MACHINE LEARNING BASED TRADING STRATEGIES FOR THE CHINESE STOCK MARKET

Thesis submitted in accordance with the requirements of

the University of Liverpool for the degree of

Doctor of Philosophy

by

Juan Du

Department of Mathematical Sciences

University of Liverpool

August 2019

Declaration

I, **Juan Du**, declare that this thesis titled, **Machine learning based trading strategies for the Chinese stock market** and the work presented in it are my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed: Juan Du

Juan Du (Candidate)

August 2019

MACHINE LEARNING BASED TRADING STRATEGIES FOR THE CHINESE STOCK MARKET

Juan Du

Abstract

This thesis focuses on the machine learning based trading strategies of China Exchange Traded Funds (ETFs). Machine learning and artificial intelligence (AI) provide an innovative level of service for financial forecasting, customer service and data security. Through the development of automated investment advisors powered by machine learning technology, financial institutions such as JPMorgan, the Bank of America and Morgan Stanley have recently achieved AI investment forecasting. This thesis intends to provide original insights into machine learning based trading strategies by producing trading signals based on forecasts of stock price movements.

Theories and models associated with algorithmic trading, price forecasting and trading signal generation are considered; in particular machine learning models such as logistic regression, support vector machine, neural network and ensemble learning methods. Each potential profitable strategy of the China ETFs is tested, and the risk-adjusted returns for corresponding strategies are analysed in detail.

The primary aim of this thesis is to develop two machine learning based trading strategies, in which machine learning models are utilised to predict trading signals. Each machine learning model and their combinations are employed to generate trading signals according to one day ahead forecasts, demonstrating that the final excess return does not cover the transaction costs. This encourages us to reduce the number of unprofitable trades in the trading system by adopting the 'multi-day forecasts' in place of the 'one day ahead forecasts'. Therefore, investors benefit from a longer prediction horizon, in which more predicted information of the total number of upward (or downward) price movements is provided. Investors can make trading decisions based on the majority of the predicted trading signals within the prediction horizon. Moreover, this method of trading rules is consistent with the industry practice. The strategy is flexible to allow risk-averse investors and risk-loving investors to make different trading decisions.

A multi-day forecast based trading system through random forest yields positive risk-adjusted returns after transaction costs. It is identified that it is possible that some machine learning techniques can successfully assist individuals in navigating their decision-making activities.

Acknowledgements

I wish to express my deepest gratitude to my primary Ph.D. supervisor, Professor Ahmet Göncü, for supporting me during these three years. Ahmet was able to guide me positively throughout the creation process. Ahmet inspired me to try something new to solve the real problems through programming, and I really appreciate that. I am also very grateful to Dr. Chao Zhou and Prof. Min Dai for their inspirations whenever I have confusions about asset pricing. I should say thanks to my co-supervisors Dr. Nan Zhang, Dr. Hongsong Chou and Dr. Hirbod Assa for their academic suggestions from industry perspectives. Without the help of my leader Wenjing Li when I took a summer internship, it is impossible for me to become proficient in analysing Vector Autoregressive Model.

Much gratitude goes to the most important people in my life - my parents Chunfa Du and Baohua Wang for their loves. Their unwavering support has been invaluable throughout my graduate studies. I cherish the time I spent with my colleagues. They were crucial in my motivation process. With the company of my friends, I feel energetic and never feel lonely in my spare time.

This work also benefited from financial support from Xi'an Jiaotong -Liverpool University with the PGRS Ref No RDF-15-01-21. With it I was able to participate in several conferences in which I presented my research, meanwhile got access to some recent studies in quantitative finance.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The chapter starts by introducing algorithmic trading strategies with several aims to be achieved, and then briefly reviews all methods used in this thesis to discover trading opportunities and evaluate trading strategies, and discusses how these methods are used. The chapter concludes with the thesis structure.

1.1 Research aims

Machine learning based trading, which is an interdisciplinary area that combines machine learning techniques and finance knowledge into developing trading strategies. These trading strategies should be of interest to both hedgers and speculators who seek to trade using machine learning models. This thesis also contributes to the academic literature as it provides empirical evidence on the forecasting and trading power of a wide variety of nonlinear machine learning models for the China ETFs. Three research aims are:

- to develop profitable trading strategies that take into account transaction costs. More importantly, this research aims to develop a robust trading system that remains valid with regard to different assets and time periods.

- to investigate whether machine learning models are able to generate more profitable trading signals compared with using statistical-based models and knowledge-based subjective models.

- to verify whether machine learning based trading strategies outperform two benchmarks, i.e. a passive trading strategy (buy-and-hold) and a traditional econometric forecasting method such as Autoregressive model (AR model) in terms of financial performance.

1.2 Research approach

The most common methods that investors use to analyse the benefits and risks associated with stock investments are fundamental analysis and technical analysis. Among these analyses, technical analysis is the main method utilised in all trading strategies in this study. Technical analysis is using past information to forecast future trends. The price trend is capable of being determined by some patterns in a chart. One example is a chart pattern trading strategy, in which a bull template was utilised as a pattern to generate buy signals. Moreover, for technical traders, it is common to employ technical indicators such as moving average convergence divergence (MACD) to trade. I also developed a MACD trend following strategy that uses technical trading rules to trigger trading signals. In addition, in machine learning based strategies, all machine learning models were trained with technical indicators to predict future price movements.

After using technical analysis to identify trading opportunities, back-testing is applied as the main technique to evaluate the trading performances in the past for all developed trading strategies. Back-testing is accomplished by reconstructing with historical data and trades that would have occurred in the past using rules defined by a given strategy. Back-testing can provide plenty of valuable statistical feedback about a given system. The universal back-testing statistics include net profit or loss measures (excess return[1] and annualised returns), volatility measures (standard deviation and maximum drawdown), and risk-adjusted return measures (Sharpe ratio and Sortino ratio).

All back-testing statistics mentioned above are used to evaluate the trading performances of the developed trading strategies, and the various trading strategies are compared from the perspective of daily and annual performance measures. The daily performance measures to differentiate the economic significance of the return series based on every day's log returns are average return, standard deviation, t-Statistic of right-tailed t-tests, minimum, median, maximum, skewness and kurtosis. In general, the daily measurements show the property of distributions for daily returns. Specifically, *daily average return* demonstrates the profitability while *daily standard deviation* shows the risk magnitude; *t-Statistic* investigates whether a trading strategy can yield significant positive daily returns, and the other estimators measure whether there are some profitable opportunities in some extreme cases.

The annual performance measures to compare all trading strategies are *annual return*, *excess return*, *standard deviation*, *downside deviation*, *Sharpe ratio*, *Sortino ratio*, and *maximum drawdown. Annual return* and *excess return* reflect the ability of generating final profits. *Excess return* shows the extra return excluding the risk-free deposit rate. The annual risk-free interest rate is 1.75%[2]. *Standard deviation*, *downside deviation* and *maximum drawdown* show the risk

[1] Returns in excess of the fixed deposit interest rate.

[2] Based on the annual interest rate of 'Bank of China 1 Year Time Deposit' on 6 August 2019, refer to https://china.financialadvisory.com/time-deposit.html

magnitude of annualised returns. *Downside deviation* is a measure of downside risk that focuses on negative returns. *Maximum drawdown* is defined as the fall in the total return curve from the previous maximum. The *Maximum drawdown* corresponds to the maximum loss experienced during the entire period. The smaller the value of the maximum drawdown, the better the strategy is. When comparing two strategies, the one with the higher *Sharpe ratio* and *Sortino ratio* is more preferable as it indicates more return for the same risk, and translates into greater risk-adjusted performance. That is to say, *Sharpe ratio* and *Sortino ratio* consider whether investors are able to be well-compensated for taking this risk.

Cumulative return plot of each trading strategy is provided as another performance measure except for daily and annual performance measures to show the total returns over some trading periods. *Cumulative return* is the total net realised profit (or loss) of all trades made by each strategy per one contract throughout the period. In these plots, investors can observe how the developed trading strategies compare in value over time.

