nature of the sector a business operates in. Some industries are considered cyclical, meaning they profit when the economy is doing well and decline in worse times. Some industries are growing, others reclining. Meanwhile some, such as utilities and consumer staples, are considered defensive as they do not change much. Testing our model across shares in the FTSE 100 index allows us to identify the share price behaviour of different sectors and see how our model performs under these influences. FTSE 100 datasets are available for free from the Yahoo Finance website [38].

6.3.1.1 FTSE 100 index separated by business sectors

There are many different types of business in the FTSE 100 index. The proportion of stocks in each sector can be seen in figure 6-11.

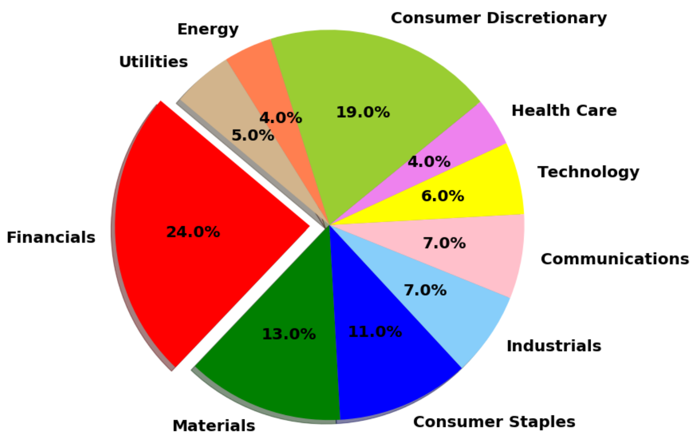


Figure 6-11: FTSE 100 data sectors

From figure 6-11, it can be clearly seen that the biggest proportion (24%) of stocks in the FTSE 100 are in the financial sector, reflecting the large impact of the financial services sector on the UK economy. Example companies in this sector are Hargreaves Lansdown plc (HL.L), HSBC Holdings plc (HSBA.L) and Prudential plc (PRU.L). The second biggest sector in the FTSE 100 is the consumer discretionary sector, which accounts for 19%. This sector relates to goods and services that are classified as non-essential by consumers but favourable if they have sufficient income. Consumer purchases in this sector are influenced by the economic situation. Consumer discretionary goods include automobiles, leisure, entertainment and durable goods. Some examples are Burberry Group plc (BRBY.L), Flutter Entertainments plc (FLTR.L) and International Consolidated Airlines Group (IAG.L).

Another two sectors representing more than ten percent share of the index are Materials and Consumer Staples (13% and 11%, respectively). The materials sector is the category of stocks from companies involved in the development and processing of raw materials. This sector also includes mining, metal refinery, forestry and chemical products. Examples of the companies in this sector are Anglo American (AAL.L), Johnson Matthey plc (JMAT.L) and BHP Group plc (BHP.L). As for consumer staples, this sector relates to essential products in everyday life, for example food & beverages, hygiene and household products. These are the products that most people are unable to cut out of their lives, regardless of their financial situation. However, tobacco and alcohol are also included in this sector. Examples of companies in this sector are Coca Cola HBC AG (CCH.L), WM Morrison Supermarkets plc (MRW.L) and British American Tobacco (BATS.L).

Apart from those large sectors mentioned above, there are other smaller sectors included in the FTSE 100, such as Communication (7%), Industrials (7%), Technology (6%), Utilities (5%), Energy (4%) and Health Care (4%). The complete list of companies in the FTSE 100 index and sector information can be seen in appendix A Table A.1 on page 298. It should be noted that sources can differ on sector labels and constituents. These sector groupings are taken from the London Stock Exchange website [35].

6.3.1.2 FTSE 100 index grouped by performances during testing period

Stocks in the FTSE 100 have delivered both positive and negative returns, which can be seen in figure 6-12.

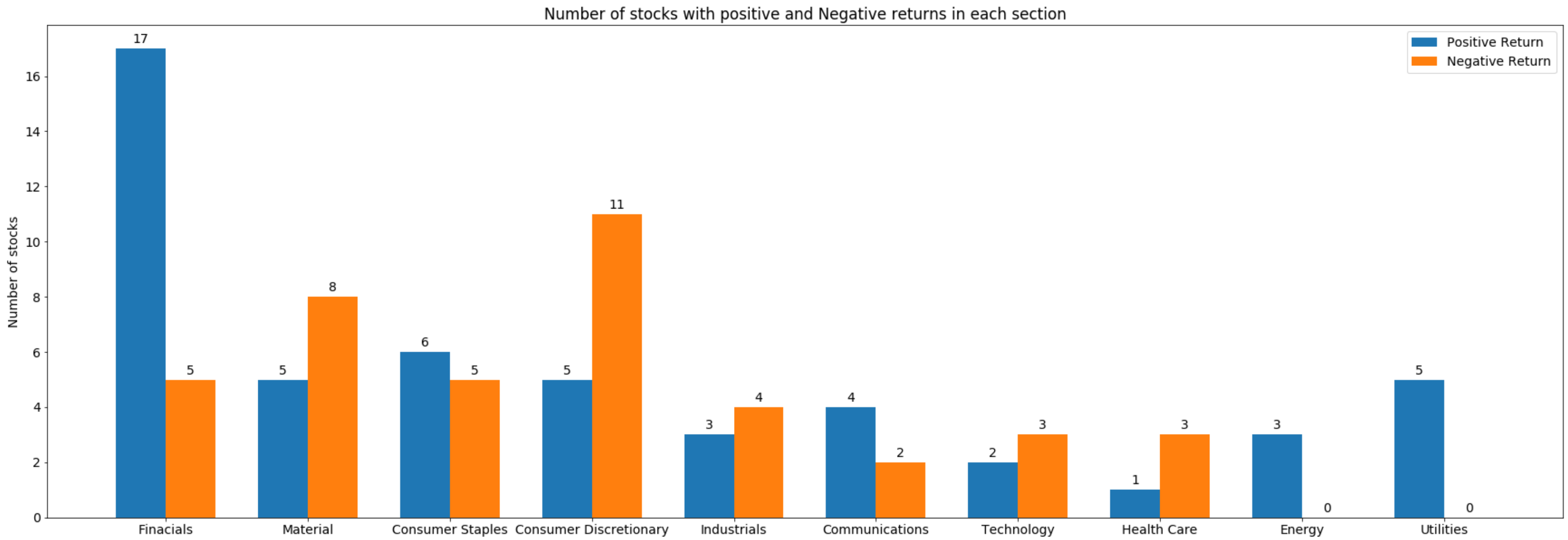


Figure 6-12: Number of stocks with positive and negative returns in FTSE100

Figure 6-12 illustrates the number of positive and negative return stocks within each sector of the FTSE 100. These returns are considered within the testing time frame from the beginning of 2017 to the end of 2018. The x-axis indicates the sector in the index, while the y-axis shows the number of stocks. It can be seen that there are a lot more positive than negative stocks in the financial sector (the ratio is 17:5), which indicates a bullish trend in this sector. On the other hand, for the consumer discretionary sector, there are more negative stocks (11) that positive ones (5), which indicates that only about half of the stocks in this sector are making a profit. As for energy and utilities, there are only positive return stocks within these sectors. However, there are only 3 and 5 stocks in the energy and utilities sectors respectively that have made their way into the FTSE 100. Therefore, one can assume that compared to other sectors, they are not very well-performing businesses.

6.3.1.3 FTSE 100 index separated by the length of becoming public

There are some stocks in the FTSE 100 for which data is shorter than others as they became public at a later stage. For example, Lloyds Banking Group plc (LLOY.L) is one of the oldest banks in the UK. They became public on the stock market on the 28th of December 1995. Therefore, we have plenty of data to train our model. Meanwhile, Just Eat plc (JE.L) is one of the most recently successful businesses, which only became public on the 3rd of April 2014. Therefore, we have very limited data on this stock.

For the reason above, we have divided the data into four different groups as shown in figure 6-13. Firstly, we assigned the stocks that were listed on the stock exchange before 2000 to be in group A. Secondly, we assigned stocks that were listed on the market after 2003 to group B. Thirdly, the stocks that were listed on the stock exchange after 2010 were placed in group C. Lastly, stocks that have been listed very recently, after 2014, were classified as group D.

The numbers of stocks in each group is shown in figure 6-13. In this research, we divided our data into three different groups as shown in figure 4-4 on page 91. There are training, validation and test sets which cover 2000-2014, 2015-2016, 2017-2018, respectively. For all of the stocks, the validation and test sets are 2015-2016 and 2017-2018. However, the length of training sets vary as detailed below:

- **Group A** : Stocks that became public before 2000. There are 76 stocks (31 positive, 45 negative) in this group. The training set ranges from 2000 to 2014. Therefore, we have more than 10 years of training data for the stocks in this

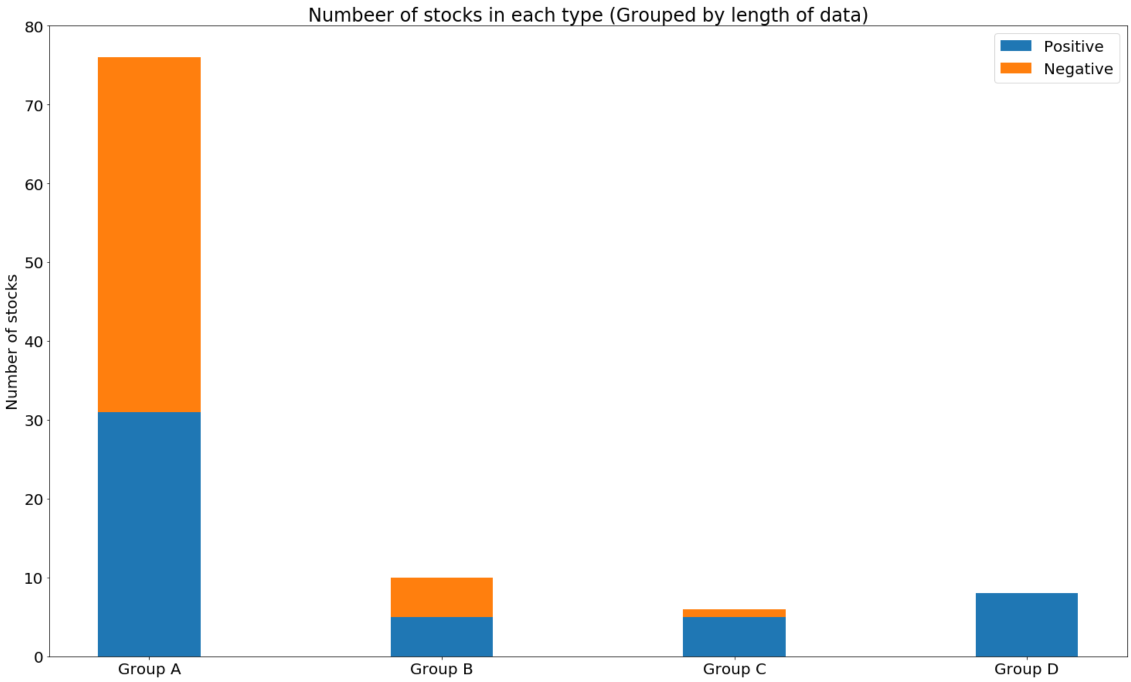


Figure 6-13: Number of stocks which overcome their Buy and Hold strategy

group.

- **Group B** : Stocks that became public between 2004 and 2010. There are 10 stocks (5 positive, 5 negative) in this group. The training set ranges from the earliest point available from the data. Therefore, for the stocks in this group, we have at least 5 years of training data but less than 10 years.

- **Group C** : Stocks that became public between 2011 and 2013. There are 6 stocks (5 positive, 1 negative) in this group. We have very limited training data, less than 5 years, for the stocks in this group, but it is still possible to test with our model.

- **Group D** : Stocks that became public after 2014. There are 8 stocks (8 positive, 0 negative) in this group. As they are very new companies to be listed on the stock exchange, we have not got enough data to work with. Therefore, we have excluded these 8 stocks from the evaluation process.

The full list of all companies which are listed on the London Stock Exchange and included in the FTSE 100 index can be seen in table A.1 in appendix A section A.1 on page 298. Column *Symbol* shows the symbol of each stock in the index and the year which data begins according to the four groups of data mentioned earlier. The second column contains the companies' full names and the last column shows the sector into which each stock is classified.

6.3.2 Experiment and result analysis

In the previous section, we provided details of the stocks within the FTSE 100. In this section, how the experiment was prepared and the results of our model on the FTSE 100 will be analysed.

6.3.2.1 Experiment

In preparation for the experiment, the stocks in group D which are shown in figure 6-13 on page 256 were excluded as they became public after 2014, which means the data is too short to be analysed by our model. After excluding the stocks in group D, we have 92 stocks left, 41 with a positive return over the testing period from 2017 to 2018, and 51 with a negative return, as shown in figure 6-14.

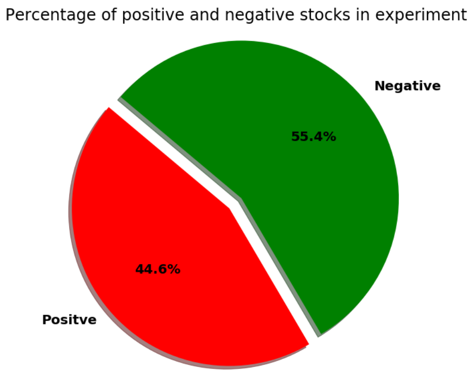


Figure 6-14: Percentages of positive and negative stocks in experiment

Table 6.5 shows that the excluded stocks comprised two stocks from the financial sector, three from consumer discretionary and one stock from the communication, technology and utilities sectors. It can be seen from the table that 3I GRP (III.L) became public in 2001. The reason it was excluded is not because of the length of the data as with the others, rather because we could not obtain a complete dataset. Therefore, we will not experiment on III.L either.

Sector	Stocks	Year of becoming public (%)
Financials	III.L, SMT.L	*2001, 2017
Consumer Discretionary	FLTR.L, JD.L, TUI.L	2019, 2018, 2014
Communication	AUTO.L	2015
Technology	JE.L	2014
Utilities	RDSA.L	2018

Table 6.5: Stocks that are excluded from experiment

In the following section, we will analyse the result from the 92 remaining stocks in order to evaluate the performance of our model. The full details of these results are shown in appendix A Section A.3 on page 307.

6.3.2.2 Results from all sectors

In this section, the overall result which is calculated from the average of every stock in every sector will be shown. This is designed to determine whether we should keep all of the datasets or whether some of them should be excluded, for example group C which has shorter datasets than the other groups. The results are shown in three different graphs, as in figure 6-15.

There are two sub-figures in figure 6-15, the top shows the average Sharpe Ratio and the bottom shows the average profit from every dataset at each different amount of initial funds. As can be seen, the results from these two graphs have very similar trends. Both Sharpe Ratio and profit increase with the amount of initial funds. The most important result that we focus on is the Sharpe Ratio from the top graph as this value takes both profit and risk in to account.

The top figure shows three different Sharpe Ratio values which are calculated from all stocks (blue), all stocks excluding group C (orange) and all stocks excluding groups B and C (green). It can be seen clearly that excluding stocks in group C (orange) does not result in a significant change to the overall result (blue). Similarly, the graph that excludes groups B and C (green) also provides similar results to overall (blue). It can be seen clearly that even though data for the stocks in groups B and C are shorter, they do not affect the overall result. Therefore, all stocks will be included in the following sections.

It can be safely said from these results that our model is able to profit on the condition that investors have initial funds of at least £2000 and that the profit will increase when

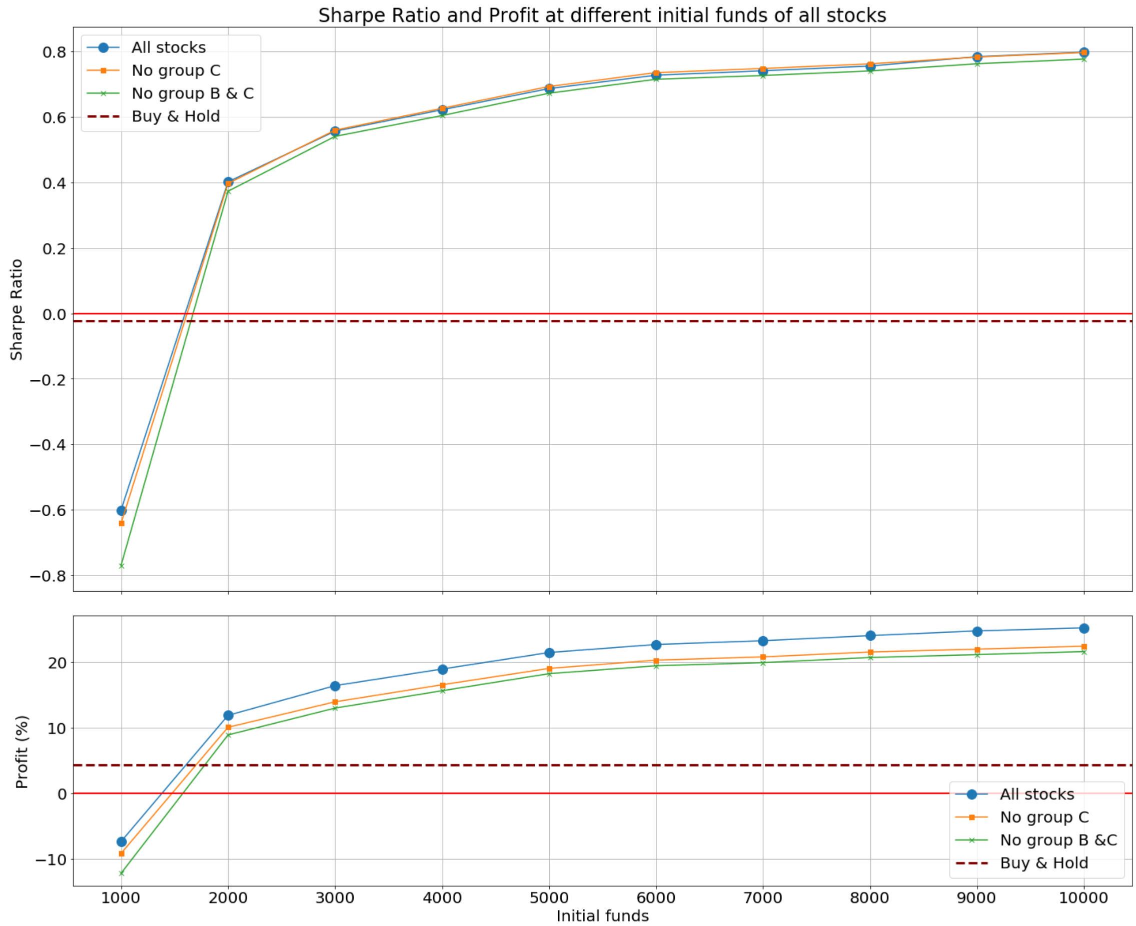


Figure 6-15: Sharpe Ratio and profit of all stocks at different initial fund

the amount of initial funds increases.

The result when starting with only £1000 capital can be seen in the first group of bars in figure 6-21 on page 269. There four bars represent the percentage of stocks that overcome the buy & hold strategy (blue), the percentage of stocks that overcome the market index (orange), the percentages of stocks that end up with a profit (green) and the percentage of stocks that end up with a loss (red). It can be seen that if an investor starts with £1000, the probability of having a profit is 58.7% and the probability of getting a loss is 41.3%. However, even though there is more chance to profit, it still is not worth an investment when the risk is taken into account, and the average Sharpe Ratio is a negative in figure 6-15.

Figure 6-16 shows the performance of our model on separate groups only to gain more understanding of the individual group's performance.

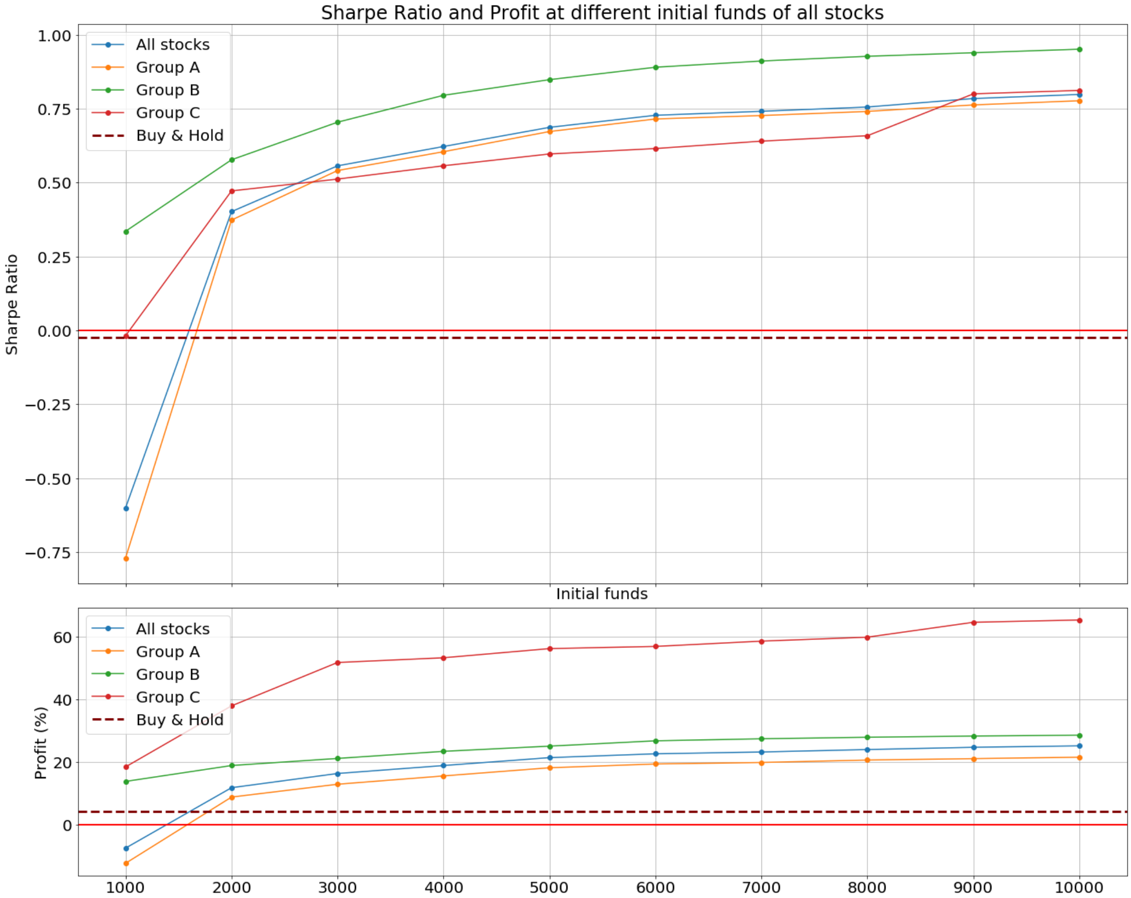


Figure 6-16: Sharpe Ratio and profit of each group at different initial fund

There are two sub-figures in figure 6-16. The stocks in group C are new stocks which have recently become available to the market and have performed very well, hence being selected to join the FTSE 100 very quickly after arriving to the market. Therefore, as can be seen in the bottom figure, the red graph which shows the average profit of stocks in group C is the highest of all. However, looking at the top graph, group C does not perform very well in terms of the Sharpe Ratio. This means that even though they are the most profitable stocks, it is not guaranteed that their performance is the best after adjusting for their risk. On the other hand, stocks in group B seem to perform better than the others, as can be seen by the high Sharpe Ratio in the top figure and the stable trend of their profit in the bottom graph. However, there are only a very small number of stocks in groups C and B (only 6 and 10 stocks) out of the 92 stocks in the experiment. Therefore, their performances do not affect the overall result, as mentioned earlier.