1.3 Thesis organisation

The rest of the chapters is organised as follows:

Chapter 2 - *Algorithmic trading strategies.* This chapter presents contextual information on algorithmic trading in the stock market, which offers the reader a clear perspective of the complexities and surrounding influences outlined in this thesis. To position this thesis within a well-established context, information on algorithmic trading is considered and general steps for a complete algorithmic trading system are proposed. Principally, common trading strategies within the financial market are outlined as benchmark strategies, and compared with contemporary strategies that employ machine learning models. The chapter concludes with the evaluations of trading performances between non-machine learning strategies.

Chapter 3 - *Machine learning methods selected for predicting and trading asset returns.* This chapter provides an overview of preceding research specific to predicting stock price movements and developing machine learning based trading strategies, and in the application of logistic regression, support vector machine, random forest and neural networks models to financial time series. I mathematically model the problem of financial market prediction in the study, and explain why these machine learning models are provided to solve the proposed problem. Machine learning based trading strategies is proposed in chapter 4 based on the predictions made in this chapter.

Chapter 4 - *Generating trading signals with machine learning.* This chapter focuses on how machine learning methods such as logistic regression, support vector machine, random forest, and neural network can be incorporated into making an investment decision for use within stock market trading systems. It creates and follows a well defined methodology for developing trading systems which focuses on signal generation. Machine learning models are trained with technical indicators and the future price movements are predicted. One day and multi-day forecasts are then used to generate trading signals to make trading decisions. In voting integration system, the final trading signals are further confirmed by majority vote based on the predicted signals generated from single machine learning models. This chapter evaluates how well the investment decisions of each machine learning based strategy perform according to out-of-sample backtesting experiments. It verifies the robustness of machine learning based strategy with empirical results.

Chapter 5 - *Conclusion.* The major empirical results related to thesis aims are summarised and future works are presented.

Chapter 2

Algorithmic trading strategies

This chapter presents contextual information on algorithmic trading in the stock market, which offers the reader a clear perspective of the complexities and surrounding influences outlined in this thesis. To position this thesis within a well-established context, information on algorithmic trading is considered and general steps for a complete algorithmic trading system are proposed. Principally, common trading strategies within the financial market are outlined as benchmark strategies, and compared with contemporary strategies that employ machine learning models. The chapter concludes with the evaluations of trading performances between non-machine learning strategies.

2.1 Background - market environment

The stock market is a marketplace where public companies trade their shares. A company becomes public after an Initial Public Offering, which enables investors to purchase and trade their shares or stocks in the company. The shares of a company represent a partial ownership of the company, with shareholders considering that the company will be profitable and that its stocks will be valuable in the future.

There is a diverse range of investors in the stock market, each undertaking varying levels of risk. Individual investors, institutional investors - such as mutual funds, ETFs, and hedge funds - and computer trading algorithms all compete within the same market with an identical goal; gaining profit from accurately forecasting future stock prices, and following the 'Buy Low, Sell High' method to trade.

ETFs are structured as open-ended funds, and trade similarly to an equity listing on the stock exchange. This provides a convenient platform for implementation and the ability to mark-to-market in real time. With the ability to trade on various exchanges electronically, it is possible to construct automated trading systems to analyse the fluctuating market data and place orders when certain criteria are met. Trading ETFs possesses two key advantages over trading stocks. First, ETFs enable investors to gain a broader exposure to equity markets at a lower cost than numerous other forms of investment [1]. Second, ETFs are comprised of a selection of stocks that are designed to reflect how well stocks are performing overall. Explicitly, ETFs are a suitable investment option for diversification.

2.1.1 China ETFs

The dataset comprises of three China ETFs -ETF50, ETF300 and ETF500- which track the underlying Index SSE50, CSI300 and S&P China 500 Index (WCHN) respectively. While SSE50 is a stock index consisting of 50 of the largest stocks of suitable liquidity listed on the Shanghai Stock Exchange (SSE),

[1] For example, no stamp tax is applicable when trading China ETFs but trading Chinese stocks need pay stamp tax.

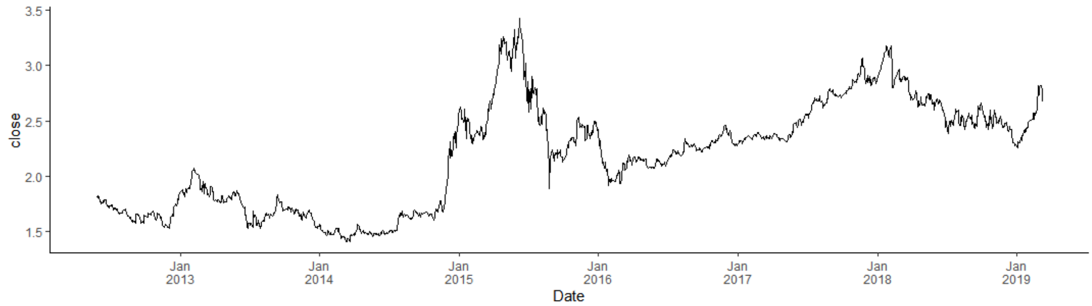
CSI300 is comprised of 300 of the largest and most liquid A-share stocks that trade on the Shanghai and Shenzhen Stock Exchange. Meanwhile, WCHN selects 500 of the largest eligible companies from the broader S&P Total China BMI Index, which characterises the entire investment environment of companies in China based on market capitalisation and trading volume. SSE50, CSI300, ETF50, ETF300, WCHN and ETF500 are supplied by the China Securities Index Company Limited[2], and are correct as of March 2019[3].

China ETFs are examined as the equity market of China is the second largest worldwide, offering potential investors innovative opportunities as it unlocks its financial markets. ETF50 and ETF300 are investigated since they are tradable and are the most renowned benchmark prices that reflect stock market performance in China. ETF50 and ETF300 are utilised to test all trading strategies to enable the performance comparability of all trading strategies. ETF500 is employed solely as a robustness check of the most profitable trading strategies to examine the strategy performance of a new ETF. The past performances of ETF50, ETF300 and ETF500 are further analysed.

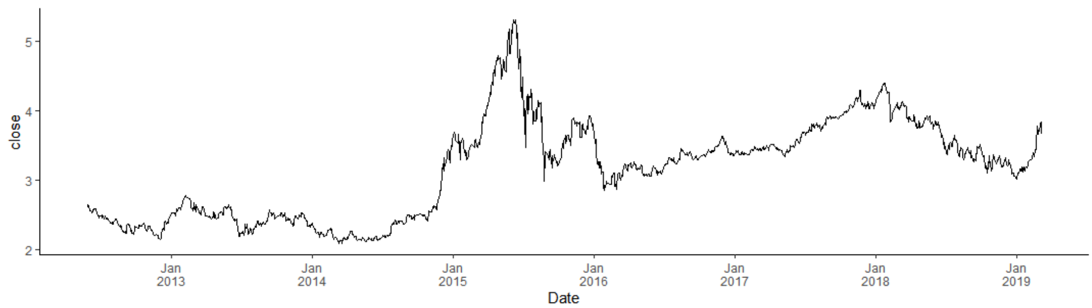
This thesis focuses on post-adjusted prices, since there is no requirement to manage fluctuations in price levels that are attributable to variations in capital structure. Returns are unaffected, but this indicates that certain prices have not existed. The adjusted close prices of ETF50, ETF300 and ETF500 during the selected period are plotted in Figure 2.1. Since the prices of ETF300 are available from May 2012, the data provided for both ETF50 and ETF300 commences from this time period to ensure that the performances of all trading strategies are comparable. As ETF500 was launched on March 2013, ETF500 has a shorter

[2]Official website: http://www.csindex.com.cn/en

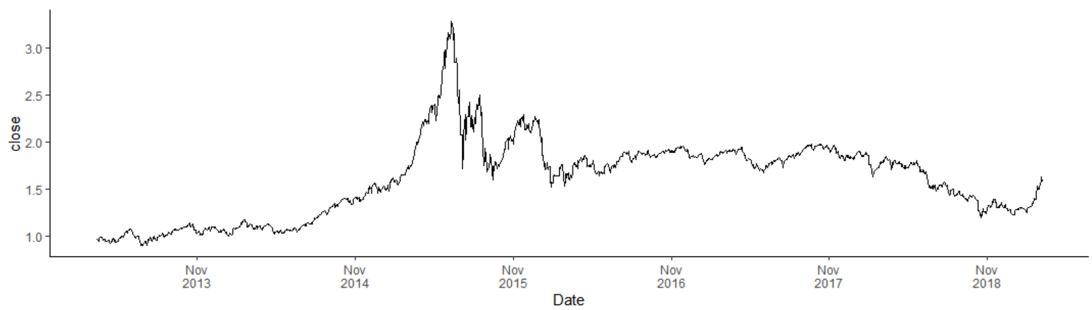
[3]It should be noted that the China ETFs are revised on a half-year basis and are updated to reflect changes in market capitalisation.



(a) ETF50 from 28 May 2012 to 8 March 2019



(b) ETF300 from 28 May 2012 to 8 March 2019



(c) ETF500 from 15 March 2013 to 8 March 2019

Figure 2.1: Closing values of ETF50 for the 1651 trading days, ETF300 for the 1651 trading days, and ETF500 for the 1456 trading days

price history and it is chosen for checking the robustness. ETF50, ETF300 and ETF500 showed a comparable market pattern in the investigated time period from March 2013. Meanwhile, there was a stable movement between 2013 and 2014. All ETFs experienced a large upward movement in June 2015, but a strong negative movement in late 2015. Following January 2016, the ETFs recovered from the loss experienced in the financial crisis, and from January 2019, a negative growth was recorded for the ETFs until June 2019.

Two test sets - *test period 1* and *test period 2* - consisting of 250 days and 500 days, respectively, are utilised in all trading strategies[4]. *Test period 1* covers the period 28th February 2018 to 8th March 2019. Throughout this period, ETF50 and ETF300 decreased by 8.98% and 8.93% respectively, which correspond to -8.83% and -10.45% at an annualised rate. The number of days with a negative return outweighs the positive by 12 for ETF50 and 24 for ETF300. *Test period 2* covers the timeframe 21st February 2017 to 8th March 2019. During this time period, the two indices of ETF50 and ETF300 increased by 22.82% and 15.72% respectively, with annualised returns of 20.12% and 15.80%. The number of days with positive returns outweighed the negative by 23 for ETF50 and 35 for ETF300.

From the trading mechanism perspectives, "T+1 trading rule"[5] is adopted for China's A-share and B-share while some China ETFs such as SSE50 ETF permit "T+0" same-day trading. Moreover, short selling mechanism for China ETFs can be realised by some possible alternatives. For example, buying inverse ETFs such as Direxion Daily China Bear 3X and ProShares Short FTSE China 50 is one alternative to do short selling.

[4] Roughly, there are 250 trading days per year. Therefore, trading performances are checked over 1 year and 2 year horizons.

[5] Under the "T+1 trading rule", stock investors cannot make settlement, payment and transfer of ownership in the same day.

2.1.2 Transaction costs

An accurate approximation of transaction costs is fundamental for a realistic evaluation of the considered trading strategies. Earlier research indicates that trading strategies that take account of transaction costs in the stock markets may not be profitable (Brock et al., 1992; Andrada-Félix and Fernández-Rodríguez, 2008). Specifically, as trading frequency increases, the effect of transaction costs outweighs the profitability of trading strategies (Bowen et al., 2010).

Numerous approaches to calculating transaction costs are outlined across various studies. Some studies focus solely on commission fees in the estimation of transaction costs (Gorgulho et al., 2011; Mabu et al., 2013). However, this is insufficient, with Narang (2009) proposing that liquidity costs should also be considered. Further research indicates that transaction costs should include the cost of commissions, trading fees, market impact and liquidity from a theoretical perspective (Hu et al., 2015b). The most common means of estimating transaction costs when dealing with stocks are the bid-ask spread and commission fees, since these two components can be explicitly observed (Pesaran and Timmermann, 1994; Korajczyk and Sadka, 2004; Demsetz, 1968).

	Long		Short		
Asset	Commission fee	Relative slippage cost	Commission fee	Relative slippage cost	Rental fee
ETF50	0.001	0.004566	0.002	0.004566	0.000317
ETF300	0.001	0.003165	0.002	0.003165	0.000317
ETF500	0.001	0.006202	0.002	0.006202	0.000317

Notes: Each year has 252 trading days.

Table 2.1: Transaction costs for each trade in all developed trading strategies.

Transaction costs in this thesis include the commission fees charged by exchanges and security firms, slippage costs based on the tick size of stocks, and the short-selling costs such as stock rental fees, which are all utilised in accordance with market practice in the stock markets in China. The corresponding values for commission fees, slippage costs and rental fees are displayed in Table 2.1. In this instance, as just one share of the ETFs is traded in each transaction, this thesis considers that one tick size approximately calculates the slippage cost in each transaction. On the Shanghai and Shenzhen stock exchange, the tick size is 0.01 RMB, and for ETF50, ETF300, and ETF500, the average prices are 2.19 RMB, 3.16 RMB and 1.61 RMB respectively. The slippage costs for ETF50 and ETF300 are approximately estimated by the tick size divided by the average prices, which are 4.566e-3, 3.165e-3 and 6.202e-3 respectively. For each transaction, the 'gap' cost is considered, as there may be a delay in taking action when a trading signal is generated. Additionally, rental fees for holding the stock in a short position are considered with a value of 8% per year based on market practice in the Chinese stock market. According to Table 2.1, all transaction costs are deducted for active trading strategies when a new position is opened at the beginning of each trade. The buy-and-hold strategy does not include any transaction costs over its holding periods.

2.2 What is algorithmic trading?

In many ways, algorithmic trading (AT) has a more sophisticated power than that of manual trading, as it analyses data, makes assumptions, acquires knowledge and provides detailed predictions on a large scale. Meanwhile, the assumptions, studies and predictions of human analysts may be limited by volume, time and cost constraints. Moreover, human investors are emotional. When confronted with continuous profits or drawdowns, humans find it challenging to overcome avaricious or fearful emotions to systematically adhere to trading disciplines. As a result, a winning trade may ultimately transform into a losing trade when human investors break the existing trading rules. In addition, AT can surmount numerous irrational behaviours such as regret and herding.

AT is an automated trade that is performed by algorithms in computers with little or no human interaction. The AT process incorporates three main steps, namely *trading signal generation*, *trading decision*, and *trade execution*. Trading signal usually provides a trigger for action, which can either be generated by humans or computers. For instance, pairs trading, which examines pairs of financial instruments that are statistically correlated based on criteria set by humans; trend following strategies, which investigate trading signals generated by man-made trading rules based on technical indicators; and chart pattern strategy, which is a similar type of trading strategy that is dependent on expert experience of pattern recognition. In these three examples, the thresholds for issuing trading signals and the threshold for stop loss protection are determined by traders. However, computers can also generate signals since it extracts information from data, and could possibly determine the inner relationship between predictors (or features) and targets, and then provide predictions.

In this research, once trading signals have been triggered, computer systems can accomplish all the tasks from trading decisions to execution. When managing trade, trading decisions comprise of several components. For instance, the performance of strategies determines trading decisions, and the trading size should be reduced during losing periods. Additionally, trading decisions are determined by the trading rules utilised in trading strategies. For example, possible trading rules for one day forecast based trading strategy indicate that if a positive, negative or neutral signal is predicted, buy, sell and hold decisions are made accordingly. Trade execution determines the trading plan not only by deciding the venue, but also the order type. Specifically, numerous assets can be traded in more than one exchange in reality, and consequently, trade on the trading venues with a high liquidity is significant to reduce transaction costs. In regard to order type, trade consisting of one large order or several small orders in the same trading strategy may affect trading profits due to transaction costs. Trade execution critically determines how to effectively execute the trade to reduce market impact and timing risk. Factors relating to exchange selection and the quantity splits of orders are not covered in this research.