6.3.2.3 Results from individual sectors

In this section, we analyse the results by sector in order to see how well our model works in specific business areas. The results separated by sector can be seen in Figure 6-17. This Figure shows the average Sharpe Ratio of each sector, which can be seen in each bar, with initial funds of £1,000, £2,000, £3,000, £9,000 and £10,000. We show the results from several different amounts of funds in order to see the results more clearly. The additional detail of the result from individual sector can be seen in Appendix A Section A.3.0.1 on page 322

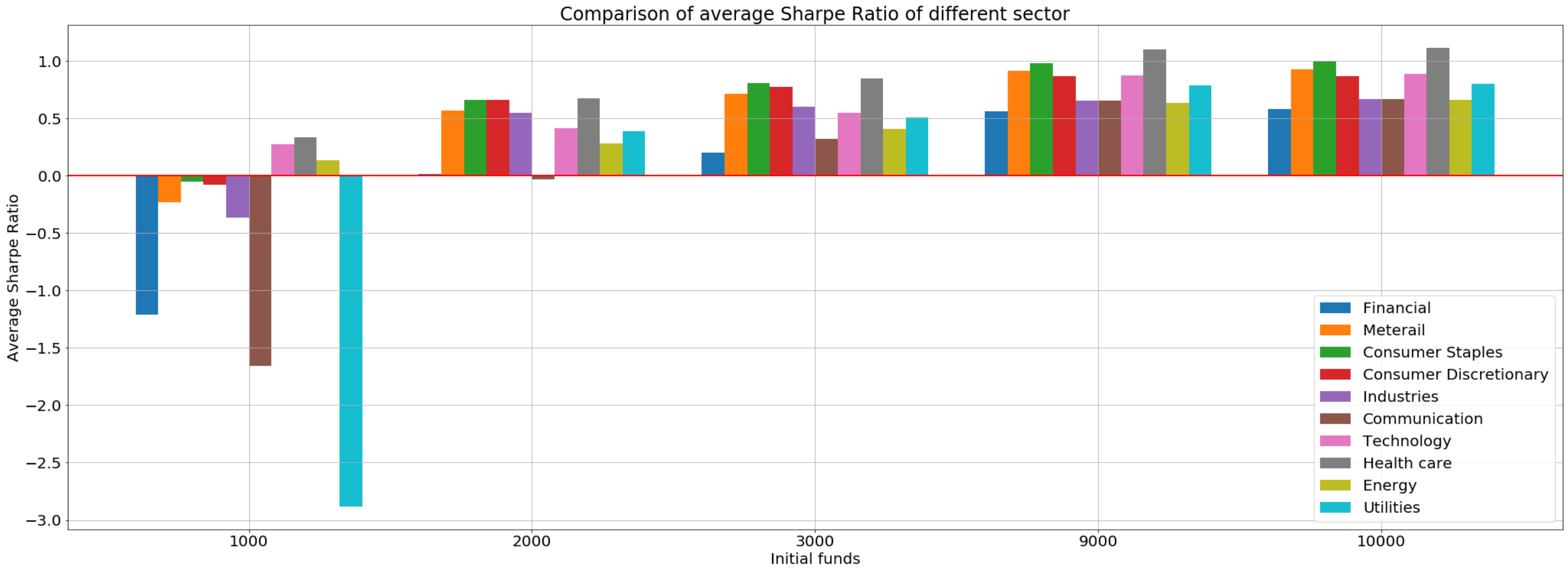


Figure 6-17: Sharpe Ratio and profit of individual sector at different initial fund

In Figure 6-17, the x-axis represents different amounts of initial funds (£1,000, £2,000, £3,000, £9,000 and £10,000), while the y-axis shows the average Sharpe Ratio. There are 10 different bars in each result, which represent the Sharpe Ratio of each business sector. It can be seen from the overall results that the average Sharpe Ratio increases with the amount of initial funds.

Our model seems to do very well in the health sector, which has the highest Sharpe Ratio regardless of initial funds. There are a number of reasons for this, firstly that stocks in healthcare sector have low volatility (the volatility of this sector is very low compared to the other). Secondly, stocks in this sector would have profited well if bought and held for the duration of the test period (having the highest average profit and Sharpe Ratio among the sectors). Finally, the average drawdown of this sector was very small, the second smallest out of all sectors. This combination of attributes meant our model provided a very good result for this sector.

On the other hand, our model did not perform very well on the financial and communication sectors, as these two sectors include the majority of low performance stocks. Moreover, stocks in the financial sector often decreased sharply, resulting in a large average drawdown. For our model, it is difficult to make a profit with stocks that perform poorly (stocks with downward trends) as our model can only go *LONG*, meaning buy in expectation of a price increase, or stay in cash when it is expected the price will decrease. It cannot *SHORT* in order to profit from a stock price decline.

Starting with initial capital of £2,000 or upwards, our model performs very well in the utilities sector, even though the buy & hold result of this sector is very bad. This sector has the lowest Sharpe Ratio and profit. Moreover, its drawdown is very large. The only positive in this sector is that the stocks are not volatile. This shows that our model can work well with badly performing stocks if the prices are not too volatile and even if it has to deal with big drops a few times, it will profit eventually.

Although our model's results are very good from initial funds of £2,000 onward, it seems to perform badly when starting with £1,000. This is because when starting with less capital, the performance will be more greatly affected by the brokerage fees. Also, the profit will mostly rely on the buy & hold performance of that individual stock, since fewer trades will be made. As can be seen, when starting with only £1000, the model tends to lose money in almost every sector. The biggest loss happens in the utilities sector as the stocks in this sector already perform poorly (the average Shape ratio of this group is only -2.7 which is the lowest Sharpe Ratio out of all sectors). Another two sectors that our model performed badly in when the starting capital was £1,000 were communication and financial, for the same reason that the stocks in these two sectors are mostly poorly performing (the Sharpe Ratios of these two sectors are only -1.4 and -1.16).

On the other hand, our model done well in the technology, health care and energy sectors, even with only £1,000 of initial funds. This is because these three sectors' buy & hold performances were very good, resulting in a high Sharpe Ratio and small drawdown.

In summary, our model seems to correlate with the buy & hold strategy and is greatly impacted by the brokerage fee if starting with only £1,000 to invest. With more funds, the model performs much better and is less correlated with the buy and hold result and less affected by the fees. Therefore, having adequate initial funds will give our model the opportunity to perform better without relying on the buy & hold result and being too affected by the brokerage fee.

6.3.2.4 Results on positive & negative stocks

In order to confirm that our model is able to work well with stocks that result in both profit and loss when bought and held, this section illustrates the model's performance on both types of stocks separately. Figure 6-18 shows the performance when using our model with stocks that achieved a positive return over the test period. On the other hand, Figure 6-19 shows the results from the losing stocks. There are two sub-figures in each figure. The top is the average Sharpe Ratio and the bottom shows the average profit. The X-axis is the amount of initial funds.

It can be seen from Figures 6-18 and 6-19 that the performance of our model is consistent. The model profits in both cases with initial funds of £2,000 upwards. The Sharpe Ratio and profit increase when investing more money.

The average Sharpe Ratio and profit of the upward stocks are much better than the downward stocks, but still have the same trend whereby profit increases with the amount of capital to begin with. It is understandable that there should be more opportunities to make a profit on stocks that already offer a positive return, especially when our model employs a *LONG*-only strategy and cannot go *SHORT*. Therefore, we only profit from an increasing price. When the price decreases, the model can only stay in cash and wait for the next opportunity.

The full details of these results are shown in appendix A Section A.3 on page 307.

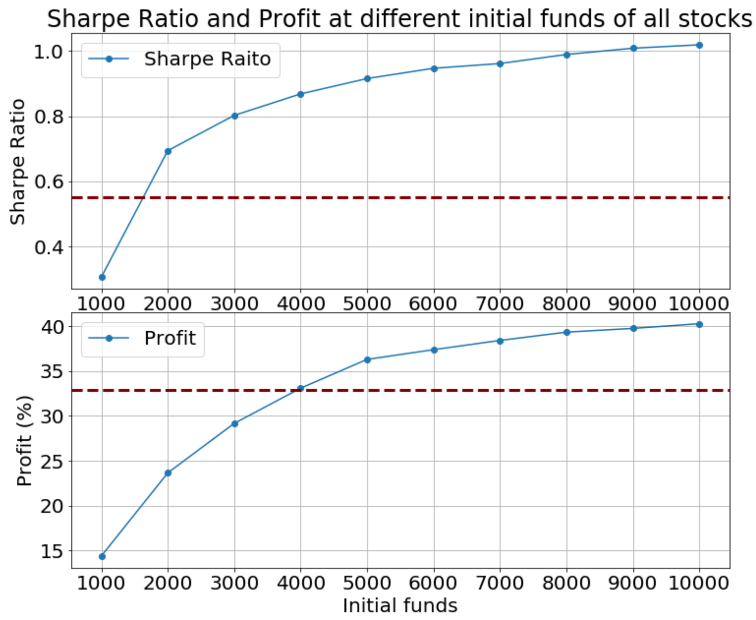


Figure 6-18: Result from positive stocks

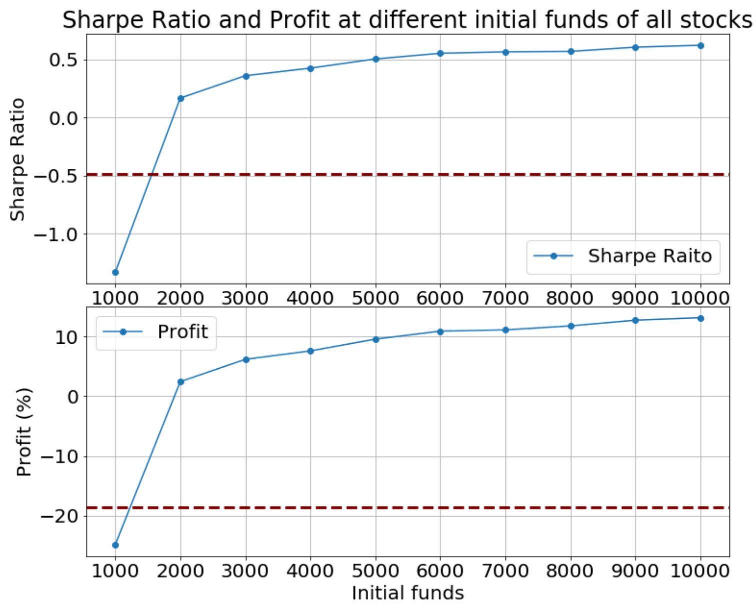


Figure 6-19: Result from positive stocks

To test the statistical significance of our model's improvements over the benchmark strategy, we ran a paired T-test on the Sharpe ratios of these 92 stocks from the FTSE 100 to compare between our scoring system at different levels of initial funds and the benchmark. The p-value of our system at £1000, £2000,£3000, £4000, £5000, £6000, £7000, £8000, £9000, £10000 versus the buy & hold were 0.000002, 0.18991, 000039, 1.29092E-07, 1.77812E-12, 7.51923E-14, 1.22771E-14, 1.65419E-15, 1.07489E-16 and 4.26843E-17, respectively. We selected a two-tailed T-test since we need to test if our system results in significantly worse performance also.

Most of the p-values validate that our approach is meaningfully better than the benchmark, above the 95% confidence level. However, there are a few points to consider. With initial funds of £1000 our T-test returned a p-value of 0.000002, meaning there is a significant difference between our model and the buy & hold with above 95% confidence. However, the mean Sharpe ratio of our model compared to the buy & hold for this level of initial funds was actually lower, at -1.07, than the buy & hold (-0.16). The variance of our model (3.86) was also higher than the buy & hold (0.39). These results mean that our model did perform differently than the benchmark at above 95% but for this case only the performance was worse. This observation leads us to conclude that our model does not work with £1000 initial funds.

Another point of interest is when initial funds of £2000 were compared with the benchmark. When increasing initial funds in our model to £2000, we obtained a p-value of 0.18991, from which we conclude that at £2000 our model performed better than buy & hold but not at a significant level (taking the standard 0.05 to be the significance level). There is too large a chance the results may have been due to random variation. Therefore, if the investors would like to be confident about improving their performance with our model, they should invest with a minimum amount of £3000. At all initial fund values £3000 and above, our model performs better than the buy & hold at above the 95% confidence level. The p-value at£3000 declines to 0.000036, and the average Sharpe Ratio (0.09) of our model is also better than the buy & hold (-0.16) at this amount of funds. We also confirm that increasing the initial funds from this point to £10000, our model will provide better results than the buy & hold with more than 95% confidence that these results are due to our system.

6.3.3 Result Comparison with benchmarks

In this section, the performance of our model will be compared with the benchmarks. We have selected two benchmarks in this section: the buy & hold strategy and the market index. Firstly, we compare our model's with the performance of the buy & hold strategy for each stock in order to confirm that using our model is better than buying at the beginning and selling at the end of the test period. Secondly, we widen our comparison to the market index, which is the standard benchmark for the fund and unit trust industry and would allow retail investors to compare our model's performance with popular alternatives available to them. The rationale behind this benchmark is that if a model cannot beat the index, it is easier and cheaper for investors to invest in index tracker funds; highly liquid, often automated instruments which seek only to match the return of an index. Comparing the performance of our model to the market index is one of the best ways to show an investor how well our model works.

6.3.3.1 Buy & Hold strategy and Market Index

In this section, we first calculated the performance of the buy & hold strategy for all 100 stocks in the FTSE 100 index. The complete list of results can be seen in Appendix A Section A.2 on page 302. Secondly, we showed the performance of the FTSE 100 index during the period of testing. These two performances were then used in order to summarise our model's efficiency.

In this experiment, we selected all of the stocks from the London Stock Exchange which are listed on the FTSE 100 index. The performance of this index during the testing period (2017-2018) will be used as the benchmark. The daily closing prices of the FTSE 100 from the beginning of 2017 to the end of 2018 can be seen in Figure 6-20 and the corresponding details of this benchmark can be seen in table 6.6.

Benchmark	Sharpe Ratio	Profit (%)	Annual Risk	Max DD	Avg DD
FTSE100 index	-0.3	-6.27	0.109	-16.34	-3.76

Table 6.6: FTSE100 performance during the testing period of 2017-2018

From Figure 6-20 it can be seen that the FTSE 100 is very fluctuated for the whole of 2017. There is a huge drop during the first quarter of 2018, however, the market recovered very quickly in the second quarter. It even made a new high price during this period. Unfortunately, before the end of the testing period, the price started to drop sharply again. The detail of the FTSE 100 index's performance in this testing period is shown in table 6.6. As expected, the FTSE 100 does not perform well and therefore ends up with a negative Sharpe Ratio of -0.3 and maximum drawdown of -16.34%.

The Sharpe Ratio of our model is compared with the buy & hold strategy and the FTSE 100 index. The summary of this comparison can be seen in Figure 6-21.

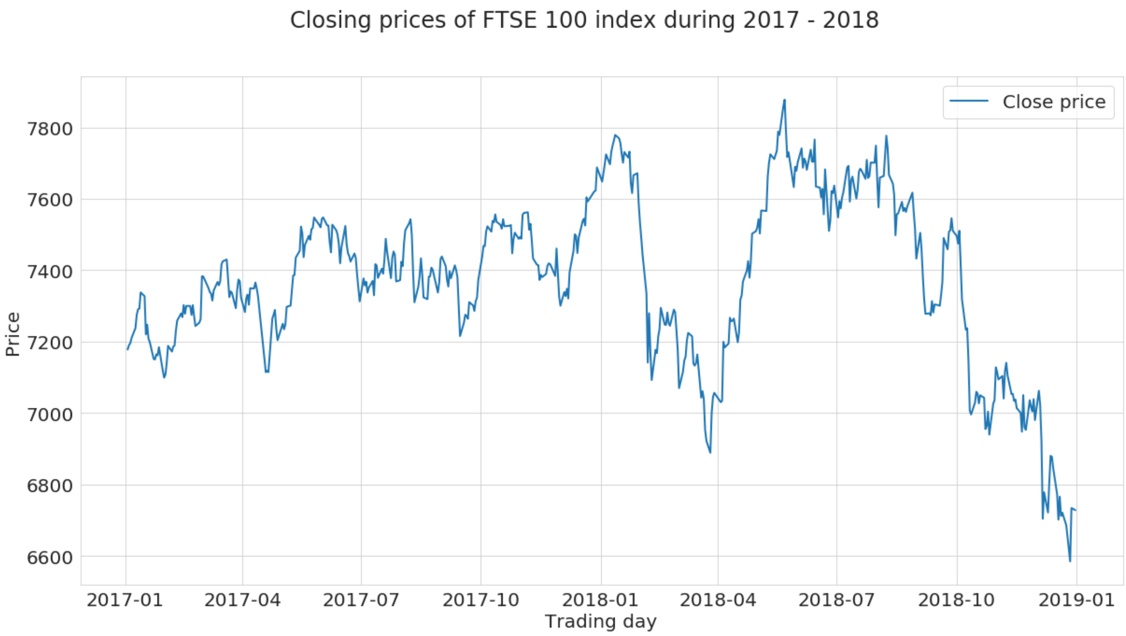


Figure 6-20: Closing price of FTSE 100 during 2017-2018

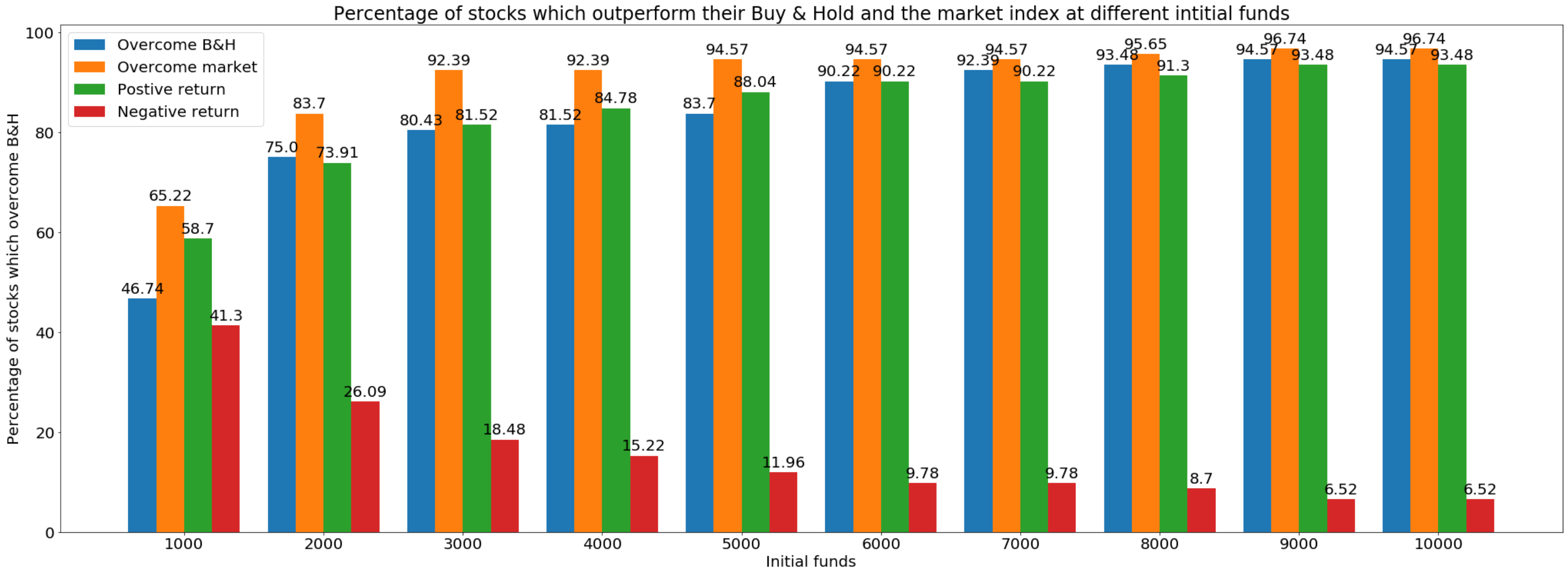


Figure 6-21: Summary of our model performance

In Figure 6-21, the x-axis shows the amount of initial funds, ranging from £1,000 to £10,000. The y-axis represents the percentage of stocks within the FTSE 100 index that meet the following 4 conditions. Firstly, the stocks which overcame their buy & hold performance (blue bars). Secondly, stocks that overcame the market index (orange bars). Thirdly, stocks that ended up with a profit after investing with our model (green bars), and lastly the stocks which ended up with a loss after investing with our model (red bars). There will be four of these values corresponding to each amount of starting capital.