Generally, trading systems consist of two processes, i.e. determining *entry strategies* and *exit strategies*. Determining entry strategies is largely dependent on the trading opportunities of specific trading strategies. For example, certain trading signals may oscillate between buy and sell signals, or may only perform effectively under strict assumptions. Thus, to prevent the risk of misusing a single signal to enter the market, multiple signals may reduce the risk of the model.

Determining exit strategies influences the time to exit after entering the trading position. Exit strategies may also be highly dependent on trading strategies under specific scenarios. For instance, traders may remain in the position as long as their assumption about the market is valid, until their assumption is proven erroneous by updated information. Traders may close the existing position if the reverse signals occur. Traders can also close the current position by implementing certain protective actions, such as stop loss order, which is an order to buy or sell an instrument once its price increases above or decreases below a pre-specified price level, to limit the loss to an investor.

Furthermore, money management, which involves determining the actual size of the trade, takes the potential risk of each trade into consideration. This means that how much money should be risked in each trade, and how much should be traded at one time, needs to be determined. As every trade possesses a potential for loss, the maximum amount of capital exposed at each trade is determined in a real trading system. In this thesis, one equity per trade is assumed, which will simplify the complexities of trading systems. In future research, money management in the trading algorithms may be considered.

While algorithm trading is powerful due to its reliable monitoring, short reaction times and data detection ability, automating the entire process from investment decisions to execution is challenging. System stability and robustness are fundamental for preventing mechanical failures. One possible view therefore believes that the less complex the system, the stronger it tends to be.

Although algorithm trading has some limitations, trading through algorithms should be encouraged from a policy perspective, as it indicates that liquidity increases with algorithmic activities (Hendershott et al., 2011). Moreover, algorithmic trading can further enhance market efficiency by narrowing spreads, reducing adverse selection and lessening trade-related price discovery. It also assists institutional market participants by decreasing transaction costs, improving entry speeds, and reducing bid-ask spread.

2.3 Trading strategies without machine learning assistance

There are numerous prevalent algorithmic trading strategies, including high frequency trading, statistical arbitrage and financial time series forecasting based trading strategies. This section presents two trading strategies, namely chart pattern and trend following strategies, along with the application of the Chinese stock markets. Both chart pattern and trend following strategies can be considered as benchmark strategies for comparing the performance of trading strategies using machine learning techniques.

2.3.1 Chart pattern trading strategy

Chart patterns are essentially price information presented in a graphical format. The technical difficulty lies in transforming the information on the k line diagram into numerical forms, such as vector and matrix, which the computer can recognise. Possible approaches of overcoming this difficulty are template-based or rule-based pattern matching (Fu et al., 2007; Sezer and Ozbayoglu, 2018). The template grid technique enables the replacement of traders to recognise anticipated patterns. In terms of algorithms, computers with an abundance of data can accomplish chart pattern recognition conveniently and efficiently.

Chartists consider that the occurrence of certain patterns can be utilised to generate profitable buy or sell signals. Head-and-shoulder, tops and bottoms, triangles, wedges, saucers and gaps, bull and bear flag are patterns that are broadly used to predict the direction of price movements for stock indices. Previous research presents a five-point chart pattern, which was originally proposed by Levy (1971). Lo et al. (2000) later employ the kernel regression approach to identify ten-point chart patterns, and examine the possibility of generating profits through charting. By identifying the perceptually significant features, pattern searching is developed (Fu et al., 2007). In previous research, both the closing price and the body of the candlesticks are investigated as data. This section focuses on the commonly used data type - close prices, while further studies on candlesticks can be obtained in Cervelló-Royo et al. (2015) and Arévalo et al. (2017).

Methodology

The entire process consists of introducing *fit function* with its two components, *Template grid* and *Image grid*, followed by several ways in which to calculate *Template grid*.

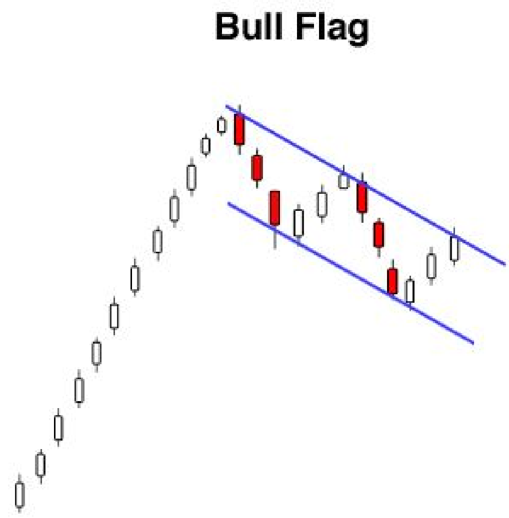


Figure 2.2: Bull flag from k-line in chart pattern strategy

.5	0	-1	-1	-1	-1	-1	-1	-1	0
1	.5	0	-.5	-1	-1	-1	-1	-.5	0
1	**1**	.5	0	-.5	-.5	-.5	-.5	0	.5
.5	**1**	**1**	.5	0	-.5	-.5	-.5	0	**1**
0	.5	**1**	**1**	.5	0	0	0	.5	**1**
0	0	.5	**1**	**1**	.5	0	0	**1**	**1**
-.5	0	0	.5	**1**	**1**	.5	.5	**1**	**1**
-.5	-1	0	0	.5	**1**	**1**	**1**	**1**	0
-1	-1	-1	-.5	0	.5	**1**	**1**	0	-2
-1	-1	-1	-1	-1	0	.5	.5	-2	-2.5

Consolidation
Breakout

Figure 2.3: Bull flag template (Leigh et al., 2002b) in chart pattern strategy

The pattern employed as an example in this section is 'bull flag'. A flag pattern is a trend continuation pattern, which forms during an uptrend, with parallel trend lines[6] above and below the price-action, which form a down slope

[6] Blue lines in Figure 2.2

(see Figure 2.2). A breakout above the price-action confirms that an uptrend is continuing. A bull flag pattern is a horizontal flag of consolidation followed by a rise in the positive direction.

The *template grid* is a weight matrix that records the price trend of graphical patterns such as head-and-shoulder, tops and bottoms, triangles, wedges, saucers and gaps, bull and bear flag. An example template is the 'bull flag template', which is exemplified in Figure 2.3. The grey grid is the matrix version of the 'bull flag' pattern. The first 7 columns indicate the consolidation process while the last 3 columns represent the sharp increase of the price (breakout). *Template grid* is adopted for calculating *fit function*, which measures how the experimental data corresponds with the anticipated bull flag designed for this research.