Overall, the potential to overcome the buy & hold (blue) and the market index (orange) increases with more initial funds, which can be seen from increasing the size of the blue and orange bars. The possibility of gaining profit is also increased with the amount of funds, as the green bars also increase when initial funds are higher. On the other hand, the sizes of the red bars, which indicate the possibility of losing money, decrease with more funds.

The biggest gaps in the possibility of winning over the buy & hold strategy (blue bar) are between initial funds of £1,000 and £2,000. With only £1,000 to invest, there is about a 46.7% chance to overcome the buy & hold strategy. However, by increasing the initial funds to £2,000, investors have more than a 75% to gain more profit using our model than following the buy & hold strategy.

Similarly to the buy & hold strategy (blue bars), an investor has a 65.22 % probability of beating the market index with initial funds of £1,000. However, this possibility increases to over 83% if the initial funds increase to £2,000.

We have mentioned the probabilities of winning over the benchmarks above, and now we will analyse the likelihood of gaining or losing money (green and red bars). As can be seen from the sizes of the green bars, which indicate the chance of making a profit, the chance of investors ending up with a profit using our model to invest in FTSE 100 constituents is 58.7% with starting capital of £1000. This will increase to more than 80% if more than £3,000 is available to invest, and more than 90% if investing more than £6000. Finally, with initial funds of £10,000, an investor would have had a 93.48% to make a profit on stocks listed in FTSE 100 during 2017-2018.

As the likelihood of a profit increases with initial funds, the possibility of losing money decreases when greater initial funds as can be noticed from the decreasing sizes of the red bars as initial funds increase. If investors had only £1,000 to invest in FTSE 100 stocks during 2017-2018, they would have had to accept that a 40% chance of losing money. However, this drops by almost half (26.09%) when initial funds increase to

£2,000. Moreover, starting from £5,000 onward decreases the risk of losing money to below 10%. With £10,000, an investor had only about a 6% chance of losing when investing in stocks listed on the FTSE 100 during 2017-2018 using our model.

6.4 Comparison with other systems

After considering the results of our model in comparison with the benchmarks in the previous sections, this section compares our model with other recent research with the same objective. In order to compare our model with other works, we ran our model on the same datasets over the same period of time as the system with which we are comparing. The most important result we consider is the Sharpe Ratio, followed by other values such as profit, volatility and drawdown.

Our experimental result in table 5.3 on page 187 indicates that accuracy alone is not enough to evaluate financial models. The full discussion on this matter can be seen right after this table in section 5.4.1 on page 192. The logic behind this is that one can still be profitable even if their predictions are wrong most of the time provided that their losing trades are much smaller than their profitable trades. For example, the result in table 5.3 on page 187 shows that when using SVR (linear kernel) with FERG stock, the accuracy was only 49.5 %. This means that more than half of the time the predictor provided the incorrect answer. However, the result still ended up with a profit of 37.62 %. This was because the model was able to pick highly profitable trades. There were about 249 days profitable days and 253 losing days. Even though the number of profitable days was lower than the losing days, the average return on winning days was higher than the losing days. The average return on the winning days was about 1.09 %, while the average return on the unprofitable days was only 0.92 %. This might not seem a big difference but once daily profits accumulate and related fees are taken in to account, the profit is increased from 18.59 % to 37.62 %, with the former being the benchmark return. More details are provided in section 5.4.1 on page 192

We found the idea that one need not be accurate to be profitable not just in our results, but reported by experienced professional traders in [95]. The superiority of asymmetric risk-reward over accuracy has been widely recognised by professional traders outside of academia. Non-academic trading literature is still largely dominated by individual, discretionary traders, as opposed to automated trading systems, so the principle of asymmetric opportunities where the potential profit is a multiple of the possible loss had to be manually enforced by the traders exiting losing positions. Hence, they often talk about "cutting" or "taking" losses, but the underlying principle to which they are referring is the same as what automated systems should be trying to achieve. It is shown that successful traders understand intuitively how they can be inaccurate yet still make a profit over the long run if they have the discipline to limit their losses to keep them much smaller than their profitable trades. It is ubiquitous across trading literature that this is perhaps the most important principle of successful trading.

Ed Seykota, an early pioneer of trading systems, summarized his approach in an interview for Jack Schwager's popular Market Wizards book: "The elements of good trading are: (1) cutting losses, (2) cutting losses, and (3) cutting losses." This sentiment is expressed in some form or other by almost all the interviewed traders, including rules-based and systematic traders like Richard Dennis ("You have to minimize your losses and try to preserve capital for those few instances where you can make a lot") and Michael Marcus ("Perhaps the most important rule is to hold on to your winners and cut your losers. Both are equally important. If you dont stay with your winners, you are not going to be able to pay for the losers.") as well as discretionary traders such as Paul Tudor Jones ("If I have positions going against me, I get right out; if they are going for me, I keep them. Risk control is the most important thing in trading.") and Bruce Kovner ("You have to be willing to make mistakes regularly; there is nothing wrong with it. Michael taught me about making your best judgement, being wrong, making your next best judgement,being wrong, making your third best judgement, and then doubling your money.")

6.4.1 Comparison with Birbeck and Cliff's system

As mentioned in chapter 3 section 3.1.10 on page 69, Birbeck and Cliff [17] introduced a new sentiment analysis labelling approach in order to generate profits in the stock market using Twitter posts. Instead of labelling the posts by the true sentiment as in previous research, for example if it is expressing a positive or negative opinion, they created a new method by reverse labelling each post based on the ground-truths, that whether the relevant stock price moved up or down during the period in which the posts were collated. In this research, Support Vector Machine, Naive Bayes, and Logistic Regression were applied in order to make predictions. The new system was tested on four well-known stocks from the technology sector: Apple (AAPL), Tesla (TSLA), Twitter (TWTR) and Facebook (FB).

We compare the performance of our model with Birbeck and Cliff's system by using our model on the same dataset and time frame (January 2017). Before having a look at the comparison, a few differences between our system and Birbeck and Cliff's should be understood. Firstly, in Birbeck and Cliff's research, the datasets come in the minute format, while our model uses the daily format. Secondly, as Birbeck and Cliff's model uses the minute data, it aims to trade much more often than our model. The number of trades can be up to 120 times per month or 6 times per day. Meanwhile our model aims to be used by the individual investor for whom we assume a restrictive amount of starting capital and potentially time to spend trading, so the number of trades and the associated transaction fees must be kept within reason otherwise they will significantly impact profitability or the model will become unrealistic for the retail investor.

We ran our model on AAPL, TSLA, TWTR and FB and compared the Sharpe Ratio and Profit with Birbeck and Cliff's model over the same period (January 2017). Before comparing these results, we would like to show the result of our model for the whole of 2017. The detailed results from testing our models on AAPL, TSLA, TWTR and FB are shown in Figures 6-22, 6-23, 6-24 and 6-25, respectively. All of these results are calculated based on initial funds of £10,000.

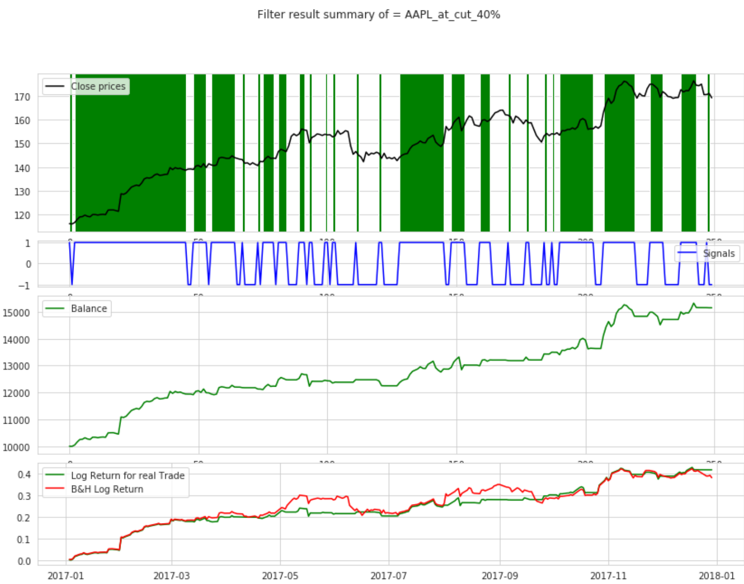


Figure 6-22: Result from AAPL in 2017

Figure 6-22 shows the final result from AAPL for the whole of 2017. Our model made 51 trades and provided a Sharpe Ratio of 3.12 and profit of 51.55 %. The maximum and average drawdown were -4.88% and -1.12% respectively. As can be seen from the top graph, only two trades happened in January which are shown in green. As the latter was held for the majority of the month, the profit and Sharpe ratio of this month were not very different from the buy & hold strategy.

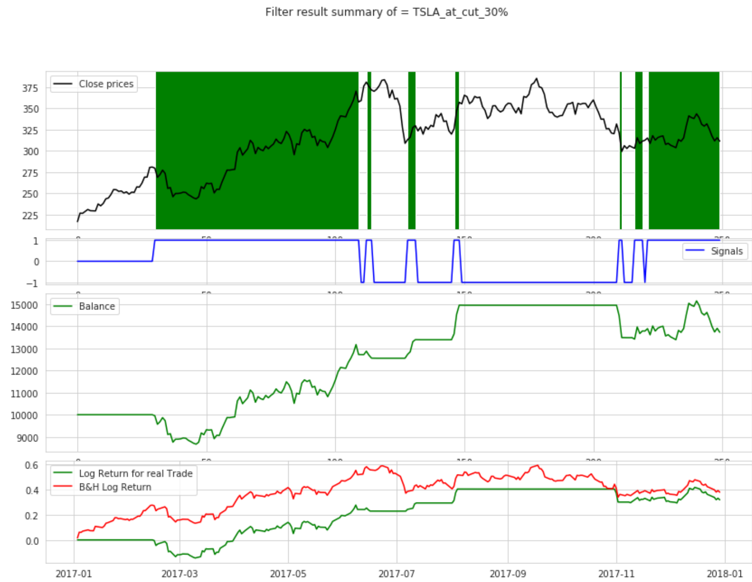


Figure 6-23: Result from TSLA in 2017

Figure 6-23 shows the result from TSLA. There are 13 trades for the whole of 2017. Our model ended with a Sharpe Ratio and profit of 1.2 and 37.18% respectively. The maximum and average drawdown were -13.3% and -2.8%. It can be seen from the top graph that our model started the first buy order just before the beginning of March and held the first trade until after June. After that there are six buy orders which were sold shortly after buying, until the model opened the long position again just after November and held this position until the end of the year.

Figure 6-24 shows result from TWTR and it can be seen that our model did not trade much in the first half of the year. The total number of trades is 11 throughout the year. The Sharpe Ratio and profit are 1.48 and 64.79%. The maximum and average drawdown are -23.39% and -6.22 %, respectively.

Figure 6-25 shows the result from FB, which is very close to the buy & hold as most of the trades were held for a long time. The total profit and Sharpe Ratio are 42.1% and 2.15 for 2017. The maximum and average drawdown are -6.32% and -1.28%, respectively. The total number of trades is 20.

All of the results mentioned earlier were tested over the whole of 2017. However, we want to compare our result with Birbeck and Cliff's system, which only shows the

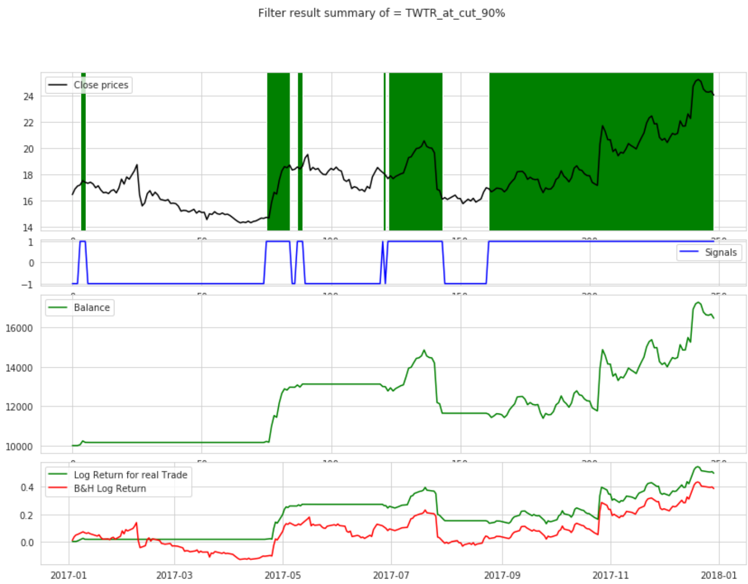


Figure 6-24: Result from TWTR in 2017

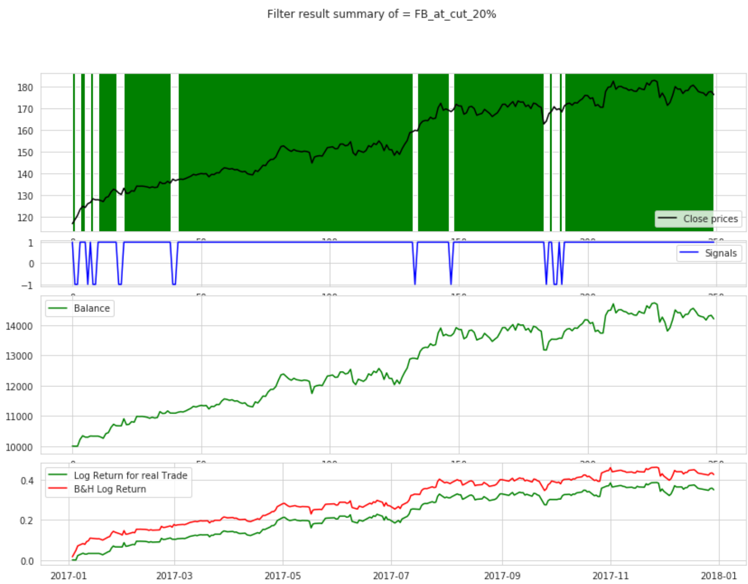


Figure 6-25: Result from FB in 2017

results for January. Therefore, we will only calculate our model's performance for January of 2017 in the comparison. The comparison of our model and Birbeck and Cliff's model for all four stocks is shown in table 6.7.

Stock	AAPL	TSLA	TWTR	FB
Profit ([17])	4.48	8.27	0.04	-4.08
Profit (Our model)	**4.51**	**0**	**1.56**	**6.69**
Sharpe Ratio ([17])	2.78	3.06	-0.016	-5.46
Sharpe Ratio (Our model)	**6.94**	**0**	**2.67**	**7.01**
Orders ([17])	120	120	120	108
Orders (Our model)	**2**	**0**	**2**	**7**

Table 6.7: Profit comparison between our model and the sentiment analysis model on January 2017

Table 6.7 shows the comparison between our model and Birbeck and Cliff's model for AAPL, TSLA, TWTR and FB during January 2017. The first difference between our model and Birbeck and Cliff's model is the number of trades. Our model only traded two times for AAPL and TWTR in that month, while Birbeck's system traded 120 times on almost every stock, except 108 times from FB. As mentioned earlier, this is a major difference between the two models as a result of their different objectives.

Both models provide a very good result for AAPL. The profits are very similar at 4.51 and 4.48. However, as our model only traded twice and both of the trades were profitable, our Sharpe Ratio (6.94) was higher than Birbeck and Cliff's model (2.78). As for TSLA, while Birbeck and cliff's model provided a very good result at 8.27% profit and 3.06 Sharpe Ratio, our model did profit at all as there were no order signals provided that month.

TWTR and FB did not perform very well with Birbeck and Cliff's model; our model provided better results on these two stocks. We only had one order for TWTR which ended up with a little profit of 1.56% profit, while Birbeck and Cliff's model resulted in 0.04%. As for the Sharpe Ratio, as our model only had one order and that order was correct, our Sharpe ratio was 2.67, while Birbeck and Cliff's model ended up with -0.016. As for FB, our model traded seven times in that month and ended up with a very good Sharpe Ratio of 7.01 and 6.69% profit.

From the results, it can be seen that our model works well on most of the datasets. It provides fewer trade instructions which is important for the individual investor. Most of the trades it made were correct, making the Sharpe Ratio for the period quite high. However, it is better to see the performance over a longer period of time as our model aims to not trade often and therefore, might not provide any order instructions over short periods of time, as shown in the result for TSLA. Please note that as this comparison period is only 1 month, we did not include the risk-free rate into the calculation in this table .

6.4.2 Comparison with Qin's system

We compared our model with that of Qin [89] which was mentioned in chapter 3 section 3.2.6 using the same stocks. Qin uses 9 stocks and 1 Index from the Singapore stock exchange, but we could only obtain data for 8 stocks from our source, Yahoo Finance (`https://finance.yahoo.com/` retrieved 2019-09-15); the STI index (Straits Times Index) and SMRT (S53) were not available at the time. The stocks tested were: Capitalland (C31), DBS (D05), UOB (U11), SGX-Singapore Exchange (S68), Starhub (CC3), Singtel (Z74), Semb Corp (U96), and and SIA-Singapore Airline (C6L). Qin's datasets cover a 5-year period from 2005-09-01 to 2010-08-31, with the first 3 years for training and the last 2 for testing. Therefore, results in this comparison are for 2009 and 2010 (417 trading days, ending 2010-08-31).

Before having a look at the comparison, we present the results from our system on the individual stocks in table 6.8. Then, we will show the comparison between our model and Qin's (Gradient Boosted Random Forest) along with buy-and-hold, Score, Least Square and Random Forest, which are also used in Qin's work in table 6.9. Please note that Qin did not show the result for each dataset separately, therefore we are only able to compare these results in average values.

Stock	**Sharpe Ratio**	**Profit**	**Volatility**	**Max DD**	**Avg DD**	**Trades**
C6L	1.85	94.54	0.219	-12.28	-2.77	164
C31	1.26	109.76	0.359	-14.76	-6.7	68
CC3	2.32	59.77	0.143	-7.56	-1.01	98
D05	1.62	80.51	0.222	-12.67	-2.25	14
S68	1.53	71.83	0.209	-15.23	-2.82	122
U11	1.39	71.19	0.234	-20.14	-4.37	124
U96	0.91	72.34	0.383	-22.49	-6.12	18
Z74	1.86	67.99	0.176	-10.05	-2.06	162
Average	**1.60**	**78.50**	**0.243**	**-14.40**	**-3.51**	**96.25**

Table 6.8: Result from our model on Qin datasets for the testing period of 417 days during 2009-2010

Table 6.9: Result Comparison with Related Work [89]

Algorithm	**Yearly Profit (%)**	**Correction Prediction (%)**	**Sharpe Ratio**	**Drawdown (Max)**
Buy & Hold	-1.94	-	0.01	-47.34
Score	21.71	19.12	0.02	-10.20
Least Square	16.77	21.97	0.03	-6.66
Random Forest	24.32	26.38	0.02	-12.05
Qin's model	25.14	30.25	0.03	-10.06
Our model	**33.6**	**57.40**	**1.60**	**-14.40**

Table 6.8 shows that our model works well with all of the datasets. Most of the Sharpe Ratios are over 1.5, which is very good. The number of trades our model made varies from 14 to 164 times during the testing period of 417 days. The average of our model's performance can be seen in the last line of this table. The average Sharpe Ratio is very high at 1.60 with over 78% profit. These average results are compared with Qin's model in table 6.9. The additional detail result from these stocks can be seen in Appendix B section B.1 on page 325.

Table 6.9 shows the average results for yearly profit, correct prediction, Sharpe Ratio and maximum drawdown during the testing period of 417 days. It can be seen that the buy & hold profit of these stocks is negative, which means that if investors had invested in all of these stocks at the beginning of the test period and held until the end, they would have ended up with about a 1.94% loss. Other algorithms, score, least square, random forest and Qin's model, preformed much better than the buy & hold with 14.71%, 16.77%, 24.32% and 25.14% returns respectively. However, our model provided a much better result than the other models mentioned, with a profit of 33.60%.

As for the most important comparison metric, Sharpe Ratio, our model gave an extraordinary result with a high Sharpe Ratio of 1.60, while the other models only ended up with below 0.1. Therefore, it can be seen clearly that after considering the profit and risk, our model performed much better than Qin's model and the others. Even though the drawdown of our model is about 4% bigger than Qin's model, our model still ended up with a high Sharpe Ratio, which includes drawdown in its consideration of risk. The percentage correct predictions is not one of our performance metrics, therefore we will not compare this value.

6.4.3 Comparison with Li's system

This paper [68] presents a system that uses machine learning models to invest in momentum and reversal strategies in the stock market. In their experiment many models of machine learning - Decision Tree (DT), Support Vector Machine (SVM), Multilayer Perceptron Neural Network (MLP) and Long Short-Term Memory Neural Network (LSTM) - are investigated. The experimental results on an index from China, CSI300, show that SVM is the best machine learning approach to capture momentum and reversal situations, leading to profitable trading. In this research, each strategy was run 20 times with different parameters. In this section, we are going to show only the results that related to minimum, average and maximum Sharpe. Results from Li's system on every machine learning model tested are shown in 6.10.