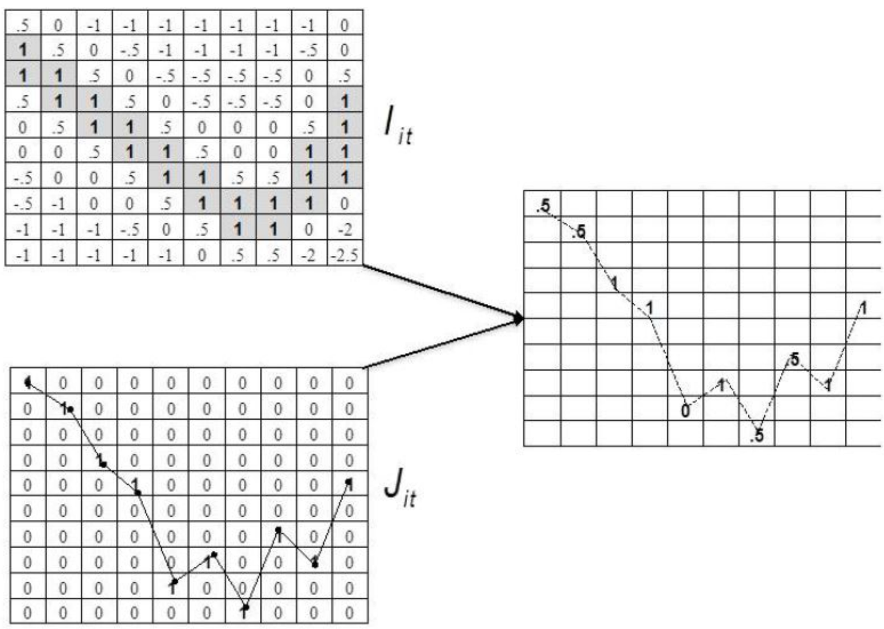


Figure 2.4: Calculation process of the fit function (Michniuk, 2017) in chart pattern strategy

The *fit function* is defined as a cross-correlation calculation between the grid of weights and the price values in a window of p periods (i.e. p=10 in Figure 2.3). This matching serves as a measurement to identify a specific pattern. An example of the *fit function* for a 10×10 grid is displayed in Figure 2.4, using the Michniuk (2017) to clarify the process. J_{it} is so called *Image grid* introduced before, which records price information in images. For J_{it}, each column ($j = 1, 2, \cdots, p$) derives from each of the p periods of the sliding window. Meanwhile, the rows - where i denotes the price rank - mark out the maximum and minimum reached by the index during p periods. Further details on how to calculate J_{it} will be explained in Algorithm 2. The *fit function* is:

$$Fit_k = \sum \sum (I_{it} * J_{it}) \tag{2.1}$$

For the estimations on *Image grid* J, it is not necessary to have a format with only one '1' and all '0's for the others in each column in the $M \times M$ template (Wang and Chan, 2009). More generally, the procedure for obtaining $J_{it}, i = 1, 2, \ldots, M, t = 1, 2, \ldots M$ is outlined below.

Suppose a window of N price values is being fitted to the template grid starting with the earliest price, with the sliding window moving daily for each of the fittings. p_t is the price value on the trading day t for the window ending on trading day k. For each trading day k, we calculate a 10 by 10 image grid, I.

The price values will relate to the rows in the grid by computing the range of N prices and scaling the range by M to get the increment (inc) value:

$$inc = (p_{max} - p_{min})/M \tag{2.2}$$

where p_{max} and p_{min} are the maximum and minimum price values found within the N values in each window separately.

After obtaining the inc, row i is allocated with an interval:

$$[p_{max} - i \times inc, p_{max} - (i-1) \times inc] \text{ for i = 1, 2, ..., M} \tag{2.3}$$

Price values for the earliest $\frac{1}{K}\%$ of the trading days are mapped to the first column of the template grid, with the most recent $\frac{1}{K}\%$ of the trading days mapped to the final columns on the right.

Next, the element value j_{it} of matrix J_{it} is determined by the portion of $\frac{N}{K}\%$ price values in each column that fall into each of the K intervals. Especially, when $N = K$, it is the same case as shown in Figure 2.3.

Finally, if the figure obtained from the data falls in grey cells, which is the ideal pattern combination, the highest score is provided. Otherwise, it will be penalised by a different weight, depending on how far it deviates from the perfect pattern.

Since the solution of pattern recognition is to estimate the *template grid*, 5 methods to achieve pattern recognition are provided.

Template 1

One means of constructing a template grid is provided by Wang and Chan (2009), who assume that the ideal pattern is defined in theory, and that the weight values are linearly decremented. They propose a simple and explicit method based on three steps:

Step1 Give the weight values '1' according to the variation of object charting pattern.

Step2 Give the weight values for the rest cells based on the distance from the cell whose weight value is '1'.

Step3 Ensure that the sum of the rows, column by column, equals 1.

An example of how to implement Template 1 is displayed in Figure 2.5, where

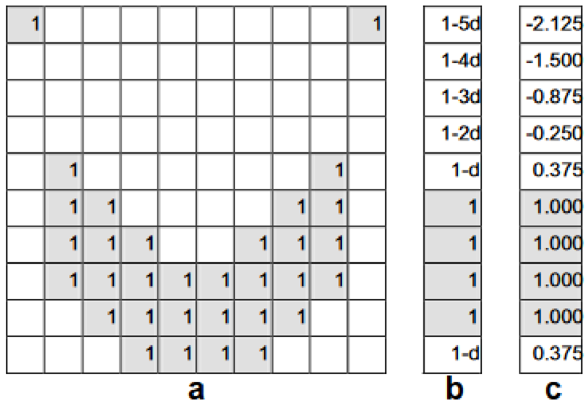


Figure 2.5: Illustration for a template formation (Wang and Chan, 2009) in chart pattern strategy

all of the weight values '1' are clearly defined by the variation of the object charting pattern. Then, take the third column as an example in 2.5 **b**. As the total sum of each column should have a weight of 1, d is determined.

Template 2

An alternate choice for template calculation is employed by inputting a three sample-period with a typical sloping flag to estimate the *template grid* (Bo et al., 2005). It appears reasonable to obtain a template based on certain sample-periods with a typical pattern. However, Bo, Linyan and Mweene do not indicate how they calculate the template from the data.

Template 3

Leigh et al. (2002b) propose another means of obtaining a template by directly utilising the template grid for the 'bull flag'. This method relates to the definition of a flag according to Downes and Goodman (2014), and further described by Duda et al. (1973)[7].

Template 4

An issue is identified with the template proposed by Leigh et al. (2002b).

[7]See Figure 2.3 for more details.

.5	0	-1	-1	-1	-1	-1	-1	-1	0
1	.5	0	-.5	-1	-1	-1	-1	-.5	0
1	**1**	.5	0	-.5	-.5	-.5	-.5	0	.5
.5	**1**	**1**	.5	0	-.5	-.5	-.5	0	**1**
0	.5	**1**	**1**	.5	0	0	0	.5	**1**
0	0	.5	**1**	**1**	.5	0	0	**1**	**1**
-.5	0	0	.5	**1**	**1**	.5	.5	**1**	**1**
-.5	-1	0	0	.5	**1**	**1**	**1**	**1**	0
-1	-1	-1	-.5	0	.5	**1**	**1**	0	-2
-1	-1	-1	-1	-1	0	.5	.5	-2	-2.5

(a)

.5	0	-1	-1	-1	-1	-1	-1	-1	0
1	.5	0	-.5	-1	-1	-1	-1	-.5	0
1	**1**	.5	0	-.5	-.5	-.5	-.5	0	.5
.5	**1**	**1**	.5	0	-.5	-.5	-.5	0	**1**
0	.5	**1**	**1**	.5	0	0	0	.5	**1**
0	0	.5	**1**	**1**	.5	0	0	**1**	**1**
-.5	0	0	.5	**1**	**1**	.5	.5	**1**	**1**
-.5	-1	0	0	.5	**1**	**1**	**1**	**1**	0
-1	-1	-1	-.5	0	.5	**1**	**1**	0	-2
-1	-1	-1	-1	-1	0	.5	.5	-2	-2.5

(b)

Figure 2.6: Two examples of bull flag templates in chart pattern strategy

According to Figure 2.6, the grey cells in 2.6(a) indicate a bull flag pattern with a fitting value of 6.5, while 2.6 (b) also displays a score of 6.5 despite not being considered a bull flag pattern. Therefore, in certain instances, the template proposed by Leigh et al. (2002b) cannot distinguish an accurate bull flag pattern.

Due to the instability of Leigh's template, Wang (2007) defines a template that is more accurate in identifying price increases, as displayed in Figure 2.7. Rather than employing a consolidation and breakout version like Leigh, Figure 2.7 presents a possible break and consolidation version, where the first 5 columns confirm an upward wave band, followed by 4 columns of horizontal consolidation and a final column for an upward-tilting breakout.