Model	**Sharpe Ratio**			**Profit (%)**		
	Min	**Avg**	**Max**	**Min**	**Avg**	**Max**
Momentum	-0.15	0.26	0.79	-33.78	21.73	116.55
Reversal	0.14	0.48	0.85	-0.87	51.95	124.32
DT	0.05	0.55	1.52	-9.00	55.73	175.90
SVM	-0.60	0.67	1.68	-37.43	77.48	239.43
MLP	-0.22	0.56	1.41	-24.52	61.43	215.26
LSTM	-0.31	0.41	1.25	-32.40	45.89	201.30

Table 6.10: Results from Li's work [68] for each strategy at the minimum, average and maximum Sharpe Ratio

In order to compare the results of Li's model with our results, we decide to compare with their average results. The reason is as Li's work run experiment multiple times with different hyperparameters and the best results from each strategy are obtained from different choices of paremeters. Therefore, this work cannot identify which parameters should be related to the best results. Therefore, instead of deliberately pick best results, which is really unfair to us, or pick the worst result, which is unfair to them, we decide to compare with the average values. The comparison between our results and Li's results can be seen in table 6.11.

Table 6.11 shows the result from CSI300 during the testing period from 2012-01-04 to 2016-02-05. There are two main values, Sharpe Ratio and profit, selected for this evaluation. It can be seen clearly that our model provides the best Sharpe Ratio (0.83)

Model	Sharpe Ratio	Profit (%)
Momentum	0.26,	21.73
Reversal	0.48	51.95
DT	0.55	55.73
SVM	0.67	77.48
MLP	0.56	61.43
LSTM	0.41	45.89
Our model	**0.83**	**89.01**

Table 6.11: Profit comparison between our model (after taking 3% risk-free rate into account) and Li's work [68]

and profit (89.01). The accuracy of our model is 43.43% The detail of this result is shown in Figure 6-26.

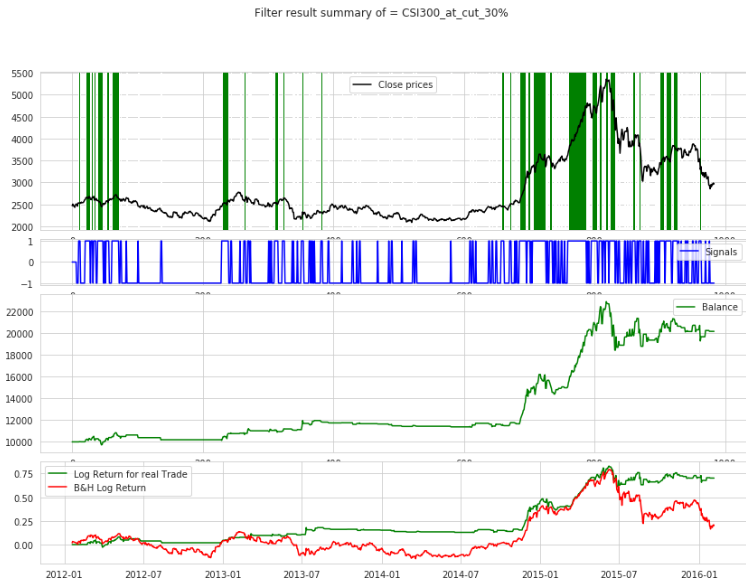


Figure 6-26: Result from our model testing on CSI300 during 2012 - 2016

The top graph of Figure 6-26 shows that our model starts sending orders at the very beginning of the testing period but stops for almost a year. Then, at the beginning of 2013, it starts to trade again but not very often. Long-term trades happen at the end of 2014 when the stock price increases. The total number of trades is 199. The middle graph shows the balance throughout the period of testing. It can be seen that our model starts to make a profit when the price increases before the end of 2014 (the closing prices of this stock is shown in the top graph in black). The comparison between our model and the buy & hold strategy can be seen in the bottom graph. Our model's return (green) is better than the buy & hold, especially when the price decreases at the end of the testing period. Our model's profit does not go down as sharply as the buy & hold return.

6.5 Discussion

This chapter illustrates the performance of our model by simulating the trading system. We have made the simulation as close to the real world as possible by taking the brokerage fee into account and the process of buying and selling is made realistic by taking the 3% risk free rate into account. As for the evaluation of our model, we have run experiments on a variety of stocks. Firstly, our model was tested on a group of 11 stocks with different characteristics to make sure that our model works well under different market conditions. Secondly, we widened the test set to the UK market. We selected the 100 stocks which make up the FTSE 100 index. These stocks come from different business sectors which make them have different characteristics. All of the results are promising. Our model shows good performance regardless of market sector.

After finishing the experiments on a wide rage of stocks, we compared our model with other recently developed systems by taking the same datasets and test periods to run with our model. This gave our model an opportunity to run on even more stocks from new markets such as the USA, China and Hong Kong. This is important not just from a geographical perspective, but also because China is widely considered an "emerging market", as opposed to "developed market" such as the US. This can effect market conditions, as higher growth but greater volatility and risk are typically expected from emerging markets. The comparison shows that our model works well with stocks from other markets in terms of Sharpe Ratio, profit, volatility and drawdown, and indicates that it can also perform well over shorter time-frames.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Predicting the future direction of stocks in order to trade profitably is extremely challenging as there are a lot of factors that affect the stock prices, and it is not possible to identify them all. Fluctuations in the data (price) also make it difficult to predict future values. However, technical investors believe that all fundamental factors are already reflected in the price movements, but analysing historical price patterns should be able to provide orders (buy or sell) based on investors' behaviour which will lead to profit.

Many machine learning models have been applied to the stock market in attempts to achieve profit and avoid risk. However, they still suffer from a lack of adaptability and only work well with specific datasets or periods of time, rather than over the whole range of market scenarios. In order to solve this problem, we created a new adaptive trading model that has been proven to work well for a wide range of data representing different business sectors and market conditions.

Our model is composed of many machine learning predictors which have been trained on specific pieces of the dataset to develop specialities in prediction across different situations. To make these machine learning predictors work together, we have created an effective scoring system which is capable of selecting suitable predictor(s) for specific periods of time. Only the selected predictor(s) will be allowed to make decisions. As the market situation and stock price characteristics are changing over time, the model needs to be able to re-select predictors in order to respond to the fluctuations. Therefore, in our scoring system, we have created different features in order to track changes in each predictor's performance. Predictors whose performance concerning these features has been strong will be selected. This process continues and scores get updated every day. Finally, at the end of the day, a new set of predictors will be activated to produce a forecast. However, not every forecast results in an instruction to buy or sell shares. The model will consider other criteria before giving instructions, for example how much capital the investor has to start with and whether the trade would be worth making in consideration of the risk and related fees.

The experimental results show that our model works effectively with a wide range of stocks from different market sectors and different markets globally. In the experiment and comparison section, we showed that our model works well on more than 100 stocks from the UK, US, Chinese and Singaporean markets. These stocks come from more than 10 different market sectors. We believe that this covers a wide-range of market conditions. Therefore, we confirm that our model shows an excellent capability of handling predictions in fluctuated situations and can be used regardless of the characteristics of the stock data.

In summary, this research proposed an ensemble system of many well-known machine learning predictors on the basis that machine learning models have already been identified that work well in certain areas or with specific types of data, but struggle to adapt when new patterns emerge. Therefore, instead of creating a novel model, we focused on combining the intelligence of existing models to produce a system that could profit under varying conditions.

7.2 Future work

While our model has shown excellent performance based on our experiments across the broad datasets, it can be readily adapted for future research and the following are some areas in which such research may focus. Firstly, the number of predictors is flexible. It is easy to increase or decrease the predictors. This model created an effective way of putting many predictors to work together automatically, however it did not vary the number of predictors or which predictors were included in the experiments. Therefore, if one wants to change the type of predictor or add or delete some of them, this is easy to perform. It is possible that performance may be improved by the addition of new predictors that can handle new stock price patterns well, however as the processing time of this system depends on the complexity of the predictor(s), it could be possible that some which might only be useful in very niche conditions are not worth keeping.

Secondly, in the case of using this model over a long period, then re-training can be considered. For example, the training process can be activated every month, for example, or at any other interval, instead of only training at the beginning and running indefinitely. Thirdly, in this model, the number of trades is likely to increase with more funds as the system allows weaker forecasts with less potential profit to be traded that would otherwise have been filtered out if starting capital was smaller. However, this is not always helpful, as for some datasets the number of trades should not have been increased even when starting with more funds, as is shown in the result of CARR in Figure 6-10 on page 245. Therefore, the process to optimise the number of trades in this model could be improved. Thirdly, even though this system was designed for individual traders, the idea of the scoring system can be adapted for institutional traders. Details and some suggestions of how it can be adapted can be seen in appendix C on page 328, where we also show results on some of our stocks after adaptations such as the inclusion of short-selling and percentage-based trading fees.

Finally, it is recommended to use this model together with other risk control methods, for example a stop loss order which automatically sells all shares in a trade in the event of a predetermined loss so as to prevent further loss of capital. This can help prevent a large drawdown, which is more likely to happen when the starting capital is low, as can be seen in the results of starting with £1,000 in Figure 6-16 on page 260. Stop losses also help protect against unforeseen events that can be of significant negative consequence to a company. These do not show up in historical volatility or past price information and cannot be predicted by models, which is why we emphasise that our model is intended for use only in conjunction with additional risk control methods. Further to this, it is not recommended to place all available capital into each trade, as was simulated here. A common recommendation is to invest a predetermined percentage of available capital into each trade, known as position sizing, such that if the worst happens and the share price goes to zero, one still has money for other trades to recoup the loss.

Another form of risk control is investing in the shares of large companies, such as constituents of the FTSE 100 index, which are said to be highly liquid. This means there are many buyers and sellers, therefore orders are more likely to be executed in the specified size and closer to the price at that moment. This is an important consideration both to minimise slippage - the difference between the price used to generate the model's instruction and that at which the shares are actually purchased - and in order to exit trades quickly when necessary to preserve capital. Slippage also affects the hypothetical performance of all trading models.

Something this model did not incorporate is diversification. Diversification in its simplest terms is closely related to position sizing in that it attempts to prevent a catastrophic loss from which one can't recover, and to do that it aims to reduce the correlation between several shares or other financial assets traded simultaneously. One way to do this is to spread shares in a portfolio across different countries. As it has been shown that our model works well for several international markets, this could be a promising area for future improvement as successfully incorporating geographical diversification could offer the profit potential of our model with even less risk.

Fundamentally, this model was created in order to handle fluctuations in time-series data, and therefore it is not necessarily limited to only the stock market. There are a lot of time-series problems in other areas that face problems of fluctuation, which makes them extremely difficult to predict. Examples could be disaster prediction or gaming, as the behaviours or situations are difficult to predict and there are a lot of factors having an impact and these factors can change rapidly over time. This idea could be used in order to handle these types of problems.

Bibliography

[1] C. A., Z. A., and Z. B. *Investment*. Homewood Illinois: 5th edition, 1993.

[2] M. H. L. Abdullah and V. Ganapathy. Neural network ensembles for financial trend prediction. *TENCON*, 3:157 – 161, September 2000.

[3] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo. Stock price prediction using the arima model. *The 16th International Conference on Coputer Modelling and Simulation (2014 UKSim-AMSS)*, pages 106 – 112, March 2014.

[4] B. Al-Hnaity and M. Abbod. A noval hybrid ensemble model to predict ftse100 index by combining nueral network and eemd. *Eurapean Control Conference*, pages 3021 – 3028, July 2015.

[5] J. Andrada-Felix, F. Fernandez-Rodriguez, M. D. Garcia-Artiles, and S. Sovilla-Rivero. An empirical evaluation of non-linear trading rules. *FEDEA and University Complutense de Madrid*, 2001.

[6] C. M. Anish and B. Majhi. An ensemble model for net asset value. *Communication and Information Technology Conference (PCITC)*, pages 392 – 396, 2015.

[7] M. Asad. Optimized stcok market prediction using ensemble learnging. *The 9th International Conference on Application of Information and Communication Technologies (AICT)*, pages 14 – 16, October 2015.

[8] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research 3*, 2002.

[9] S. A. J. Babulo, B. Janaki, and C. Jeeva. Stock market indices prediction with various neural network models. *International Journal of Computer Science and Mobile Application*, 2:42–46, March 2014.

[10] T. Balch. Machine learning for trading. Technical report, Georgia Technology, https://www.udacity.com/, 2016.

[11] H. Bandy. *Quantitative Technical Analysis.* Blue Owl Press, 2015.

[12] A. Belousov, S. Verzakov, and J. von Frese. A flexible classification approach with optimal generalisation performance: support vector machines. *Chemometrics and Intelligent Laboratory Systems*, 64(1):15 – 25, 2002.

[13] L. Benali, G. Notton, A. Fouilloy, C. Voyant, and R. Dizene. Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components. *Renewable Energy*, 132:871 – 884, 2019.

[14] J. Berg, R. Forsythe, F. Nelson, and T. Rietz. *Results from a Dozen Years of Election Futures Markets Research.*, pages 742 – 751. Handbook of Experimental Economics Result, 2008.

[15] J. Berg and T. A. Rietz. Prediction markets as decision support systems. *Information System Frontiers*, 5:79 – 93, 2003.

[16] D. Bhuriya, G. Kaushal, A. Sharma, and U. Singh. Stock market predication using a linear regression. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, volume 2, pages 510–513, 2017.

[17] E. Birbeck and D. Cliff. Using stock prices as ground truth in sentiment analysis to generate profitable trading signals. *IEEE Symposium Series on Computational Intelligence (SSCI 2018)*, pages 1868 – 1875, 2018.

[18] Bo Zhang and Chen Wang. Automatic generation of test data for path testing by adaptive genetic simulated annealing algorithm. In *2011 IEEE International Conference on Computer Science and Automation Engineering*, volume 2, pages 38–42, 2011.

[19] J. Bollinger. Using bollinger bands. *Stock and Commodities*, 10:2:47–51.

[20] D. Bray, K. Croxson, and W. H. Dutton. Information markets: Feasibility and performance. *papers.ssrn.com*, 2008.

[21] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140.

[22] L. Breiman. Pasting small votes for classification in large databases and on-line. *Machine Learning*, 36:85–103, 2004.

[23] L. Brock, J. Lakonishok, and B. Lebaron. Simple technical trading rules and the stochastic properties of stock return. *Journal of finance*, 1992.

[24] C. Caginalp and G. Caginalp. Stochastic asset price dynamics and volatility using a symmetric supply and demand price equation. *Physica A: Statistical Mechanics and its Applications*, 523:807 – 824, 2019.

[25] C. Campbell. Statistical pattern recognition. Technical report, University of Bristol, 2014.

[26] J. Y. Campbell and T. Vuolteenaho. Bad beta, good beta. *American Economic Review*, 94(5):1249–1275, December 2004.

[27] J.-W. Chen, C.-H. Chen, and S.-C. Chen. Application of fuzzy k-mean cluster and fuzz similarity in soil classification. 2005.

[28] C.-H. Cheng and H.-Y. Shiu. A noval ga-svr time series model based on selected indicators method for forecasting stock price. *International Conference on Information Science, Electronics and Electrical Engineering (ISEEE)*, pages 395–399, April 2014.

[29] R. W. Colby. *The Encyclopedia of Technicial Market Indicators*. McGraw-Hill, 2003.

[30] D. Davis, Y. Luo, and K. Liu. A multi-agent framework for stock trading. *Conference on Intelligent Information Processing*, 2000.

[31] T. G. Dietterich. *Ensemble Method in Machine Learning*, volume 2. Springer Berlin Heidelberg, 2000.

[32] K. Drachal and M. Pawowski. A review of the applications of genetic algorithms to forecasting prices of commodities. *Economies*, 9(1), 2021.

[33] G. Dudek. Short-term load forecasting using random forest. *Advances in Intelligent Systems and Computing*, 323:821–835, 2015.

[34] S. Dzeroski and B. Zenko. Is combining classifiers with stacking better than the best one. *Machine Learning*, pages 255 – 273, 2004.

[35] L. S. Exchange. London stock exchange.

[36] E. F. Fama. The behavior of stock-market prices. *The Journal of Business*, 38(1):34–105, January 1965.

[37] E. F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417, May 1970.

[38] Y. Finance. Yahoo finance.

[39] A. Geron. *Hands-On Machine Learning with Scikit-Learn and Tensorflow*, volume 1. O' Reilly Media Inc, 2017.

[40] S. T. R. Gilani, M. Nawaz, M. I. Shakoor, and M. Z. Asab. Testing the weak form efficiency of islamabad stock exchange (ise). *Developing Country Studies*, 4:79–86, 04 2014.

[41] R. T. Gonzalez, C. A. Padilha, and D. A. C. Brrone. Ensemble system basedon genetic algorithm for stock market forecasting. *IEEE Congress on Evolutionary Computation (CEC)*, pages 3102 – 3108, May 2015.

[42] Y. Guo, S. Han, Y. Li, C. Zhang, and Y. Bai. K-nearest neighbor combined with guided filter for hyperspectral image classification. *Procedia Computer Science*, 129:159 – 165, 2018. 2017 INTERNATIONAL CONFERENCE ON IDENTIFICATION,INFORMATION AND KNOWLEDGEIN THE INTERNET OF THINGS.

[43] A. Gupta and B. Dhingra. Stock market prediction using hidden markov models. *Engineering and System (SCES)*, 2012.

[44] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transection of Pattern Analysis and Machine Intelligence*, 12:993 – 1001, August 1990.

[45] M. R. Hassan. A combination of hidden markov model and fuzzy model for stock a combination of hidden markov model and fuzzy model for stock market forecasting. *Computational and Ambient Intelligence*, 2009.

[46] M. R. Hassan, B. Nath, and M. Kirley. A fusion model of hmm, ann and ga for stock market forecasting. *Expert Systems with Applications*, 33(1):171 – 180, 2007.

[47] D. M. Hawkins. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12, 2004. PMID: 14741005.

[48] T. K. Ho. Random decision forest. *Proceeding of 3rd International Conference on Document Analysis and Recognition*, 1:278–282.

[49] L. Hong and S. E. Page. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceeding of the National Academy of Sciences of the United States of America*, 101:16385 – 16389, November 2004.

[50] A. Hryshko and T. Downs. An implementation of genetic algorithms as a basis for a trading system on the foreign exchange market. *The 2003 Congress on Evolutionary Computation*, 3:1695 – 1701, 2003.

[51] J. hui Yang and X.-Z. Dai. Prediction of shanghai and shenzhen 300 indel based on emd-svm model. *International Conference on Systems and Informatics (ICSAI)*, pages 1209 – 1212, May 2012.

[52] J. C. i Silverstre, Barrio-Castro, and E. Lopex-Bazo. Breaking the panels: an application to gdp per capita. *Econometrics Journal*, 8:159–175, 2005.

[53] S. B. Imandoust and M. Bolandraftar. Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *Journal of Engineering Research and Application*, 3(5):605–610, 2013.

[54] Investopedia. Strong form efficiency, 2019.

[55] Investopedia. Weak form efficiency, 2019.

[56] Investopedia. Why the 10-year u.s. treasury yield matters, 2019.

[57] F. Jahedpari. *Artificial Prediction Markets for Online Prediction of Continuous Variables*. PhD thesis, University of Bath, March 2016.

[58] O. Jan. Weak form market efficiency, 2019.

[59] S. K and K. Sanningammananvara. The efficiency testing of weak form of the indian stock market. *International Journal of Engineering and Management Research*, 4:44–53, 08 2014.

[60] M. G. Kendall. The analysis of economic time-series - part i: Prices. *Journal of the Royal Statistical Society*, 16(1):11–34, 1953.

[61] L. Khaidem, S. Saha, and S. R. Dey. Predicting the direction of stock market prices using random forest. *ArXiv*, abs/1605.00003, 2016.

[62] F. Konak and Y. Seker. The efficiency of developed markets: Empirical evidence from ftse 100. *Journal of Economic Perspectives*, 17(1):59–82, 2014.

[63] R. Kumari, Sheetanshu, M. K. Singh, R. Jha, and N. K. Singh. Anomaly detection in network traffic using k-mean clustering. In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pages 387–393, 2016.

[64] M. B. Kursa and W. R. Rudnicki. Feature selection with the boruta package. *Jounal of Statistical Software*, 36:1–13, 2010.

[65] C.-C. Lee, J.-D. Lee, and C.-C. Lee. Stock prices and the efficient market hypothesis: Evidence from a panel stationary test with structural breaks. *Japan and the World Economy*, 22(1):49 – 58, 2010.

[66] Y. Leu and T. Chiu. An effective stock portfolio trading strategy using genetic algorithms and weighted fuzzy time series. *The 15th Noth-East Asia Symposium on Nana, Information Technology and Reliability (NASNIT)*, pages 70–75, Octorber 2011.

[67] H. Levenbach. Short term forecasting with arima time series models. *CPDF Training and Certification Program*, 2013.

[68] Z. Li and V. Tam. A machine learning view on momentum and reversal trading. *Algorithms*, 11, 2018.

[69] P. K. Mahato and V. Attar. Prediction of gold and silver stock price using prediction of gold and silver stock price using ensemble model. *International Conference on Advances in Engineering and Technology Research (ICAETR)*, pages 1 – 4, August 2014.

[70] B. Malkiel. *A random walk down Wall Street : the time-tested strategy for successful investing.* W.W. Norton, New York, 2007.

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

[72] D. Mallick, V. C. Lee, and Y. S. Ong. An empirical study of genetic programming generated trading rules in computerized stock trading service system. *International Conference on Service System and Service Management*, pages 1–6, June 2008.

[73] C. J. Merz. Using correspondence analysis to combline classifiers. *Machine Learning*, 36:33 –58, 1999.

[74] D. Meyer, F. Leisch, and K. Hornik. The support vector machine under test. *Neurocomputing*, 55(1):169 – 186, 2003. Support Vector Machines.

[75] Min-Ling Zhang and Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *2005 IEEE International Conference on Granular Computing*, volume 2, pages 718–721 Vol. 2, 2005.

[76] P. K. Mishra. Weak from market efficiency: Evidence from emerging and developed world. *Journal of Commerce*, 3(2):26–34, 2011.

[77] R. Nadal, A. Szklo, and A. Lucena. Time-varying impacts of demand and supply oil shocks on correlations between crude oil prices and stock markets indices. *Research in International Business and Finance*, 42:1011 – 1020, 2017.

[78] N. Nguyen-Thanh, D. Marinca, K. Khawam, D. Rohde, F. Vasile, E. S. Lohan, S. Martin, and D. Quadri. Recommendation system-based upper confidence bound for online advertising. *CoRR*, abs/1909.04190, 2019.

[79] N. Nguyen-Thanh, D. Marinca, K. Khawam, D. Rohde, F. Vasile, E. S. Lohan, S. Martin, and D. Quadri. Recommendation system-based upper confidence bound for online advertising. *CoRR*, abs/1909.04190, 2019.

[80] E. Nikolova and R. Sami. A strategic model for in formation market. *ACM Conference on Electronic commerce*, pages 316 – 325, 2007.

[81] S. Nisar and M. Hanif. Testing weak form of efficient market hypothesis: Empirical evidence from south asia. *World Applied Sciences Journal*, 17(4):414–427, 2012.

[82] L. Nunno. Stock market price prediction using linear and polynomial regression models. 2014.

[83] U. D. of the Treasury. Daily treasure long term rate data, 2020.

[84] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. How many trees in a random forest? In P. Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 154–168, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[85] G. A. V. Pai and T. Michel. Evolutionary optimization of constrained k -means clustered assets for diversification in small portfolios. *IEEE Transactions on Evolutionary Computation*, 13(5):1030–1053, Oct 2009.

[86] Y. Pan, E. Haran, S. Manago, and Y. Hu. Co-movement of european stock markets based on association rule mining. *The 3rd International Conference on Data Analystics*, 2014.

[87] A. F. Perold. *The Capital Asset Pricing Model*, volume 18. American Economic Association, 2004.

[88] S. Poshakwale. Evidence on weak form efficiency and day of the week effect in the indian stock market. *Finance India*, X(3):605–616, 1996.

[89] Q. Qin, Q.-G. Wang, J. Li, and S. S. Ge. Linear and nonlinear trading models with gradient boosted random froest and application to singapore stock market. *Journal of Intelligent Learning System and Application (JILSA)*, November 2012.

[90] R. C. Radcliffe. *Investment*. Longman: 3rd revised edition, 1990.

[91] R. Ray. Prediction market and the finance "wisdom of the crowds". *Journal of Behavioral Finance*, pages 2–4, 2006.

[92] B. Reinkensmeyer. Features and fees glossary.

[93] L. Rokach. Ensemble-based classifiers. *Springer Science and Business Media*, 2009.

[94] C. Schoreels, B. Logan, and J. M. Garibaldi. Agent based genetic algorithm employing financial technical analysis for making trading decisions using historical equity market data. *International Conference on Intelligent Agent Technology (IAT)*, pages 421 – 424, 2004.

[95] J. D. Schwager. *Market wizards: interviews with top traders*. HarperCollins, 1993.

[96] N. Sharma and A. Juneja. Combining of random forest estimates using lsboost for stock market index prediction. In *2017 2nd International Conference for Convergence in Technology (I2CT)*, pages 1199–1202, 2017.

[97] W. F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk*. *The Journal of Finance*, 19(3):425–442, 1964.

[98] W. F. Sharpe. Mutual fund performance. *The Journal of Business*, 39:119–138, 1966.

[99] W. F. Sharpe. The sharpe ratio. *The Journal of Portfolio Management*, 21:49–58, 1994.

[100] J. Shen, S. I. Chang, E. S. Lee, Y. Deng, and S. J. Brown. Determination of cluster number in clustering microarray data. *Appl. Math. Comput.*, 169(2):11721185, Oct. 2005.

[101] A. Slivkins. Introduction to multi-armed bandits. *CoRR*, abs/1904.07272, 2019.

[102] P. Somani, S. Talele, and S. Sawant. Stock market prediction using hidden markov model. *Information Technology and Artificial Intelligence*, 2014.

[103] Sonali, B. Maind, and P. Wankar. Research paper on basic of artificial neural network. 2014.

[104] A. Spanjersberg. Combination of different systems for the recognition of handwritten digits. *Joint Conference on Pattern Recognition*, pages 208 – 209, 1974.

[105] S. Sun and R. Huang. An adaptive k-nearest neighbor algorithm. In *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, volume 1, pages 91–94, 2010.

[106] M. Tiwari and B. Misra. Application of cluster analysis in agriculture - a review article. volume 36, 2011.

[107] E. Tomasini and U. Jaekle. *Trading System : A New Approach to System develepement and Portfolio Optimization.* Harriman House, 2009.

[108] T. B. Trafalis and H. Ince. Support vector machine for regression and applications to financial forecasting. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 6, pages 348–353 vol.6, 2000.

[109] J. V. Tu. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 49(11):1225 – 1231, 1996.

[110] M. Velic, T. Grzinic, and I. Padavic. Wisdom of the crowd algorithm for stock market prediction. *Proceeding of the ITI 2013 35th International Conference on Information Technology Interface*, pages 137 – 144, June 2013.

[111] C. Vens. Random forest. In *Encyclopedia of Systems Biology*, pages 1812–1815, 2013.

[112] V. VErma, S. Bhardwaj, and H. Singh. A hybrid k-mean clustering algorithm for prediction analysis. volume 9(28), 2016.

[113] D. Vidal-Toms and A. Ibaez. Semi-strong efficiency of bitcoin. *Finance Research Letters*, 27:259 – 265, 2018.

[114] G. Vidyamurthy. *Pair Trading Quantitative Methods and Analysis.* John Wiley and Sons, 2004.

[115] J.-H. Wang and J.-Y. Leu. Stock market trend prediction using arima-based neural network. *IEEE International Conference on Neural Network*, 4:2160–2165, June 1996.

[116] W. Wang and Y. L. V. A study of the usdx based on arima model - a correlation analysis between the usdx and the shanghai index. *International Conference on Communications and Networks*, pages 49–53, November 2013.

[117] S. Wichaidit. Predicting set50 stock prices using carima (cross correlation arima). *The 19th International Computer Science and Engineering Conference (ICSEC)*, November 2015.

[118] S. Wichaidit, P. Wardkean, and W. Wettayaprasit. New hybrid adaptive ant colony optimization and self-organizing map for dna microarray group findding. *International Conference on Computer Science and Automation Engineering*, pages 444–447, May 2012.

[119] W. Wilder. *New Concepts in Technical Trading Systems.* Hunter Publishing Company, 1978.

[120] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data miningusing correspondence analysis to combline classifiers. *Knowledge and Information Systems*, 14:1–37, 2008.

[121] W. Xu, M. Zuo, M. Zhang, and R. He. Constraint bagging for stock price prediction using neural networks. *Proceeding of 2010 International Conference on Modelling, Identification and Control*, pages 606 – 610, July 2010.

[122] H. Yang, L. Chan, and I. King. Support vector machine regression for volatility stock market prediction. *Intelligent Data Engineering and Automated Learning*, 2412:391–396, 2002. PMID: 14741005.

[123] D. Yates and M. Z. Islam. Fastforest: Increasing random forest processing speed while maintaining accuracy. *Information Sciences*, 557:130 – 152, 2021.

[124] Ying Li and Bo Cheng. An improved k-nearest neighbor algorithm and its application to high resolution remote sensing image classification. In *2009 17th International Conference on Geoinformatics*, pages 1–4, 2009.

[125] G. Zhang, B. Eddy Patuwo, and M. Y. Hu. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, 14(1):35 – 62, 1998.

[126] W.-L. Zhao, C.-H. Deng, and C.-W. Ngo. k-means: A revisit. *Neurocomputing*, 291:195 – 206, 2018.

[127] Y. Zhao, J. Gao, and X. Yang. A servey of neural network ensemble. *International Conference on Neural Networks and Brain*, 1:438 – 442, Octorber 2005.

[128] B. Zhou and J. Hu. A dynamic pattern recognition approach based on neural network for stock time-series. *World Congress on Nature and Biologically Inspired Computing*, pages 1552 – 1555, 2009.

[129] G. Zuckerman. *The Man Who Solved the Market: How Jim Simons Launched the Quant Revolution*. Portfolio, 2019.

Appendices

Appendix A

FTSE 100 Experiment

A.1 List of stocks in FTSE 100

Table A.1: Stock list (Access on 2 Aug. 2019)

Symbol	Company Name	Sector
AAL.L (2000)	Anglo American	Materials
ABF.L (2000)	Associated British Food	Consumer Staples
*ADM.L (2004)	Admiral Group plc	Financials
BDEV.L (2000)	Ashtead Group plc	Industrials
ANTO.L (2000)	Antofagasta plc	Materials
*** AUTO.L (2015)	Auto Trader Group plc	Communications
AV.L (2000)	Aviva	Financials
AVV.L (2000)	Aveva Group plc	Technology
AZN.L (2000)	Astrazeneca plc	Health Care
BA.L (2000)	Bae sys.	Industrials
BARC.L (2000)	Barclays plc	Financials
BATS.L (2000)	British American Tobacco	Consumer Staples
BDEV.L (2000)	Barratt Developments plc	Consumer Discretionary
BHP.L (2000)	BHP Group plc	Materials
BKG.L (2000)	Berkeley Group Holdings plc	Consumer Discretionary
BLND.L (2000)	British Land Co plc	Financials
BNZL.L (2000)	Bunzl plc	Consumer Discretionary
BP.L (2000)	BP plc	Energy

Continued on next page

Table A.1 – *Continued from previous page*

Symbol	Company Name	Sector
BT.A.L (2000)	BT Group plc	Communications
*** III.L (2001)	3I GRP.	Financials
BRBY.L (2002)	Burberry Group plc	Consumer Discretionary
** CCH.L (2013)	Cocacola HBC AG	Consumer Staples
CCL.L (2000)	Carnival plc	Consumer Discretionary
CNA.L (2000)	Centrica plc	Utilities
CPG.L (2000)	Compass Group plc	Consumer Discretionary
CRDA.L (2000)	Croda International plc.	Materials
CRH.L (2000)	CRH plc.	Materials
DCC.L (2000)	DCC plc.	Energy
DGE.L (2000)	Diageo plc	Consumer Staples
** DLG.L (2012)	Direct Line Insurance Group plc.	Financials
** EVR.L (2011)	EVRAZ plc.	Materials
* EXPN.L (2006)	Experian plc	Technology
FERG.L (2000)	Ferguson plc.	Industrials
* FRES.L (2008)	Fresnillo plc	Materials
** GLEN.L (2011)	Glencore plc	Materials
GSK.L (2000)	GlaxoSmithKline plc	Health Care
*** FLTR.L (2019)	Flutter Entertainment plc	Consumer Discretionary
* HL.L (2007)	Hargreaves Lansdown plc	Financials
HLMA.L (2000)	HALMA plc	Industrials
HSX.L (2000)	Hiscox Ltd	Financials
HSBA.L (2000)	HSBC Holdings plc	Financials
IAG.L (2003)	International Consolidated Airlines	Consumer Discretionary
IHG.L (2003)	InterContinental Hotels Group plc	Consumer Discretionary
IMB.L (2000)	Imperial Brands plc	Consumer Staples
INF.L (2000)	Informa plc	Communications
ITRK.L (2002)	Intertek Group plc	Consumer Discretionary
ITV.L (2000)	ITV plc	Communications
*** JD.L (2018)	JD.com Inc	Consumer Discretionary
JMAT.L (2000)	Johnson Matthey plc	Materials
*** JE.L (2014)	Just Eat plc	Technology
KGF.L (2000)	Kingfisher plc	Consumer Discretionary
LAND.L (2000)	Land Securities Group plc	Financials

Continued on next page

Table A.1 – *Continued from previous page*

Symbol	Company Name	Sector
LGEN.L (2000)	Legal & General Group plc	Financials
LLOY.L (2000)	Lloyds Banking Group plc	Financials
LSE.L (2001)	London Stock Exchange Group plc	Financials
* MCRO.L (2005)	Micro Focus International plc	Technology
MKS.L (2000)	Marks & Spencer Group plc	Consumer Discretionary
* MNDI.L (2007)	Mondi plc	Materials
MRO.L (2003)	Melrose Industries plc	Financials
MRW.L (2000)	Wm Morrison Supermarkets plc	Consumer Staples
NG.L (2000)	National Grid plc	Utilities
** NMC.L (2012)	NMC Health plc	Health Care
NXT.L (2000)	Next plc	Consumer Discretionary
** OCDO.L (2010)	Ocado Group plc	Consumer Staples
* PHNX.L (2009)	Phoenix Group Holdings	Financials
PRU.L (2000)	Prudential plc	Financials
PSN.L (2000)	Persimmon plc	Consumer Discretionary
PSON.L (2000)	Pearson plc	Communications
RB.L (2000)	Reckitt Benckiser Group plc	Consumer Staples
RBS.L (2000)	The Royal Bank of Scotland Group plc	Financials
*** RDSA.L (2018)	Royal Dutch Shell plc	Energy
RDSB.L (2000)	Royal Dutch Shell plc	Energy
REL.L (2000)	RELX plc	Technology
RIO.L (2000)	Rio Tinto plc	Materials
* RMV.L (2006)	Rightmove plc	Communications
RR.L (2000)	Rolls-Royce Holdings plc	Industrials
RSA.L (2000)	RSA Insurance Group plc	Financials
RTO.L (2000)	Rentokil Initial plc	Consumer Discretionary
SBRY.L (2000)	J Sainsbury plc	Consumer Staples
SGE.L (2000)	The sage Group plc	Technology
SDR.L (2000)	Schroders plc	Financials
SGRO.L (2000)	Segro plc	Financials
* SKG.L (2007)	Smurfit Kappa Group plc	Materials
* SLA.L (2006)	Standard Life Aberdeen plc	Financials
SMDS.L (2000)	DS Smith plc	Materials
SMIN.L (2000)	Smith Group plc	Industrials

Continued on next page

Table A.1 – *Continued from previous page*

Symbol	Company Name	Sector
*** SMT.L (2017)	Scottish Mortgage Investment Trust plc	Financials
SN.L (2000)	Smith & Nephew plc	Health Care
SPX.L (2000)	Spirax-Sarco Engineering plc	Industrials
SSE.L (2000)	SSE plc	Utilities
STAN.L (2000)	Standard Chartered plc	Financials
STJ.L (2000)	St. James's Place plc	Financials
SVT.L (2000)	Severn Trent plc	Utilities
TSCO.L (2000)	Tesco plc	Consumer Staples
*** TUI.L (2014)	TUI AG	Consumer Discretionary
TW.L (2000)	Taylor Wimpey plc	Consumer Discretionary
ULVR.L (2000)	Unilever plc	Consumer Staples
UU.L (2000)	United Utilities Group plc	Utilities
VOD.L (2000)	Vodafone Group plc	Communications
WTB.L (2000)	Whitbread plc	Consumer Discretionary

A.2 FTSE 100 index's Buy & Hold performance

Table A.2: FTSE 100's Buy & Hold performance

Symbol	Sharpe Ratio	Profit (%)	Annual Volatility	Max DD (%)	Accuracy (%)
AAL.L	0.52	44.58	0.34	-31.93	51.89
ABF.L (2000)	-0.84	-31.66	0.22	-39.45	49.8
*ADM.L (2004)	0.15	5.95	0.19	-18.09	52.31
AHT.L (2000)	-0.04	-2.48	0.28	-34.93	53.41
ANTO.L (2000)	0.14	9.94	0.33	-36.71	52.79
*** AUTO.L (2015)	N/A	N/A	N/A	N/A	N/A
AV.L (2000)	-0.96	-28.89	0.17	-33.95	48.78
AVV.L (2000)	0.22	22.7	0.46	-40.96	53.55
AZN.L (2000)	0.48	26.26	0.23	-21.48	52.0
BA.L (2000)	-0.84	-28.46	0.19	-34.19	47.84
BARC.L (2000)	-1.02	-38.73	0.23	-38.94	48.5
BATS.L (2000)	-1.36	-52.0	0.25	-56.36	46.92
BDEV.L (2000)	-0.11	-6.0	0.26	-38.0	53.35
BHP.L (2000)	0.31	20.32	0.28	-24.55	53.09
BKG.L (2000)	0.34	17.81	0.23	-25.34	54.4
BLND.L (2000)	-0.66	-21.39	0.18	-24.59	52.56
BNZL.L (2000)	0.17	6.24	0.17	-21.46	51.93
BP.L (2000)	-0.22	-8.77	0.2	-18.79	50.2
BT.A.L (2000)	-0.87	-41.19	0.28	-48.86	46.61

Continued on next page

Table A.2 – *Continued from previous page*

Symbol	Sharpe Ratio	Profit (%)	Annual Volatility	Max DD (%)	Accuracy (%)
*** III.L (2001)	N/A	N/A	N/A	N/A	N/A
BRBY.L (2002)	0.17	9.84	0.27	-28.26	55.56
** CCH.L (2013)	0.65	32.44	0.21	-19.89	51.81
CCL.L (2000)	-0.34	-14.87	0.23	-31.99	51.11
CNA.L (2000)	-1.17	-48.47	0.27	-47.37	50.51
CPG.L (2000)	-0.01	-0.39	0.18	-18.92	50.2
CRDA.L (2000)	0.9	40.5	0.18	-16.0	54.56
CRH.L (2000)	-0.82	-32.91	0.23	-32.48	46.41
DCC.L (2000)	-0.17	-7.0	0.21	-28.37	51.51
DGE.L (2000)	0.74	26.37	0.15	-13.6	53.61
** DLG.L (2012)	-0.52	-19.81	0.2	-25.87	54.11
** EVR.L (2011)	0.81	110.55	0.44	-31.78	55.29
* EXPN.L (2006)	0.38	14.94	0.18	-15.19	52.65
FERG.L (2000)	-0.26	-10.3	0.2	-27.56	48.59
* FRES.L (2008)	-0.57	-35.66	0.36	-56.73	49.1
** GLEN.L (2011)	-0.02	-1.04	0.32	-34.72	53.57
GSK.L (2000)	-0.29	-10.62	0.19	-27.83	50.2
*** FLTR.L (2019)	N/A	N/A	N/A	N/A	N/A
* HL.L (2007)	0.82	46.34	0.23	-23.19	58.82
HLMA.L (2000)	0.96	45.89	0.2	-17.75	55.28
HSX.L (2000)	1.14	53.3	0.18	-11.58	52.45
HSBA.L (2000)	-0.21	-7.61	0.18	-24.52	50.6

Continued on next page

Table A.2 – *Continued from previous page*

Symbol	**Sharpe Ratio**	**Profit (%)**	**Annual Volatility**	**Max DD (%)**	**Accuracy (%)**
IAG.L (2003)	0.58	34.08	0.24	-23.34	50.6
IHG.L (2003)	0.14	5.42	0.18	-20.5	54.71
IMB.L (2000)	-1.13	-38.99	0.2	-41.27	46.71
INF.L (2000)	-0.38	-13.41	0.18	-28.59	49.8
ITRK.L (2002)	0.63	31.8	0.21	-27.05	51.81
ITV.L (2000)	-1.12	-45.6	0.25	-44.06	45.78
*** JD.L (2018)	N/A	N/A	N/A	N/A	N/A
JMAT.L (2000)	-0.4	-19.94	0.26	-31.47	50.2
*** JE.L (2014)	N/A	N/A	N/A	N/A	N/A
KGF.L (2000)	-1.15	-46.86	0.26	-44.2	49.6
LAND.L (2000)	-1.21	-35.35	0.17	-34.48	50.1
LGEN.L (2000)	-0.39	-12.79	0.17	-21.84	52.72
LLOY.L (2000)	-0.7	-23.14	0.18	-31.56	46.29
LSE.L (2001)	0.79	33.31	0.18	-19.34	52.63
* MCRO.L (2005)	-0.54	-47.3	0.55	-68.25	52.89
MKS.L (2000)	-0.94	-36.97	0.23	-39.99	50.5
* MNDI.L (2007)	-0.17	-8.04	0.24	-29.16	52.49
MRO.L (2003)	-0.41	-23.23	0.31	-43.99	50.51
MRW.L (2000)	-0.39	-13.65	0.18	-20.84	52.61
NG.L (2000)	-0.96	-32.48	0.19	-36.34	51.3
** NMC.L (2012)	0.89	71.11	0.3	-34.61	51.72
NXT.L (2000)	-0.46	-25.97	0.31	-35.65	49.6

Continued on next page

Table A.2 – *Continued from previous page*

Symbol	Sharpe Ratio	Profit (%)	Annual Volatility	Max DD (%)	Accuracy (%)
** OCDO.L (2010)	1.02	193.04	0.51	-34.57	49.4
* PHNX.L (2009)	-0.65	-20.85	0.17	-25.77	49.9
PRU.L (2000)	-0.55	-21.19	0.21	-33.13	51.0
PSN.L (2000)	0.04	1.87	0.24	-36.08	55.47
PSON.L (2000)	-0.02	-1.64	0.34	-30.79	54.07
RB.L (2000)	-0.48	-18.77	0.21	-32.87	49.9
RBS.L (2000)	-0.2	-9.61	0.24	-32.87	48.69
*** RDSA.L (2018)	N/A	N/A	N/A	N/A	N/A
RDSB.L (2000)	-0.17	-6.68	0.19	-20.15	52.68
REL.L (2000)	0.12	3.6	0.15	-19.78	52.06
RIO.L (2000)	0.19	11.3	0.27	-22.4	52.5
* RMV.L (2006)	0.12	4.66	0.18	-21.33	54.03
RR.L (2000)	0.29	18.16	0.27	-30.6	49.3
RSA.L (2000)	-0.55	-18.44	0.18	-27.16	50.71
RTO.L (2000)	0.86	45.67	0.21	-22.96	54.42
SBRY.L (2000)	0.01	0.21	0.23	-23.6	52.94
SGE.L (2000)	-0.32	-14.27	0.23	-36.01	54.86
SDR.L (2000)	-0.79	-26.04	0.18	-39.23	51.9
SGRO.L (2000)	0.74	28.31	0.16	-13.81	53.12
* SKG.L (2007)	0.07	4.42	0.31	-41.25	51.3
* SLA.L (2006)	-1.15	-45.7	0.25	-56.0	48.59
SMDS.L (2000)	-0.65	-27.27	0.23	-45.78	47.4