-.25	-.4	-.45	-.7	-1.5	-1.6	-1.6	-1.6	-1.6	-.7
-.25	-.4	-.45	-.6	-.75	-1.4	-1.4	-1.4	-.8	1
-.25	-.4	-.45	-.55	-.5	-.75	-.75	-.5	-.5	.4
-.25	-.4	-.45	-.55	-.25	.9	.9	.9	-.15	-.35
-.25	-.5	-.6	-.25	.9	1	1	1	1	-.55
-.3	-.6	-.25	.8	1	.9	.9	.9	.8	-.45
-.35	.1	.8	1	.65	.6	.6	.4	.75	-.15
.1	.8	1	.5	.3	.5	.5	.3	0	.1
.8	1	.5	.35	.15	0	0	0	.3	.35
1	.8	.35	0	0	0	0	.1	.25	.3

Figure 2.7: Bull flag template (Wang, 2007) in chart pattern strategy

Template 5

0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	-1	-1	-1	-1	-1	-1
0	**0**	**0**	-1	-2	-2	-2	-2	-2	-2
0	**0**	-1	-3	-3	-3	-3	-3	-3	-3
0	-1	-3	-5	-5	-5	-5	-5	-5	-5
0	-1	-5	-5	-5	-5	-5	-5	-5	-5
0	-1	-5	-5	-5	-5	-5	-5	-5	-5
5	-1	-5	-5	-5	-5	-5	-5	-5	-5

Figure 2.8: Bull flag template (Cervelló-Royo et al., 2015) in chart pattern strategy

Cervelló-Royo et al. (2015) later propose a template that utilises a dissimilar breakout and consolidation approach, as shown in Figure 2.8. The only positive weight in all cells is located at the lowest corner on the left. All price paths in I with a positive fitness score must pass through this starting point (Cervelló-Royo et al., 2015). According to this method, the cells labeled with a negative value indicate that a punishment will be provided to the candidate that sought to match the bull flag template.

Application to up-trend trading strategies by chart patterns on China ETFs

In this section, the chart pattern trading strategy is back tested to examine whether the 'bull template' pattern can be recognised and predicted using template grids. The idea of the chart pattern strategy is to utilise the sliding window to locate at what time point the bull template pattern appears. Subsequently, the buy signal is generated at this time point. The criteria for determining the occurrence of the 'bull template' -time point - can be obtained from the concepts with certain assumptions, or from the previous time series of prices in the periods that bull template patterns occurred. Once the trading position is opened, stop loss rules should be established to reduce price downward risks.

Specifically, the following 3 steps are expected to be applied to the trading strategy in this research, and these steps are developed by recognising patterns from the empirical data, establishing trading rules, and analysing model performances with the transaction costs. The procedure for establishing the template grid follows with the work of Chen and Chen (2016), as displayed in Algorithm 1. They adopt the weighted averages methods to generate the weight ($w_{i,t}$) of each cell in the template (I_{it}). The main idea is to obtain Template I from the historical close prices, which covers the period that the anticipated patterns appeared. However, it is difficult to consult with experts to recognise the bull flag pattern in this research. Thus, the same bull flag template is adopted according to Bo et al. (2005) as their data also arises from the Chinese stock indices.

The recognised pattern triggers the trading signals through a threshold. To

Step 1: $P = p_1, p_2, \cdots, p_k$ is a set of daily closing index containing the pattern, where k is the fitting window size.
Step 2: Rank the index value in set P at a decrement.
Step 3: Calculate template I_{it} for trading day $i(1 \leq i \leq k)$, and t is $1 \leq t \leq M$.
$I_{i,t} = 1$ if $Rank(p_t)$ falls in the i-th interval.
$I_{i,t} = 0$ otherwise.
Step 4: Repeat Step 3 until the last I_{it} of possible pattern is calculated.
Step 5: Calculate $w_{i,t}$ in template grids for all n patterns.
$w_{i,t} = \frac{\sum_{b=1}^{b=n} Pattern_b(I_{i,t})}{n}$

Algorithm 1: Template (I) building for pattern recognition

determine this threshold, some studies are reviewed. Wang (2007); Zapranis and Tsinaslanidis (2012) specify how to acquire thresholds and holding periods. According to some studies, they support the idea that trading thresholds can be obtained by the empirical data of the previous trading days (Bo et al., 2005; Chen and Chen, 2016; Leigh et al., 2002a). The trading rule is implemented by taking long positions in assets on the day, which a buy signal is triggered. After entering a trading position, another threshold needs to be determined is 'stop loss', e.g. Cervelló-Royo et al. (2015); Arévalo et al. (2017) employ stop loss and take profit in their studies.

The investigated chart pattern strategy coheres with the previous work, and is presented in Algorithm 2. The input data is the historical close prices for the investigated asset. During the initial $Wind$ days in the test set, the *ImgGrid* is generated daily based on the updated sliding window with training data from i to $i + wind$. During the *Wind* days, the values for the earliest $(P_{Max} - P_{Min})/T$ % of the trading days[8] are mapped to the first column of the grid (ImgGrid). The values for the subsequent earliest $(P_{Max} - P_{Min})/T$ % trading days are mapped to the second column of the grid, with the most recent $(P_{Max} - P_{Min})/T$ % trading

[8] P_{Max} and P_{Min} represent the maximum and minimum prices over the T days in the rolling window.

```
procedure MatchTemplateSearch(data, TemGrid, hold, Wind, M,
Threshold, StopLoss)
Input: data, TemGrid, Wind, M, Threshold, hold
Output: CumsumReturn
for i in test set do
    PRICE = data[i : i + Wind];
    pMax = max(price);
    pMin = max(price);
    Calculate inc;
    for a = 1, 2, ··· , M do
        for b = 1, 2, ··· , M do
            p = price[(Wind/M) b : (Wind/M)(b + 1)];
            ImgGrid[a,b] = len([pp for pp in p if pp falls in a-th
              interval])/len(p)
        end
        FitScore = sum(sum(ImgGrid ∗ TemGrid))
    end
        S=0
    if FitScore ≥ Threshold then
        S = 1 ;                                      // Open "Long"
        P_buy = price[i + wind +1];
        Start = i+wind+1;
        j = i + wind + 2
        while p[j] > StopLoss do
            j = j + 1;
            if j==i + wind + hold then
                break
            end
            P_sell = p[i + wind + hold];
            End = i+wind+Hold;
            n = End - Start ;                 // Number of holding days
            S = 0 ;                                 // Close "Long"
        end
        P_sell = p[j]
        End = j;
        n = End - Start;
        S = 0 ;                                     // Close "Long"
    end
    Return = (ln(P_sell/P_buy)- cost)/n ;  // Daily return per trade
    CumsumReturn = CumsumReturn + Return
end
return CumsumReturn
```

Algorithm 2: Backtest the chart pattern strategy days mapped to the farthest column on the right. In line with the study Bo et al. (2005), the weight matrix *TempGrid* has been given directly, and *ImgGrid* and *TempGrid* generate the fitted value for a daily stock pattern, together.

When the *FitScore* is above the *Threshold*, a buy signal occurs. The *Threshold* is defined as 70% of the maximum previous *FitScore*. *StopLoss* is closely related to the buying price at the time when the long position was opened. Once a buy signal occurs, the stock will be sold on the *hold* day unless the threshold (*StopLoss*) is touched prior to the *hold* day. That is to say, as soon as the price reaches the *StopLoss*, the position will be automatically closed at the current market price, preventing further risk to the capital. In this example, *StopLoss* is 0.97 $\times$ the buying price, which indicates that the investor only affords a 3% loss at most in each trade. If the stock price does not touch the stop loss protection before *hold* day, the long position has to be closed on the specific *hold* day. Then investors need to wait for a new buy signal to occur. The profit for each transaction is calculated based on log returns ($ln(P\_sell/P\_buy)$). The daily return is the profit for each transaction per *hold* day, which is ($ln(P\_sell/P\_buy)/hold$).