Continued on next page

Table A.2 – *Continued from previous page*

Symbol	Sharpe Ratio	Profit (%)	Annual Volatility	Max DD (%)	Accuracy (%)
SMIN.L (2000)	-0.28	-11.39	0.21	-28.96	51.01
*** SMT.L (2017)	N/A	N/A	N/A	N/A	N/A
SN.L (2000)	0.36	13.63	0.17	-15.09	51.94
SPX.L (2000)	0.85	43.05	0.21	-19.67	54.6
SSE.L (2000)	-1.09	-36.45	0.2	-33.69	50.31
STAN.L (2000)	-0.32	-14.27	0.23	-38.79	49.01
STJ.L (2000)	-0.39	-14.86	0.2	-29.5	53.69
SVT.L (2000)	-0.64	-24.38	0.21	-33.29	49.7
TSCO.L (2000)	-0.32	-14.49	0.23	-28.79	50.4
*** TUI.L (2014)	N/A	N/A	N/A	N/A	N/A
TW.L (2000)	-0.39	-17.12	0.23	-38.64	54.93
ULVR.L (2000)	0.41	18.69	0.2	-18.76	51.39
UU.L	-0.62	-24.27	0.21	-37.88	51.72
VOD.L	-0.8	-29.58	0.2	-39.53	48.3
WTB.L	0.3	15.18	0.22	-18.46	48.8

A.3 Results of FTSE 100

Table A.3: Stock list (Access on 2 Aug. 2019)

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
AAL.L	Sharpe Ratio	-0.46	1.13	1.25	1.3	1.34	1.36	1.37	1.39	1.4	1.4
	Profit	-23.35	52.72	59.55	62.97	65.02	66.39	67.36	68.09	68.66	69.12
	Accuracy	51.11	48.84	48.84	48.84	48.84	48.84	48.84	48.84	48.84	48.84
ABF.L	Sharpe Ratio	-1.57	0.31	0.75	0.84	0.9	0.93	0.96	0.98	0.99	1.01
	Profit	-6.42	4.03	9.08	10.28	11.01	11.49	11.83	12.09	12.29	12.45
	Accuracy	50.0	31.25	29.41	29.41	29.41	29.41	29.41	29.41	29.41	29.41
* ADM.L	Sharpe Ratio	0.25	0.61	0.7	0.74	0.77	0.79	0.8	0.81	0.82	0.82
	Profit	7.41	15.15	17.48	18.64	19.34	19.8	20.14	20.39	20.58	20.74
	Accuracy	66.67	68.75	68.75	68.75	68.75	68.75	68.75	68.75	68.75	68.75
AHT.L	Sharpe Ratio	0.19	0.22	0.23	-0.36	-0.28	-0.23	-0.19	-0.61	-0.58	-0.56
	Profit	10.45	12.3	12.92	-14.1	-11.22	-9.3	-7.92	-20.99	-20.19	-19.55
	Accuracy	75.0	75.0	75.0	52.0	52.0	52.0	52.0	48.75	48.75	48.75
ANTO.L	Sharpe Ratio	0.27	0.48	0.23	0.35	0.43	0.47	0.5	0.53	0.55	0.56
	Profit	6.1	10.95	13.08	20.77	25.38	28.46	30.65	32.3	33.58	34.61
	Accuracy	58.33	58.33	52.83	52.83	52.83	52.83	52.83	52.83	52.83	52.83
*** AUTO.L	Sharpe Ratio	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Profit	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Accuracy	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
AV.L	Sharpe Ratio	-4.86	-2.3	-0.05	0.12	0.22	0.29	0.33	0.37	0.39	0.42
	Profit	-78.1	-14.95	-0.9	1.87	3.53	4.64	5.43	6.02	6.49	6.85
	Accuracy	44.44	57.14	48.78	48.78	48.78	48.78	48.78	48.78	48.78	48.78
AVV.L	Sharpe Ratio	0.19	0.44	0.71	0.83	0.9	0.95	0.98	1.0	1.02	1.04
	Profit	11.74	24.79	43.16	52.35	57.86	61.54	64.16	66.13	67.66	68.89
	Accuracy	58.46	54.81	54.81	54.81	54.81	54.81	54.81	54.81	54.81	54.81
AZN.L	Sharpe Ratio	0.41	0.19	0.45	0.67	0.8	0.88	0.93	0.98	1.01	1.03
	Profit	21.87	9.21	15.11	23.29	28.19	31.47	33.8	35.56	36.92	38.01
	Accuracy	77.78	61.29	56.48	56.48	56.48	56.48	56.48	56.48	56.48	56.48
BA.L	Sharpe Ratio	-0.87	-0.83	-0.82	-0.81	-0.81	-0.8	-0.8	-0.8	-0.8	-0.8
	Profit	-4.1	-3.91	-3.85	-3.82	-3.8	-3.78	-3.78	-3.77	-3.76	-3.76
	Accuracy	50.25	50.29	50.29	50.29	50.29	50.29	50.29	50.29	50.29	50.29
BARC.L	Sharpe Ratio	-4.22	-0.83	-0.57	-0.44	-0.36	-0.31	-0.28	-0.25	-0.23	-0.21
	Profit	-69.17	-14.46	-10.05	-7.84	-6.52	-5.64	-5.01	-4.54	-4.17	-3.87
	Accuracy	44.59	39.39	39.39	39.39	39.39	39.39	39.39	39.39	39.39	39.39
BATS.L	Sharpe Ratio	-3.07	-1.18	0.08	0.28	0.39	0.46	0.52	0.55	0.58	0.61
	Profit	-69.39	-38.19	2.33	7.94	11.31	13.55	15.15	16.36	17.29	18.04
	Accuracy	49.46	43.33	47.56	47.56	47.56	47.56	47.56	47.56	47.56	47.56
BDEV.L	Sharpe Ratio	0.38	0.7	0.79	0.9	0.92	0.85	0.88	0.9	0.92	0.93
	Profit	11.63	22.05	25.52	32.73	33.78	32.92	34.26	35.27	36.05	36.67

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	54.17	54.17	54.17	54.17	54.17	54.69	54.69	54.69	54.69	54.69
BHP.L	Sharpe Ratio	0.32	0.55	0.62	0.66	0.68	0.7	0.71	0.72	0.72	0.73
	Profit	4.41	7.5	8.53	9.05	9.36	9.56	9.71	9.82	9.91	9.98
	Accuracy	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14
BKG.L	Sharpe Ratio	0.37	0.61	0.68	0.5	0.74	0.38	0.39	0.57	0.58	0.42
	Profit	7.58	12.42	14.04	9.75	15.33	7.57	7.87	11.24	11.41	8.41
	Accuracy	36.36	36.36	36.36	50.0	36.36	46.67	46.67	50.0	50.0	46.67
BLND.L	Sharpe Ratio	-3.73	-2.83	-2.82	-2.82	-0.7	-0.6	-0.53	-0.48	-0.43	-0.4
	Profit	-67.43	-7.89	-7.58	-7.42	-17.87	-15.49	-13.79	-12.51	-11.52	-10.72
	Accuracy	49.44	50.0	50.0	50.0	48.42	48.42	48.42	48.42	48.42	48.42
BNZL.L	Sharpe Ratio	-0.29	-0.1	0.25	0.31	0.35	0.37	0.39	0.68	0.71	0.74
	Profit	-4.08	-1.42	7.09	8.84	9.89	10.59	11.1	13.18	13.93	14.53
	Accuracy	50.0	50.0	45.83	45.83	45.83	45.83	45.83	56.25	56.25	56.25
BP.L	Sharpe Ratio	-1.11	-1.05	-0.55	-0.24	0.18	0.28	0.35	0.4	0.44	0.47
	Profit	-13.64	-12.88	-12.9	-5.82	5.04	7.8	9.78	11.25	12.4	13.32
	Accuracy	50.0	50.0	50.5	50.5	52.08	52.08	52.08	52.08	52.08	52.08
BT.A.L	Sharpe Ratio	0.18	0.35	0.41	0.44	0.61	0.63	0.65	0.66	0.67	0.68
	Profit	2.66	5.22	6.07	6.5	13.18	13.65	13.98	14.23	14.42	14.58
	Accuracy	50.0	50.0	50.0	50.0	43.75	43.75	43.75	43.75	43.75	43.75
*** III.L	Sharpe Ratio	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Profit	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
BRBY.L	Sharpe Ratio	0.77	0.93	0.99	1.01	1.03	1.04	1.04	1.05	1.05	1.06
	Profit	11.23	13.73	14.57	14.98	15.23	15.4	15.52	15.61	15.68	15.73
	Accuracy	83.33	83.33	83.33	83.33	83.33	83.33	83.33	83.33	83.33	83.33
CCL.L	Sharpe Ratio	-0.38	-0.69	-0.63	-0.6	-0.58	-0.57	-0.56	-0.26	-0.22	-0.19
	Profit	-16.33	-3.4	-3.14	-3.01	-2.94	-2.88	-2.85	-6.85	-5.79	-4.94
	Accuracy	85.71	100.0	100.0	100.0	100.0	100.0	100.0	55.0	55.0	55.0
CNA.L	Sharpe Ratio	-5.19	-1.06	-0.84	-0.74	-0.68	-0.63	-0.6	-0.58	-0.56	-0.55
	Profit	-94.78	-9.13	-7.37	-6.49	-5.96	-5.61	-5.36	-5.17	-5.02	-4.9
	Accuracy	43.56	30.77	30.77	30.77	30.77	30.77	30.77	30.77	30.77	30.77
** CCH.L	Sharpe Ratio	0.55	0.61	0.63	0.64	0.65	0.58	0.62	0.65	0.68	0.69
	Profit	26.91	30.14	31.22	31.76	32.08	24.77	26.68	28.12	29.24	30.13
	Accuracy	42.86	42.86	42.86	42.86	42.86	48.94	48.94	48.94	48.94	48.94
CPG.L	Sharpe Ratio	-0.22	-0.12	-0.08	-0.07	-0.06	-0.06	0.03	0.1	0.15	0.19
	Profit	-7.68	-4.15	-2.97	-2.38	-2.03	-1.61	0.7	2.44	3.79	4.87
	Accuracy	66.67	66.67	66.67	66.67	66.67	55.83	55.83	55.83	55.83	55.83
CRH.L	Sharpe Ratio	-0.94	-0.56	-0.31	-0.2	-0.13	-0.09	-0.05	-0.03	-0.01	0.0
	Profit	-34.81	-16.66	-9.67	-6.17	-4.07	-2.67	-1.68	-0.93	-0.34	0.12
	Accuracy	50.0	32.79	32.79	32.79	32.79	32.79	32.79	32.79	32.79	32.79
CRDA.L	Sharpe Ratio	0.81	0.85	0.43	0.56	0.64	0.69	0.73	0.76	1.08	1.11
	Profit	36.0	37.39	11.05	14.72	16.92	18.39	19.44	20.22	34.37	35.4

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	100.0	72.73	55.56	55.56	55.56	55.56	55.56	55.56	58.33	58.33
DCC.L	Sharpe Ratio	-0.08	-0.02	-0.37	-0.42	-0.26	-0.16	-0.09	-0.04	-0.0	0.03
	Profit	-2.28	-0.59	-9.96	-11.37	-7.28	-4.56	-2.61	-1.15	-0.01	0.9
	Accuracy	25.0	25.0	58.14	60.75	60.75	60.75	60.75	60.75	60.75	60.75
DGE.L	Sharpe Ratio	0.23	0.75	0.81	0.55	0.52	0.62	0.68	0.73	0.77	0.8
	Profit	6.22	15.54	16.71	14.59	13.54	16.15	18.01	19.41	20.5	21.37
	Accuracy	55.56	50.0	50.0	62.16	61.63	61.63	61.63	61.63	61.63	61.63
**DLG.L	Sharpe Ratio	-1.77	-1.66	-1.62	-0.85	-0.76	-0.7	-0.65	-0.62	0.1	0.11
	Profit	-12.44	-11.52	-11.22	-27.21	-24.61	-22.88	-21.64	-20.71	2.57	2.86
	Accuracy	50.0	50.0	50.0	48.65	48.65	48.65	48.65	48.65	51.61	51.61
**EVR.L	Sharpe Ratio	-3.1	0.7	0.74	0.76	0.77	0.78	0.78	0.79	0.79	0.79
	Profit	-9.37	59.41	63.21	65.1	66.24	67.0	67.54	67.94	68.26	68.51
	Accuracy	nan	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
*EXPN.L	Sharpe Ratio	0.44	0.49	0.65	0.83	0.93	1.13	1.19	1.23	1.26	1.29
	Profit	15.45	17.25	13.87	17.94	20.39	30.71	32.46	33.77	34.79	35.61
	Accuracy	75.0	75.0	61.82	61.82	61.82	61.54	61.54	61.54	61.54	61.54
FERG.L	Sharpe Ratio	-5.09	-0.16	-0.25	-0.16	-0.07	0.01	0.07	0.11	0.14	0.16
	Profit	-85.68	-6.31	-9.58	-6.21	-2.43	0.32	2.29	3.76	4.91	5.83
	Accuracy	44.07	40.0	43.4	43.4	48.04	48.04	48.04	48.04	48.04	48.04
*** FLTR.L	Sharpe Ratio	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Profit	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
*FRES.L	Sharpe Ratio	0.25	0.84	1.02	1.11	1.16	1.19	1.21	1.23	1.25	1.26
	Profit	7.39	26.89	33.39	36.64	38.59	39.89	40.82	41.52	42.06	42.49
	Accuracy	43.18	43.18	43.18	43.18	43.18	43.18	43.18	43.18	43.18	43.18
GSK.L	Sharpe Ratio	-0.44	0.15	0.41	0.53	0.6	0.65	0.69	0.71	0.73	0.75
	Profit	-9.53	2.71	7.49	9.88	11.31	12.27	12.95	13.46	13.86	14.18
	Accuracy	64.29	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25	56.25
**GLEN.L	Sharpe Ratio	-0.09	-0.02	-0.0	0.17	0.27	0.33	0.37	0.41	0.43	0.45
	Profit	-2.15	-0.56	-0.03	7.63	12.15	15.16	17.31	18.93	20.18	21.18
	Accuracy	100.0	100.0	100.0	48.08	48.08	48.08	48.08	48.08	48.08	48.08
HLMA.L	Sharpe Ratio	0.96	0.99	0.99	1.39	1.44	1.47	1.49	1.51	1.52	1.53
	Profit	45.37	46.8	47.28	56.22	58.63	60.23	61.38	62.24	62.91	63.44
	Accuracy	100.0	100.0	100.0	55.32	55.32	55.32	55.32	55.32	55.32	55.32
*HL.L	Sharpe Ratio	1.16	1.23	1.25	1.26	1.27	1.28	1.28	1.28	1.28	1.28
	Profit	52.54	56.19	57.41	58.02	58.38	58.63	58.8	58.93	59.03	59.11
	Accuracy	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14	57.14
HSX.L	Sharpe Ratio	1.12	0.72	0.89	0.97	1.02	1.05	1.07	1.09	1.1	1.11
	Profit	51.76	18.61	23.39	25.78	27.21	28.17	28.85	29.36	29.76	30.08
	Accuracy	nan	53.12	53.12	53.12	53.12	53.12	53.12	53.12	53.12	53.12
HSBA.L	Sharpe Ratio	-6.67	-0.24	-0.68	-0.43	-0.29	-0.19	-0.13	-0.08	-0.04	-0.01
	Profit	-89.84	-7.72	-9.37	-6.06	-4.08	-2.76	-1.81	-1.1	-0.55	-0.11

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	50.94	33.33	46.0	46.0	46.0	46.0	46.0	46.0	46.0	46.0
IMB.L	Sharpe Ratio	0.3	0.38	0.41	0.42	0.43	0.44	0.44	0.44	0.45	0.45
	Profit	3.47	4.46	4.79	4.96	5.06	5.12	5.17	5.21	5.23	5.25
	Accuracy	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
INF.L	Sharpe Ratio	-0.46	-0.45	0.53	0.68	0.77	0.82	0.86	0.89	0.91	0.93
	Profit	-15.76	-15.42	16.84	21.94	25.01	27.05	28.51	29.6	30.46	31.14
	Accuracy	100.0	100.0	53.62	53.62	53.62	53.62	53.62	53.62	53.62	53.62
IHG.L	Sharpe Ratio	0.11	0.11	0.02	0.02	0.03	0.1	0.17	0.23	0.27	0.3
	Profit	4.15	4.36	0.69	0.91	1.03	3.18	5.49	7.23	8.58	9.66
	Accuracy	nan	nan	66.67	66.67	66.67	48.72	48.72	48.72	48.72	48.72
ITRK.L	Sharpe Ratio	0.99	1.05	1.07	1.08	1.09	1.09	1.09	1.1	1.1	1.1
	Profit	38.89	41.66	42.58	43.04	43.32	43.5	43.64	43.74	43.81	43.87
	Accuracy	83.33	83.33	83.33	83.33	83.33	83.33	83.33	83.33	83.33	83.33
IAG.L	Sharpe Ratio	0.47	0.6	0.64	0.66	0.67	0.68	0.68	0.69	0.69	0.69
	Profit	25.1	32.7	35.24	36.5	37.26	37.77	38.13	38.4	38.61	38.78
	Accuracy	52.94	52.94	52.94	52.94	52.94	52.94	52.94	52.94	52.94	52.94
ITV.L	Sharpe Ratio	-1.93	-1.06	-0.42	-0.14	0.02	0.12	0.19	0.24	0.28	0.31
	Profit	-16.97	-30.78	-13.35	-4.64	0.59	4.08	6.57	8.44	9.89	11.05
	Accuracy	0.0	41.59	41.59	41.59	41.59	41.59	41.59	41.59	41.59	41.59
*** JD.L	Sharpe Ratio	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Profit	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
JMAT.L	Sharpe Ratio	-3.92	0.5	0.59	0.63	0.66	0.67	0.69	0.69	0.7	0.71
	Profit	-78.45	14.05	16.68	17.99	18.78	19.3	19.68	19.96	20.18	20.35
	Accuracy	52.04	47.37	47.37	47.37	47.37	47.37	47.37	47.37	47.37	47.37
*** JE.L	Sharpe Ratio	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Profit	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Accuracy	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
KGF.L	Sharpe Ratio	-2.87	-3.17	-3.01	-0.36	-0.04	0.02	0.06	0.1	0.12	0.14
	Profit	-75.6	-5.47	-5.28	-14.67	-1.87	0.76	2.64	4.05	5.14	6.02
	Accuracy	49.0	100.0	100.0	48.36	51.89	51.89	51.89	51.89	51.89	51.89
LAND.L	Sharpe Ratio	-2.13	-2.06	-2.03	-2.01	-2.01	-2.0	-1.99	-1.99	-1.99	-1.99
	Profit	-8.76	-8.38	-8.26	-8.19	-8.16	-8.13	-8.11	-8.1	-8.09	-8.08
	Accuracy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LGEN.L	Sharpe Ratio	-4.58	-1.81	-1.16	-0.74	-0.5	-0.35	-0.25	-0.17	-0.11	-0.07
	Profit	-7.84	-14.53	-26.22	-17.47	-12.22	-8.72	-6.22	-4.34	-2.88	-1.71
	Accuracy	50.0	33.33	45.97	45.97	45.97	45.97	45.97	45.97	45.97	45.97
LLOY.L	Sharpe Ratio	-1.02	-0.65	-0.36	-0.22	-0.14	-0.08	-0.05	-0.02	0.01	0.02
	Profit	-118.67	-4.74	-2.69	-1.66	-1.04	-0.63	-0.33	-0.11	0.06	0.2
	Accuracy	44.14	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0
LSE.L	Sharpe Ratio	0.76	0.89	1.19	1.9	1.99	2.04	2.1	2.14	2.17	2.19
	Profit	32.06	11.45	15.69	69.17	73.19	75.87	67.06	68.72	70.02	71.05

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	100.0	62.07	62.07	53.16	53.16	53.16	57.14	57.14	57.14	57.14
MKS.L	Sharpe Ratio	-4.77	0.25	0.56	0.43	0.53	0.6	0.64	0.68	0.71	0.73
	Profit	-84.04	4.31	13.49	11.92	14.98	17.02	18.48	19.57	20.42	21.1
	Accuracy	51.72	48.28	44.64	42.47	42.47	42.47	42.47	42.47	42.47	42.47
MRO.L	Sharpe Ratio	-0.41	-0.35	-0.33	-0.26	-0.19	0.06	0.09	0.11	0.13	0.14
	Profit	-22.43	-19.57	-18.61	-12.46	-9.21	2.61	4.09	5.2	6.06	6.75
	Accuracy	44.44	44.44	44.44	52.13	52.13	55.81	55.81	55.81	55.81	55.81
* MCRO.L	Sharpe Ratio	0.3	0.4	0.43	0.45	0.46	0.46	0.47	0.47	0.47	0.48
	Profit	11.7	15.7	17.03	17.7	18.1	18.37	18.56	18.7	18.81	18.9
	Accuracy	33.33	33.33	33.33	33.33	33.33	33.33	33.33	33.33	33.33	33.33
* MNDI.L	Sharpe Ratio	-0.74	-0.02	0.21	0.32	0.39	0.43	0.46	0.49	0.5	0.52
	Profit	-5.93	-0.13	1.8	2.77	3.35	3.74	4.01	4.22	4.38	4.51
	Accuracy	71.43	71.43	71.43	71.43	71.43	71.43	71.43	71.43	71.43	71.43
MRW.L	Sharpe Ratio	-0.14	-0.07	0.16	0.25	0.44	0.53	0.59	0.63	0.67	0.69
	Profit	-0.79	-0.4	2.37	3.71	8.79	10.54	11.79	12.72	13.45	14.03
	Accuracy	100.0	100.0	60.0	60.0	57.14	57.14	57.14	57.14	57.14	57.14
NG.L	Sharpe Ratio	-4.61	-1.69	-0.43	-0.33	-0.28	-0.24	-0.21	-0.19	-0.18	-0.16
	Profit	-85.97	-6.86	-3.39	-2.64	-2.19	-1.89	-1.67	-1.51	-1.39	-1.29
	Accuracy	42.35	0.0	36.36	36.36	36.36	36.36	36.36	36.36	36.36	36.36
NXT.L	Sharpe Ratio	-1.84	-1.55	-1.45	-1.41	-1.38	-1.36	-1.35	-1.34	-1.33	-1.33
	Profit	-4.32	-3.94	-3.81	-3.74	-3.71	-3.68	-3.66	-3.65	-3.64	-3.63

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	51.22	51.22	51.22	51.22	48.04	48.04	48.04	48.04	48.04	48.04
** NMC.L	Sharpe Ratio	0.85	0.87	0.87	0.88	0.88	0.88	0.88	0.88	0.88	0.88
	Profit	66.43	68.18	68.76	69.06	69.23	69.35	69.43	69.49	69.54	69.58
	Accuracy	75.0	75.0	75.0	75.0	75.0	75.0	75.0	75.0	75.0	75.0
** OCDO.L	Sharpe Ratio	0.06	0.76	0.92	0.99	1.03	1.06	1.08	1.1	1.11	1.12
	Profit	5.1	45.47	122.08	136.7	145.48	151.33	155.5	158.64	161.08	163.02
	Accuracy	55.14	54.55	55.14	55.14	55.14	55.14	55.14	55.14	55.14	55.14
PSON.L	Sharpe Ratio	-3.84	-0.41	-0.39	-0.55	-0.51	-0.48	-0.46	-0.5	-0.46	-0.42
	Profit	-79.84	-20.92	-20.09	-26.88	-24.95	-23.67	-22.75	-14.14	-12.9	-11.9
	Accuracy	54.26	66.67	66.67	60.47	60.47	60.47	60.47	48.7	48.7	48.7
PSN.L	Sharpe Ratio	-0.06	0.67	0.69	0.7	0.71	0.69	0.33	0.35	0.36	0.37
	Profit	-2.94	23.41	24.37	24.86	25.14	12.49	14.74	15.56	16.2	16.72
	Accuracy	100.0	100.0	100.0	100.0	100.0	83.33	51.67	51.67	51.67	51.67
* PHNX.L	Sharpe Ratio	-0.61	-0.43	-0.37	-0.34	0.08	0.13	0.17	0.19	0.22	0.23
	Profit	-7.46	-5.32	-4.6	-4.25	1.79	3.05	3.95	4.62	5.14	5.56
	Accuracy	40.0	40.0	40.0	40.0	42.55	42.55	42.55	42.55	42.55	42.55
PRU.L	Sharpe Ratio	-0.26	-0.16	-0.12	-0.11	0.23	0.29	0.32	0.26	0.37	0.39
	Profit	-3.46	-2.08	-1.62	-1.39	4.65	5.72	6.48	5.32	7.5	7.86
	Accuracy	66.67	66.67	66.67	66.67	45.95	45.95	45.95	50.0	45.95	45.95
*** RDSA.L	Sharpe Ratio	/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Profit	/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RDSB.L	Sharpe Ratio	0.13	0.42	0.51	0.56	0.59	0.61	0.62	0.63	0.64	0.64
	Profit	2.69	9.0	11.1	12.15	12.78	13.2	13.5	13.72	13.9	14.04
	Accuracy	46.67	46.67	46.67	46.67	46.67	46.67	46.67	46.67	46.67	46.67
RB.L	Sharpe Ratio	0.64	0.74	0.77	0.78	0.79	0.8	0.8	0.65	0.69	0.72
	Profit	9.25	10.53	10.96	11.18	11.3	11.39	11.45	16.48	17.41	18.16
	Accuracy	66.67	66.67	66.67	66.67	66.67	66.67	66.67	45.57	45.57	45.57
REL.L	Sharpe Ratio	-0.23	0.05	0.14	0.13	0.22	0.27	0.31	0.34	0.36	0.38
	Profit	-4.2	0.67	2.29	2.6	4.31	5.45	6.26	6.87	7.35	7.73
	Accuracy	58.33	58.33	58.33	50.0	50.0	50.0	50.0	50.0	50.0	50.0
RTO.L	Sharpe Ratio	1.56	1.7	1.75	1.77	1.78	1.79	1.8	1.8	1.81	1.81
	Profit	74.33	82.94	85.81	87.24	88.11	88.68	89.09	89.4	89.64	89.83
	Accuracy	68.75	68.75	68.75	68.75	68.75	68.75	68.75	68.75	68.75	68.75
* RMV.L	Sharpe Ratio	0.09	0.1	0.37	0.74	0.83	0.9	0.94	0.97	0.99	1.01
	Profit	3.54	3.96	14.06	24.67	28.18	30.53	32.2	33.46	34.44	35.22
	Accuracy	100.0	100.0	58.33	62.67	62.67	62.67	62.67	62.67	62.67	62.67
RIO.L	Sharpe Ratio	0.06	-0.24	0.22	0.42	0.54	0.61	0.66	0.7	0.73	0.75
	Profit	3.5	-9.82	10.3	20.35	26.39	30.41	33.29	35.44	37.12	38.46
	Accuracy	66.67	47.24	47.24	47.24	47.24	47.24	47.24	47.24	47.24	47.24
RR.L	Sharpe Ratio	-1.72	-0.31	0.02	0.17	0.26	0.31	0.35	0.38	0.27	0.28
	Profit	-56.22	-13.26	1.07	8.23	12.53	15.39	17.44	18.97	5.63	5.95

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	49.11	49.11	49.11	49.11	49.11	49.11	49.11	49.11	68.57	68.57
RBS.L	Sharpe Ratio	-3.72	0.16	0.34	0.43	0.48	0.51	0.53	0.55	0.57	0.58
	Profit	-76.47	4.68	10.4	13.27	14.98	16.13	16.95	17.56	18.04	18.42
	Accuracy	46.15	55.26	55.26	55.26	55.26	55.26	55.26	55.26	55.26	55.26
RSA.L	Sharpe Ratio	-2.03	-0.91	-0.85	-0.68	-0.52	-0.41	-0.34	-0.29	-0.25	-0.21
	Profit	-17.88	-27.81	-26.17	-16.43	-12.78	-10.34	-8.6	-7.29	-6.28	-5.47
	Accuracy	62.5	43.75	43.75	43.62	43.62	43.62	43.62	43.62	43.62	43.62
SGE.L	Sharpe Ratio	-0.28	-0.26	-0.25	0.04	0.08	0.1	0.11	0.13	0.13	0.14
	Profit	-12.4	-11.66	-11.42	1.04	1.84	2.38	2.77	3.05	3.28	3.46
	Accuracy	50.0	50.0	50.0	31.58	31.58	31.58	31.58	31.58	31.58	31.58
SBRY.L	Sharpe Ratio	-0.94	-0.66	-0.57	-0.52	0.92	0.96	0.99	1.01	1.02	1.04
	Profit	-3.58	-2.6	-2.27	-2.11	43.01	45.25	46.84	48.04	48.97	49.71
	Accuracy	100.0	100.0	100.0	100.0	52.94	52.94	52.94	52.94	52.94	52.94
SDR.L	Sharpe Ratio	-2.39	-1.35	-0.94	-0.64	-0.47	-0.35	-0.28	-0.22	-0.18	-0.14
	Profit	-50.39	-31.61	-22.74	-17.19	-12.8	-9.88	-7.79	-6.22	-5.0	-4.03
	Accuracy	53.57	53.57	54.55	53.98	53.98	53.98	53.98	53.98	53.98	53.98
*** SMT.L	Sharpe Ratio	/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Profit	/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Accuracy	/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SGRO.L	Sharpe Ratio	0.91	1.02	0.77	0.8	0.82	0.84	0.85	0.85	0.86	0.86
	Profit	26.39	29.99	23.19	24.3	24.96	25.4	25.72	25.96	26.14	26.29

Continued on next page

Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	42.86	42.86	35.71	35.71	35.71	35.71	35.71	35.71	35.71	35.71
SVT.L	Sharpe Ratio	-6.16	-0.63	-0.41	-0.31	-0.24	-0.2	-0.17	-0.15	-0.13	-0.12
	Profit	-90.86	-3.59	-2.39	-1.79	-1.43	-1.19	-1.02	-0.89	-0.79	-0.71
	Accuracy	55.56	55.56	55.56	55.56	55.56	55.56	55.56	55.56	55.56	55.56
SN.L	Sharpe Ratio	-4.5	0.25	0.36	0.41	0.44	0.46	0.47	0.49	0.49	0.5
	Profit	-6.08	2.32	3.33	3.83	4.13	4.33	4.48	4.59	4.67	4.74
	Accuracy	100.0	71.43	71.43	71.43	71.43	71.43	71.43	71.43	71.43	71.43
SMDS.L	Sharpe Ratio	0.01	0.18	0.23	0.26	0.27	0.29	-0.14	-0.11	-0.09	-0.07
	Profit	0.38	6.37	8.36	9.36	9.96	10.36	-5.05	-3.96	-3.11	-2.43
	Accuracy	43.75	43.75	43.75	43.75	43.75	43.75	50.59	50.59	50.59	50.59
SMIN.L	Sharpe Ratio	0.54	0.47	0.53	0.56	0.58	0.59	0.6	0.6	0.61	0.61
	Profit	15.3	12.27	14.03	14.92	15.44	15.8	16.05	16.24	16.38	16.5
	Accuracy	20.0	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08
* SKG.L	Sharpe Ratio	0.11	0.21	0.24	0.26	0.27	0.27	0.28	0.28	0.28	0.29
	Profit	4.28	8.1	9.37	10.0	10.38	10.64	10.82	10.95	11.06	11.14
	Accuracy	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
SPX.L	Sharpe Ratio	1.02	1.05	1.06	1.07	0.75	0.76	0.76	0.76	0.76	0.77
	Profit	23.43	24.25	24.52	24.66	19.17	19.32	19.42	19.5	19.57	19.62
	Accuracy	100.0	100.0	100.0	100.0	50.0	50.0	50.0	50.0	50.0	50.0
SSE.L	Sharpe Ratio	-0.65	-0.24	-0.39	-0.26	-0.19	-0.14	-0.1	-0.07	-0.05	-0.04
	Profit	-3.64	-1.44	-3.82	-2.62	-1.9	-1.43	-1.08	-0.83	-0.63	-0.47

Continued on next page

Table A.3 – *Continued from previous page*

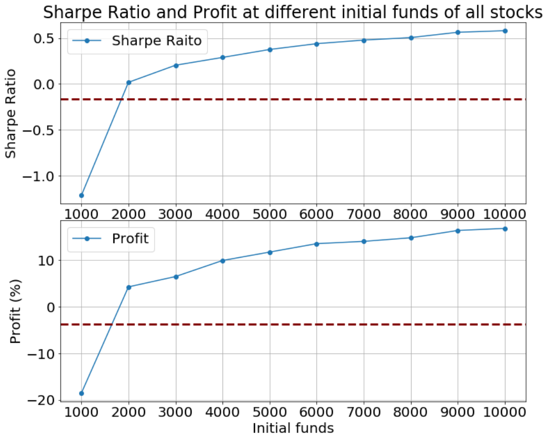
Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	0.0	0.0	38.89	38.89	38.89	38.89	38.89	38.89	38.89	38.89
STJ.L	Sharpe Ratio	-0.46	-0.72	-0.67	-0.65	-0.63	-0.62	-0.62	-0.55	-0.61	-0.6
	Profit	-17.22	-25.27	-23.73	-22.96	-22.5	-22.19	-21.97	-19.45	-21.67	-21.57
	Accuracy	100.0	60.0	60.0	60.0	60.0	60.0	60.0	63.64	60.0	60.0
STAN.L	Sharpe Ratio	-3.68	-0.1	0.35	0.55	0.67	0.74	0.8	0.83	0.86	0.89
	Profit	-74.29	-3.32	12.4	20.26	24.98	28.12	30.37	32.05	33.36	34.41
	Accuracy	43.14	42.34	42.34	42.34	42.34	42.34	42.34	42.34	42.34	42.34
* SLA.L	Sharpe Ratio	-1.03	-0.87	-0.82	-0.79	-0.78	-0.76	-0.76	-0.75	-0.75	-0.74
	Profit	-11.0	-9.25	-8.67	-8.38	-8.21	-8.09	-8.01	-7.95	-7.9	-7.86
	Accuracy	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
TW.L	Sharpe Ratio	-0.15	-0.27	-0.22	-0.2	-0.18	-0.17	-0.16	-0.16	-0.16	-0.15
	Profit	-1.51	-3.08	-2.52	-2.24	-2.07	-1.96	-1.88	-1.82	-1.77	-1.73
	Accuracy	100.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
TSCO.L	Sharpe Ratio	-2.92	-0.04	0.05	0.1	0.12	0.25	0.29	0.32	0.35	0.37
	Profit	-71.09	-0.92	1.06	2.05	2.65	6.97	8.27	9.24	10.0	10.61
	Accuracy	49.48	33.33	33.33	33.33	33.33	53.73	53.73	53.73	53.73	53.73
*** TUI.L	Sharpe Ratio	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Profit	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Accuracy	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
ULVR.L	Sharpe Ratio	-0.13	0.04	0.0	0.18	0.28	0.35	0.4	0.43	0.46	0.48
	Profit	-1.96	0.66	0.12	5.05	8.0	9.97	11.38	12.44	13.26	13.91

Continued on next page

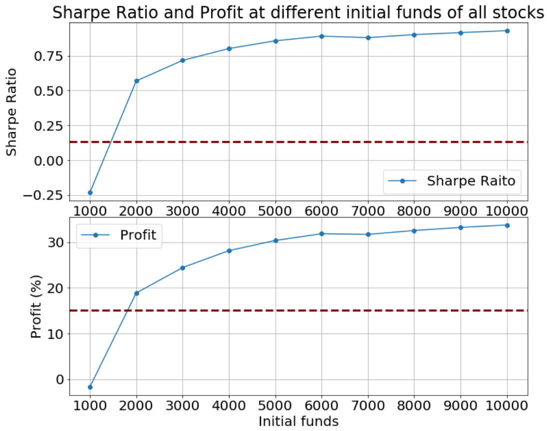
Table A.3 – *Continued from previous page*

Stock	Metrics	Initial funds (£)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
	Accuracy	66.67	66.67	54.05	54.05	54.05	54.05	54.05	54.05	54.05	54.05
UU.L	Sharpe Ratio	0.15	0.75	0.96	1.06	1.12	1.16	1.19	1.21	1.23	1.24
	Profit	1.87	10.07	13.11	14.64	15.55	16.16	16.6	16.92	17.18	17.38
	Accuracy	55.56	52.38	52.38	52.38	52.38	52.38	52.38	52.38	52.38	52.38
VOD.L	Sharpe Ratio	-5.75	-0.19	-0.12	-0.08	-0.05	-0.04	-0.03	-0.02	-0.01	-0.01
	Profit	-86.35	-3.04	-1.79	-1.16	-0.79	-0.54	-0.36	-0.23	-0.12	-0.04
	Accuracy	47.06	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0
WTB.L	Sharpe Ratio	-1.68	0.51	0.63	0.77	0.88	0.94	0.99	1.02	1.05	1.07
	Profit	-49.09	14.34	18.02	28.85	33.14	36.01	38.05	39.59	40.78	41.73
	Accuracy	57.58	52.17	52.17	52.33	52.33	52.33	52.33	52.33	52.33	52.33

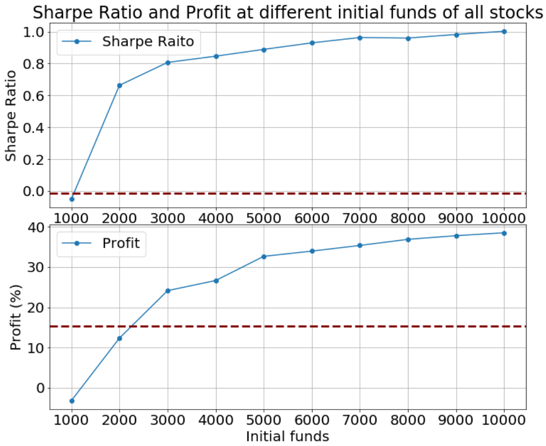
A.3.0.1 Individual sector



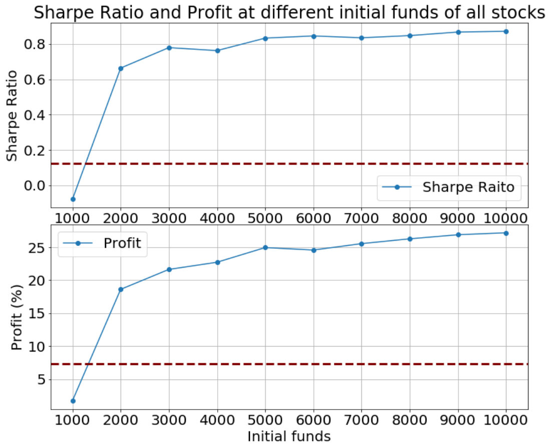
(a) Financial sector



(b) Material sector

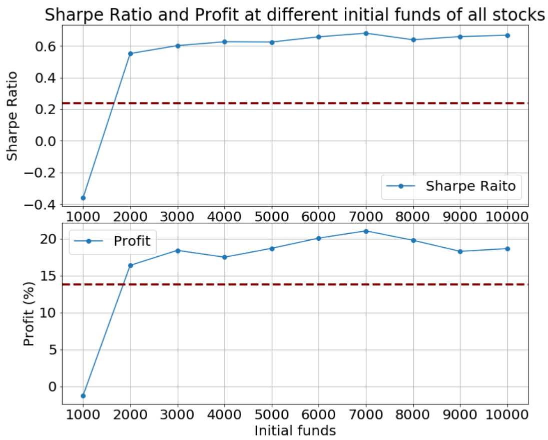


(c) Consumer Staple sector

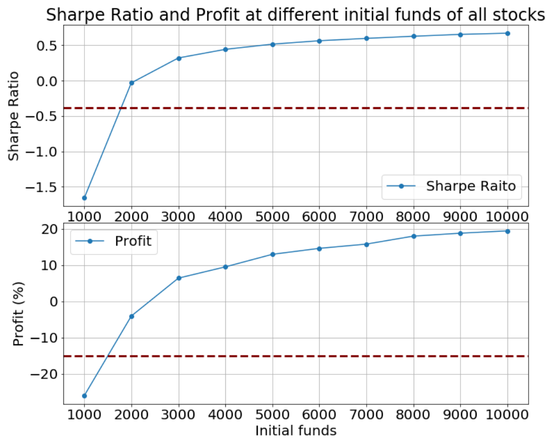


(d) Consumer Discretionary sector

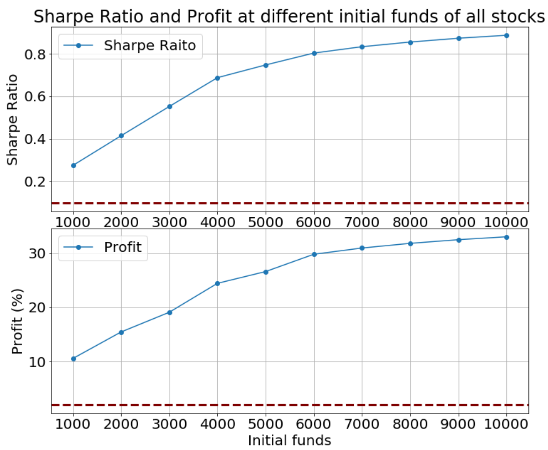
Figure A-1: Results sectors Financial, Material, Consumer Staple and Consumer Discretionary



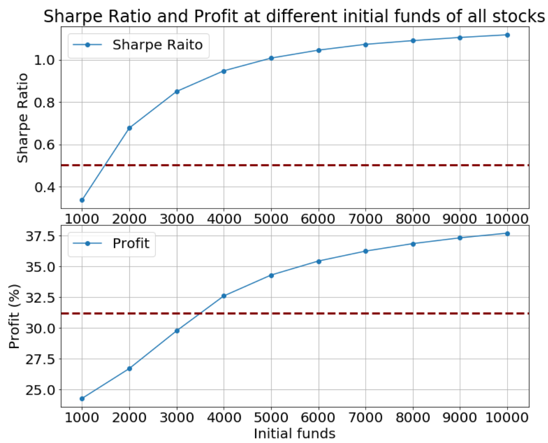
(a) Industrial sector



(b) Communication sector

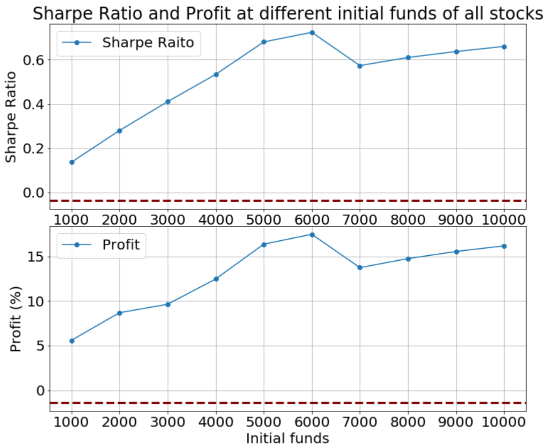


(c) Technology sector

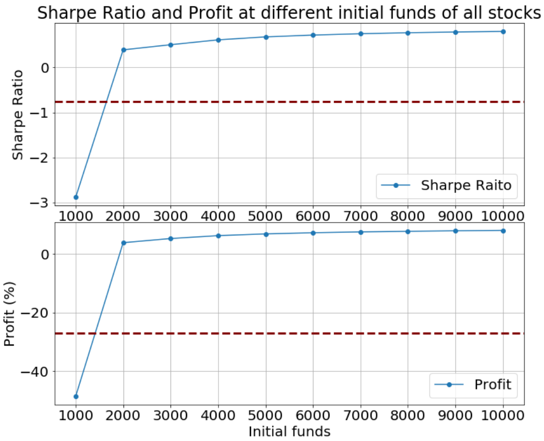


(d) Health Care sector

Figure A-2: Results sectors Industrial, Communication, Technology and Health Care