The passive trading strategy, buy-and-hold, is considered a comparison to the chart pattern strategy. The buy-and-hold strategy is a simple strategy in which investors purchase stocks on the first day of the holding period, and hold the stock until it is sold at the end of the holding period. The current daily return[9] of buy-and-hold is the current log return calculated from the current price and the future price.

The empirical results are presented from the perspective of daily returns and annual returns analysis. Although the most commonly employed statistics relating to the chart pattern strategy in previous research are based on the return of

[9] $Return_{BH}(t) = ln(P(t+1)/P(t))$

	Before transaction costs		After transaction costs			
	ETF50	ETF300	ETF50		ETF300	
	L	L	L	BH	L	BH
Panel A: 250 trading day daily returns when Wind = 120, hold=25						
Average return (e-3)	-1.5850	-1.3810	-1.7910	-0.3505	-1.5160	-0.4148
Standard deviation	0.0054	0.0048	0.0060	0.0146	0.0053	0.0147
t-Statistic	-4.6790	-4.5290	-4.7400	-0.3807	-4.5580	-0.4458
Minimum	-0.0474	-0.0472	-0.0530	-0.0486	-0.0514	-0.0528
Median	0	0	0	-0.1751e-3	0	-1.2489e-3
Maximum	0.0008	0.0020	0.0006	0.0729	0.0018	0.0682
Skewness	5.2620	-5.5490	-5.1560	0.0729	-5.5100	0.3210
Kurtosis	20.3800	14.2800	18.6700	5.7300	13.3100	15.6100
Panel B: 500 trading day daily returns when Wind = 120, hold=25						
Average return (e-3)	-0.3966	-0.3354	-0.5940	0.2211	-0.4707	1.0134
Standard deviation	0.0039	0.0037	0.0043	0.0121	0.0040	0.0120
t-Statistic	-2.2220	-2.0090	-3.0270	0.4091	-2.6000	0.1962
Minimum	-0.0473	-0.0472	-0.0529	-0.0486	-0.0513	-0.0528
Median	0	0	0	0.4195e-3	0	0.5345e-3
Maximum	0.0042	0.0036	0.0039	0.0729	0.0034	0.0682
Skewness	-6.7250	-7.4050	-6.8370	0.0517	-7.5630	0.0414
Kurtosis	19.0700	13.9400	17.6700	3.9690	12.8300	4.2840

Notes: 'L', 'LS', and 'BH' stand for 'long' strategy, 'long and short' strategy, and 'buy-and-hold' strategy.
The test set sizes for Panel A and Panel B are 250 days and 500 days, individually.
t test is one side test to check whether positive returns are significant.
Significant estimate for t test at 0.1% (***), 1% (**) 5% (*) and 10% (+) levels are marked correspondingly.

Table 2.2: Daily trading profits of chart pattern strategy.

each trade, all the results are finally shown in a daily basis to compare with other trading strategies.

Table 2.2 displays the daily return results for two possible test periods, which cover 250 and 500 trading days individually. The results show that all daily returns of chart pattern strategies are negative. Table 2.3 illustrates the annual return results of the same instance as Table 2.2. The charting strategy appears unprofitable, since the Sharpe and Sortino ratios after transaction costs are generally negative.

As the daily return of chart pattern strategy is the average log returns in each

	Before transaction costs		After transaction costs			
	ETF50	ETF300	ETF50		ETF300	
	L	L	L	BH	L	BH
Panel A: 250 trading day annual returns when Wind = 120, hold=25						
Annual return	-0.3993	-0.3480	-0.4513	-0.0883	-0.3820	-0.1045
Excess return	-0.4168	-0.3655	-0.4688	-0.1058	-0.3995	-0.1220
Standard deviation	0.0850	0.0765	0.0948	0.1919	0.0834	0.1897
Downside deviation	0.0885	0.0795	0.0988	0.1619	0.0867	0.1613
Sharpe ratio	-4.9040	-4.7760	-4.9430	-0.4579	-4.7860	-0.5225
Sortino ratio	-4.7110	-4.6010	-4.7440	-0.6535	-4.6080	-0.7462
Maximum drawdown	0.4450	0.4390	0.4960	0.2120	0.4780	0.2120
Panel B: 500 trading day annual returns when Wind = 120, hold=25						
Annual return	-0.0999	-0.0845	-0.1496	0.0557	-0.1186	0.0264
Excess return	-0.1174	-0.1020	-0.1672	0.0382	-0.1361	0.0089
Standard deviation	0.9854	0.9375	0.9756	0.2550	0.9273	0.2595
Downside deviation	0.0611	0.0578	0.0684	0.1352	0.0634	0.1340
Sharpe ratio	-1.8540	-1.7220	-2.4000	0.1992	-2.1180	0.0470
Sortino ratio	-1.9200	-1.7650	0.2827	-2.4440	0.0666	-2.1470
Maximum drawdown	0.5150	0.5080	0.5690	0.2895	0.5480	0.2895

Notes: 'L', 'LS', and 'BH' stand for 'long' strategy, 'long and short' strategy, and 'buy-and-hold' strategy.
The test set sizes for Panel A and Panel B are 250 days and 500 days, individually.
Table 2.3 presents the annualised return converted from Table 2.2 with the assumption that one year contains 252 trading days.

Table 2.3: Annualised trading profits of chart pattern strategy.

trade while the daily return of buy-and-hold is the daily log returns for the stock price, it is difficult to compare the total return of chart pattern strategies with that of buy-and-hold strategy in a graph. Therefore, the cumulative return plots are not provided for chart pattern strategies.

To summarise, the risk management tool (stop loss) is utilised in the investigative chart pattern trading strategies. In this scenario, stop loss is pre-determined by the extent of the loss that can be tolerated by investors to safeguard investors against an extensive loss in all opened positions. Although the chart pattern trading algorithm enables investors to accurately capture the target pattern and take efficient action, solid experience and knowledge of how to identify pattern forms is required. Algorithms solely replace humans in monitoring price changes, and take action if a pattern is acknowledged. The success of chart pattern strategy is largely dependent on whether the pattern proposed by experts can truly generate a buy or sell signal. Considering each of the aforementioned characteristics, chart pattern trading strategies are not extended in this thesis.

2.3.2 Trend following strategy

Literature review

Trend following strategies are widely accepted by traders who seek to avoid making predictions when designing trading strategies. Traders employ technical indicators to track historical trends, and react to changes of the trends by observing buy and sell signals from previous market movements. Thus, trend following automates the buying or selling process according to the position of the price relative to a long-term moving average value, and holds the position until it closes when the signals change. In the short-term, trend following does not guarantee profits for every trade, but in the long-term, a positive cumulative return may be achieved since the positive returns can suppress the negative returns.

Some successful trend following strategies are outlined. The trend following strategy, which aims to determine the appropriate entry and exit time, is conducted by monitoring the trend in the upward direction and the opposite trend after a 'pivotal' is observed (Fong et al., 2011). Later, a valid trend following step is employed to catch the minor fluctuations after the trend following, as the trend following may work by leveling out the averages of the time series. It appears that the improved trend following model with trend recalling is more effective than other trend following methods (Fong et al., 2012). Upon verification of this rule to generate positive profits, an evolutionary trend following strategy is identified in Hu et al. (2015a).

Different types of trend following trading strategies are summarised. Fong and Yong (2005) apply 840 moving average rules to a sample of 30 leading Internet stocks during 1998-2002. Narayan et al. (2015) examine whether momentum-based trading strategies that buy past winners and sell past losers work in the commodity futures market. Szakmary et al. (2010) seek to whether momentum strategies, dual moving average crossover strategies and channel strategies yield positive mean excess returns net of transaction costs in at least 22 of the 28 markets. Han et al. (2016) unify in a single framework the three major price patterns: the short-term reversal effects, the momentum effects, and the long-term reversal effects. Moreover, trend following strategies can be combined with other strategies to develop profitable trading strategies. Fuertes et al. (2010) illustrate that the double-sort strategy that exploits both momentum and term structure signals outperforms the single-sort strategies using momentum and term structure signals separately.