(a) Industrial sector



(b) Communication sector

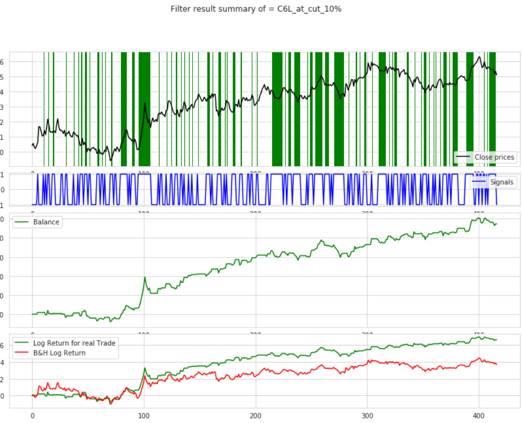
Figure A-3: Results sectors Industrial and Communication

Appendix B

Additional results from the comparison section

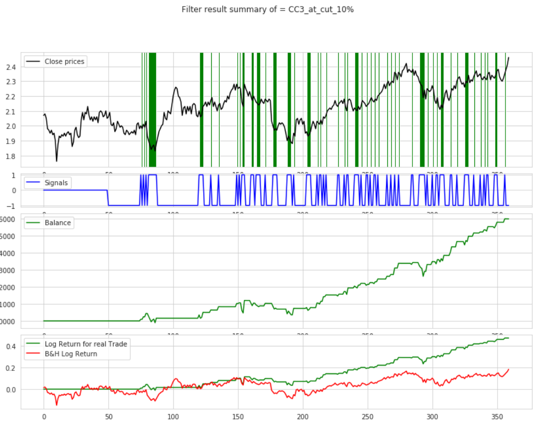
B.1 Result from dataset from Qin's system

This section shows the results from our system testing on the same datasets as Qin et al. research [89] in section 6.4.2 on page 277. The stocks tested were: Capitalland (C31), DBS (D05), UOB (U11), SGX-Singapore Exchange (S68), Starhub (CC3), Singtel (Z74), Semb Corp (U96), and and SIA-Singapore Airline (C6L). The testing period is between 2009 and 2010 (417 trading days, ending 2010-08-31). Results of these 8 stocks can be seen in Figures B-1 and B-2

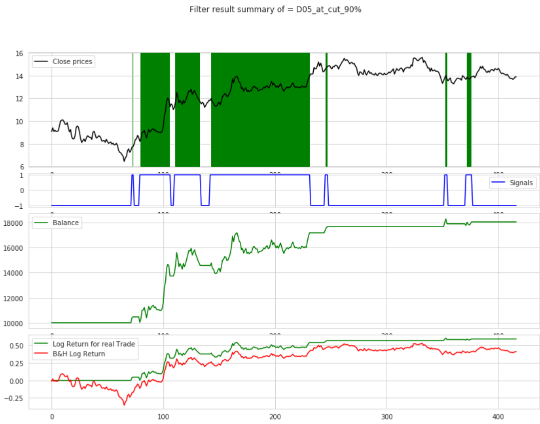


(a) C6L

(b) C31

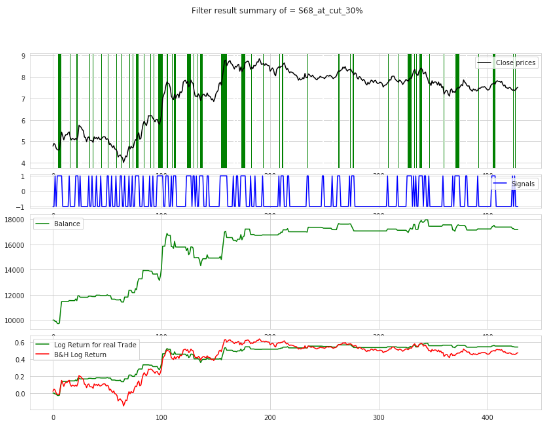


(c) CC3

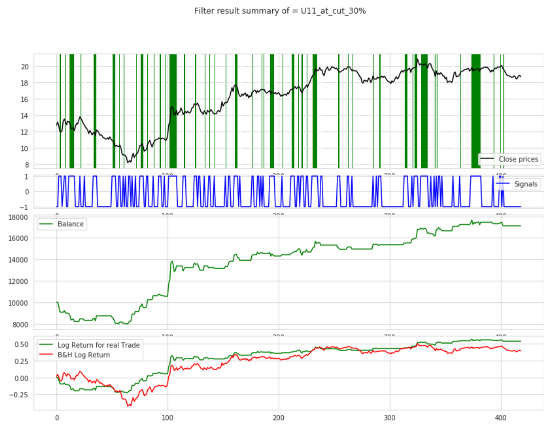


(d) D05

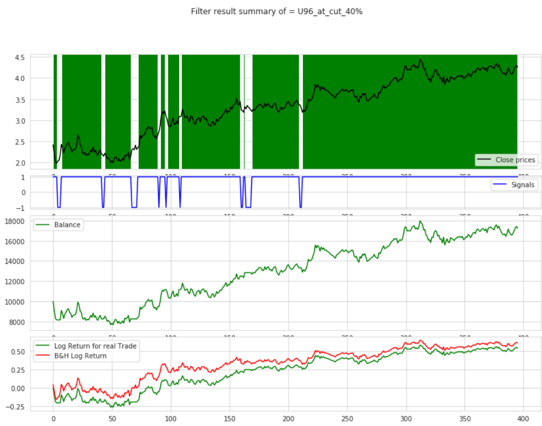
Figure B-1: Results from C6L, C31, CC3 and D05



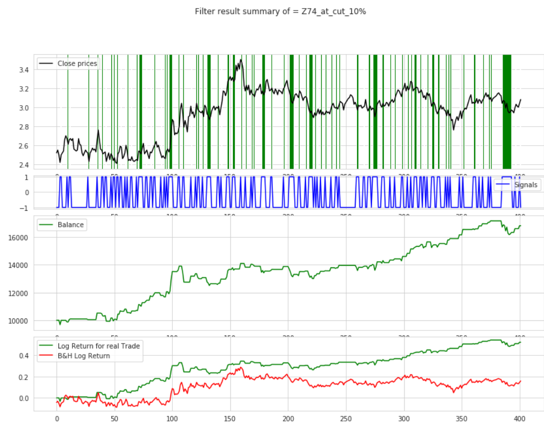
(a) S68



(b) U11



(c) U96



(d) Z74

Figure B-2: Clustering features

Appendix C

Institutional Trading: Long - Short Strategy

Stock market trading can be divided into two styles: institutional and retail trading. Institutions are companies such as investment banks, hedge funds and mutual funds, participants who invest the most in the markets and often manage money on behalf of others. Retail traders are individuals primarily investing for their own account. Trading may not necessarily be their primary source of income. In this work we focused on the latter, who we also refer to as individual investors. We present the reasons why we selected individual investors below, but in this section we explain how our approach could be adapted for an institutional context.

1. Supportive: We would like to support traders with limited amounts of funds to start trading responsibly. For some, sensible trading offers a chance for financial independence. There are a lot of traders who would like to start trading with this goal or others in mind, but often they cannot afford to open an account with a big broker. Many brokers have minimum deposits required to open an account, which are often too much for many people as are the additional fees which come later. We support the idea that everyone should be given the opportunity to make money, that people learn best by their own experiences and can judge for themselves if trading is for them. Therefore, we aimed for a system that can profit with limited initial funds and in spite of trading fees.

2. Realistic: We realise that designing parameters to mimic institutional trading is extremely difficult for outsiders with no experience. For example, trading fees may be negotiated with investment banks or stock exchanges, and there might be other

fees or interest payments (institutional traders may use leverage; they can borrow money to make larger profits) that remain unrevealed to the public. We also cannot know the capital allocated to individual traders or trading desks within institutions, some of which will be allocated to hedging positions. Moreover, the ability to go short for individual investors is almost impossible without using alternative products such as spread bets. Most brokers do not offer a short-selling service, especially the now-popular online platforms which form one of our focus groups. In addition to our support of helping individual investors to trade with their broker of choice, we believe attempting to mimic institutional trading may be very unrealistic.

3. Goes with the trend: Nowadays, there are a lot of online trading platforms for people who would like to trade from home, and these platforms have grown significantly in recent decades as technology has improved also. People are able to learn how to trade by themselves or trade as a way to exercise their own interests. We would like to be part of this trend as this is a most interesting area for us too. Therefore, we decided to build a system that people can realistically follow. For example, one that is able to connect with a broker to buy and sell online.

4. Risk control: Shares are convex products, by which we mean a share price can only decrease to 0, therefore the maximum loss (assuming a trader has not borrowed money to invest) is the total invested in the shares, but the potential profit is unlimited. On the other hand, as the price of a share can increase unlimitedly, short-selling shares can cause a loss greater than the total invested, such that one ends up owing money. Stop-losses are rarely guaranteed and can be missed in fast-moving markets. As our system does not have a module to manage risk, we decided to go with long only for our individual trading system, which for the aforementioned reasons is inherently less risky.

Although we made the decision to create our system for individual investors, we understand that one might want to know how it may perform for an institutional client. Therefore, we include this appendix as a feasibility study to give an idea of whether our system may be appropriate for institutional traders. What follows are the adaptations we made in order to show the results from our system as if traded institutionally.

- **Long & Short signal**: Normally, our system is long only. This means that once the system gets a signal 1, it will send out the buy order. But once it gets a signal -1, it will only close existing long positions and stay in cash. However, if we want to imitate the institutional trading, a reasonable first step is to include short selling. So we have taken signal -1 as a signal to short sell. Once the system gets a signal -1, it will not only close an existing long position, but it will also open a new short-selling position.

- **Trading Fee**: In our research, we used a flat rate brokerage fee, which is realistic for individual investors. However, to mimic institutional trading, the fee needs to be changed. Please note that we are aware that there might be some hidden fees or interest involved that we are unable to find information about. We also expect in reality that trading fees for institutions are negotiated based on the ongoing business relationship, liquidity of the shares being purchased or any number of other factors. In this simulation, we will only add the most common fee, which is a brokerage fee. The brokerage fee here changes from a flat rate to a percentage. The percentage we are going to use is 0.2%, as most of the brokers publish this as their standard fee.

- **Initial funds**: The initial funds available here will be changed to £1000000, as institutional traders have more equity to trade with.

- **Signal Optimisation**: As our system is aimed at individual investors, we have added a module for signal optimisation. The signal optimisation module filters some signals out if they are not worth taking with the amount of money an investor has. However, institutional traders with more capital should not need this optimisation and may make all the trades suggested by the system. Therefore, it is fair to simulate trading without signal filtering. However, the results when the signals are filtered will be shown as well for the sake of comparison.

In order to demonstrate the results, we have selected 5 stocks, comprising 3 positive and 2 negative-performing stocks. Positive and negative stocks are compared against the buy & hold return during the testing period, from the beginning of 2017 to the end of 2018.

Table C.1 shows the results of five stocks when our system is traded as though an institutional trader. The results presented are the Sharpe Ratio, profit, volatility, maximum drawdown and accuracy. The trading period spanned 2 years, starting from the beginning of 2017 until the end of 2018. The first column indicates the performance of the buy & hold strategy, while the next 2 columns show the performance from our system before and after performing signal filtering.

The results show that the system could not make a profit without signal filtering. All of

Performance	Buy & Hold	Before Signal filtering	After Signal Filtering
AAL			
Sharpe Ratio	0.61	-0.42	0.7
Profit (%)	50.67	-16.66	60.83
Volatility	0.3390	0.34	0.339
Maximum Drawdown	-31.93	-54.11	-29.52
Accuracy	-	47.71	50.7
Number of Trades	2	214	88
ABF			
Sharpe Ratio	-0.7	-0.28	1.28
Profit (%)	-25.57	-12.13	68.49
Volatility	0.215	0.216	0.214
Maximum Drawdown	-39.45	-28.06	-17.28
Accuracy	-	50.4	51.82
Number of Trades	2	229	36
AHT			
Sharpe Ratio	0.23	-1.35	0.46
Profit (%)	3.16	-51.38	30.18
Volatility	0.276	0.277	0.276
Maximum Drawdown	-34.93	-61.74	-30.13
Accuracy	-	47.79	54.22
Number of Trades	2	201	10
AV			
Sharpe Ratio	-0.78	-0.64	1.17
Profit (%)	-22.8	-17.86	48.64
Volatility	0.17	0.171	0.168
Maximum Drawdown	-33.95	-28.9	-11.29
Accuracy	-	46.73	50.82
Number of Trades	2	217	84
BA			
Sharpe Ratio	-0.68	-0.92	0.71
Profit (%)	-22.37	-29.03	30.04
Volatility	0.193	0.194	0.192
Maximum Drawdown	-34.19	-38.36	-21.73
Accuracy	-	47.02	52.16
Number of Trades	2	198	2

Table C.1: Performances summary AAL, ABF, AHT, AV and BA from the beginning of 2017 to the end of 2018 £1000000

the results ended up negative. This suggests that even though institutional traders can take more chances to make a profit, as they can take all signals, it is costly for them to do so. The signals the system provided comprised both weak and strong signals which can be detected by our optimisation module. Without optimisation, investors need to open positions following every signal, including the weak ones, but clearly the small profit available is not worth the risk. This explains the poor results in the second column.

As mentioned, our system is able to filter out the weak signals. This was necessary for individual traders with little capital. The details can be seen in section 5.5.2 on page 212. We performed another experiment for the institutional system but this time filtering the weak signals. Then leftover signals were all followed, opening both long and short positions. The results turned out much better for every dataset, as shown in the third column. Most of the results ended up with much higher profit than the buy only strategy (the results can be seen in table A.1 on page 298), especially BA. For BA, the long only system made almost no profit at all and ended up with only 2.33% profit over two years. However, with the long & short strategy available to an institution, it ended up over 30% in profit. The results from AHT were also much better too. With the long only strategy, it ended up with a -13.46% loss, but when trading with the institutional long & short strategy, it made a profit of 48.64% over the two-year period. However, this was not the case for AAL. For AAL, the buy only strategy made 75.21% but when trading with the buy & sell strategy, the profit decreased to 60.83%.

From these results, we conclude that our system is able to make more profit by trading both long & short signals (1 and -1). However, it must be borne in mind that it has to include the signal optimisation process. Once the weak signals are filtered out, for both long and short, the system has a chance to make more profit. Also, there might be some more fees and interests that need to be taken into account, so our results may be optimistic. Some further suggestions for improvement when adapting our trading system to trade using the long & short strategy include:

- **Consider more hidden issues**: As we have no experiences with institutional trading, there may be information - for example other types of charges - that we have overlooked. Therefore, these could be taken into account.
- **Redesign the optimisation process**: The optimisation module of our system works by considering the limitation of initial funds. With less funds, the system will try to take less risk, resulting in trading less often. However, for institutional trading, limited funds may not be a problem and our method of filtering may not be optimal. Therefore, we suggest finding new criteria to filter the signals out as we have already seen that this process is very important.
- **Prepare plenty of funds to start with!**

Appendix D

Multiple Stock Training

In this research, we train our predictors on a single stock basis. This means that for every stock included, we will collect historical data and use it to train our predictors. These trained predictors will then be used to predict a test set from the same stock. There are reasons we selected to train predictors on single stocks. Firstly, we would like our predictors to be able to recognise specific patterns which only happen in the stock of our interest. If we train them with other stocks (even including the focus stock itself), it is possible that our predictors will be subjected or overfit to other patterns which are not useful when we test our predictions on the stock initially selected. Secondly, time is not a problem here. As we have shown in table 5.4 on page 208, the training process is very quick, so it will be no problem at all to regularly retrain predictors, which will be important for investors using the system. However, we do not retrain the predictors in this research. We will only train them once per stock.

One might suggest, given the training process is so quick, training on multiple stocks at the same time. Therefore, the predictors will only be trained once but theoretically can be used to predict any stock. This would lead to another question, that of how many we should train. For example, if we plan to use our system in the UK, we might train predictors on all FTSE 100 stocks. However, given so much data the predictor is likely to discover, and use, spurious correlations during the training period that do not exist in practise, and hence fail during testing. We ran a little experiment to demonstrate the effect of training our predictors on a multiple stock basis.

This section takes the top 5 stocks from table 5.2 on page 153, which are CARR, COST, D4T4, FERG and GOOG. This table shows the performance of our predictors, which were trained on a single stock basis. There are two predictors included in this experiment. They were trained on a single stock using the training data from the year 2000 to 2012. Then, these trained predictors were taken to predict the same stock but for different years (year 2013-2014). The result from this algorithm was shown in *Statistical Selection* strategy (the fourth line of the results from each stock) in the original table (table 5.2 on page 153). However, in this section, this result will be shown as *Statistical Selection*.

In this experiment, we take the same stocks from the top of this table and set every condition the same way. The only difference is that instead of training our predictors on a single stock basis, we train our predictors with all stocks in one go. These predictors will then be used perform prediction on each stock individually later. Here are the comparisons of the new training method (shows as *Multiple Training*) and the previous one (shows as *Statistical Selection*).

Table D.1: The comparison between Buy & Hold, Single predictor, Multiple predictors (with bias) and Statistical Selection (Multiple predictor with bias removal) methods

Data & Model	**Profit (%)**	**Sharpe Ratio**	**Max DD (%)**	**Accuracy (%)**	**Stdev**
CARR					
Buy & Hold	62.38	0.77	-26.32	52.89	0.302
Statistical Selection	74.01	0.89	-33.23	49.89	0.302
Multiple Training	-45.09	-1.17	-48.75	47.75	0.302
COST					
Buy & Hold	11.57	0.09	-22.26	51.26	0.296
Statistical Selection	-10.94	-0.31	-25.88	49.79	0.296
Multiple Training	-29.71	-0.73	-34.96	48.95	0.296
D4T4					
Buy & Hold	39.10	1.19	-37.76	51.75	0.586
Statistical Selection	105.02	2.69	-31.57	57.89	0.579
Multiple Training	91.74	2.43	-28.07	57.02	0.581
FERG					
Buy & Hold	18.59	0.26	-16.09	49.10	0.211
Statistical Selection	-43.56	-1.51	-45.74	48.7	0.210
Multiple Training	-27.65	-0.91	-36.35	49.1	0.211
GOOG					
Buy & Hold	45.71	0.74	-18.72	50.10	0.213
Statistical Selection	5.12	0.02	-24.23	47.91	0.213

Continued on next page

Table D.1 – *Continued from previous page*

Data & Model	Profit (%)	Sharpe Ratio	Max DD (%)	Accuracy (%)	Stdev
Multiple Training	4.41	0.18	-17.06	48.11	0.213

Table D.1 shows that when changing the training method from a single-stock basis to a multiple-stock basis the predictors do not work as well. Most of the results from multiple stock training are much worse than the single stock training method, especially CARR. CARR is one of the stocks that our model worked very well with. The buy & hold strategy also did well. However, the model trained on multiple stocks ended up with a big loss of -45.09%. This is not surprising because we thought training our predictors with highly fluctuated data (data from many stocks) would make our system worse as the predictors could not recognise specific characteristics included in the data of the stock of interest. In fact, they try to learn to recognise broad characteristics that might hold across stocks. With these results and the observation above that the training time is not a problem, we cannot see any benefits from including the multiple-stock training method in our trading system. Therefore, we used a single-stock training basis.

Appendix E

Statistical Results

Here we will provide the detail of statistical tests we have run on our results from table 5.3 on page 187.

	Two-layer scoring system	Buy-and-hold benchmark
Mean	1.65545	0.23182
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00074**	

(a) Buy & hold

	Two-layer scoring system	Polynomial regression
Mean	1.65545	-0.06182
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00294**	

(b) Polynomial Regression

	Two-layer scoring system	Linear regression
Mean	1.65545	-0.11364
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00236**	

(c) Linearn Regression

	Two-layer scoring system	SVR-RBF
Mean	1.65545	0.27364
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00045**	

(d) SVR(rbf)

	Two-layer scoring system	SVR-Linear
Mean	1.65545	-0.02364
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00126**	

(e) SVR(linear)

	Two-layer scoring system	SVR-Polynomial
Mean	1.65545	-0.17727
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00023**	

(f) SVR(Polynomial)

	Two-layer scoring system	Random Forest
Mean	1.65545	0.25636
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00063**	

(g) Random Forest

	Two-layer scoring system	Averaging
Mean	1.65545	0.09273
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00016**	

(h) Average

	Two-layer scoring system	UCB
Mean	1.65545	-0.01909
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00009**	

(i) UCB

	Two-layer scoring system	One-layer scoring system
Mean	1.65545	-0.15000
Observations	11.00000	11.00000
P(T<=t) two-tail	**0.00066**	

(j) 1L Scoing

Figure E-1: P-value when comparing our model (two-layer scoring system) with other strategies