The most straightforward trend following strategies focus on moving averages. Moving average rules are mechanical trading rules that endeavour to capture trends. The price crossover rule (MACD trend following strategy) generates the buy and sell decisions solely from historical close prices. The theory of the MACD trend following strategy is that if the actual price on a set day is above the moving average, the strategy provides a long signal. If the actual price is below the moving average, the signal is short. As recent prices are lower than earlier prices, the price is determined to be in a downtrend and a sell signal is produced. The response following a buy or sell signal is to buy or sell. Moving average trading is profitable if the variation in price level between buy and sell signals is sufficient to cover the costs. The actual number of average days was selected to provide the most sufficient information ratio for the stock.

Application of MACD trend following trading strategy on China ETFs

Two steps have been applied to the trend following strategy in this research; namely algorithm demonstrations, and model performance analyses with transaction costs.

The MACD is calculated as the difference between a short-term period of 12 days and a long-term period of 26 days exponential moving average (EMA). The actual number of average days was determined based on the research of Fernández-Blanco et al. (2008); Eric et al. (2009); Beyaz et al. (2018). The 'signal line' is obtained by the 9-day EMA of the MACD.

Figure 2.9: Sliding window process on the 1-day horizon

The dynamic training process is defined in Figure 2.9, consisting of a training period $Wind$ as an initial window to generate $MACD$ and $SignalLine$ in Algorithms 3 and 4. To verify the one-side active trading strategy ('L') and the two-sided active trading strategy ('LS'), this research will proceed as follows. On the first trading day, trading decisions are determined based on trading strategies. For the 'L' strategy, a long position is taken if the MACD ($MACD$ in Algorithms 3 and 4) crosses above the signal line ($SingalLine$). A long position means purchasing a stock for later resale, while a short position means selling the stock now and purchasing it later. A long position is maintained if the MACD is above the

```
Input: data_close, Wind, cost
Output: TotalReturn_L
Data: Close price
Calculate MACD and SignalLine
Initial State S = 0
for i in test set do
    if S = 0 AND MACD(i) > SignalLine(i) AND
    MACD(i − 1) < SignalLine(i − 1) then
        S=1 ;                                   // Open "Long"
        Return = ln(P(i+1)/ P(i)) -cost
    if S = 1 AND MACD(i) > SignalLine(i) then
        Return = ln(P(i+1)/ P(i)) - cost ;      // Keep "Long"
    else
        S = 0
        Return = 0 ;                            // Close "Long"
    end
end
TotalReturn_LS = cumsum(Return)
```

Algorithm 3: Backtest MACD Strategy ('L') in test set

signal line. In the 'LS' strategy, a buy (or sell) signal is generated when the price shifts above (or below) the longer moving average. Following the first trading day, the training window shifts forward by one day, and the trading decision on the second day is considered immediately following the updated training window. This process is repeated until the data in the training window is exhausted.

The main difference between Algorithms 3 and 4 is that there is one state ("Long") for Algorithm 3 while Algorithm 4 considers two states ("Long" and "Short"). For Algorithms 3, three options can be chosen- enter, stay or exit the state "Long". However, only the periods of open and hold the "Long" position are recorded. For Algorithm 4, investors not only can enter, stay and exit either "Long" or "Short", but also transfer from one state to the other state. During the periods of open and hold the "Long" or "Short" position, return are calculated.

Tables 2.4 and 2.5 present the daily and annual returns of trend following

```
Input: data_close, Wind, cost
Output: TotalReturn_LS
Data: Close price

Calculate MACD and SignalLine

Initial State S = 0
for i in test set do
    if S = 0 AND MACD(i) > SignalLine(i) AND
    MACD(i − 1) < SignalLine(i − 1) then
        S=1 ;                                          // Open "Long"
        Price_Buy = ln(P(i + 1)/P(i)) − cost
        Return = 0
    if S = 0 AND MACD(i) < SignalLine(i) AND
    MACD(i − 1) > SignalLine(i − 1) then
        S=-1 ;                                         // Open "Short"
        Price_Sell = ln(P(i)/P(i + 1)) − cost
    if S = 1 AND MACD(i) > SignalLine(i) then
        Return = ln(P(i+1)/P(i)) ;                     // Keep "Long"
    if S = −1 AND MACD(i) < SignalLine(i) then
        Return = ln(P(i)/ P(i+1)) ;                    // Keep "Short"
    else
        S = 0
        Return = 0 ;                                   // Wait
    end
end

TotalReturn_LS = cumsum(Return)
```

Algorithm 4: Backtest MACD Strategy ('LS') in test set

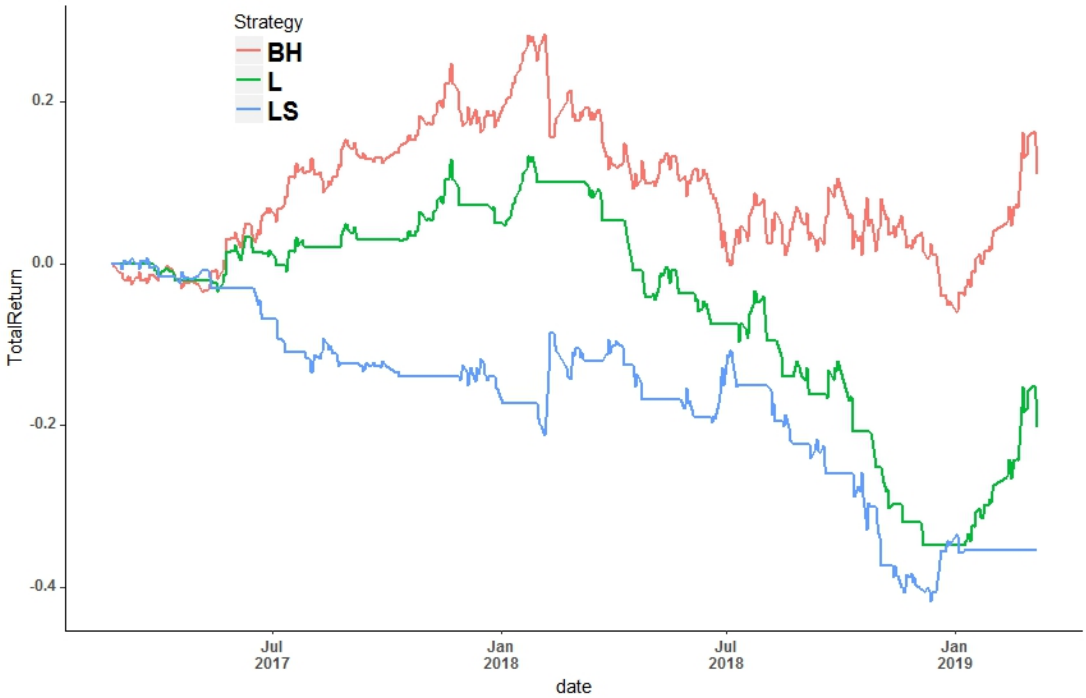
	Before transaction costs				After transaction costs					
	ETF50		ETF300		ETF50			ETF300		
	L	LS	L	LS	L	LS	BH	L	LS	BH
Panel A: 250 trading day daily returns when Wind = 1401										
Average return(e-3)	-0.9303	-0.9303	-1.2209	-1.2209	-0.6260	-0.6260	-0.3505	-0.8435	-0.8435	-0.4148
Standard deviation	0.0111	0.0111	0.0113	0.0113	0.0114	0.0114	0.0146	0.0115	0.0115	0.0147
t-Statistic	-1.3150	-1.3150	-1.7030	-1.7030	-0.8640	-0.8640	-0.3807	-1.1580	-1.1580	-0.4458
Minimum	-0.0485	-0.0485	-0.0485	-0.0485	-0.0527	-0.0527	-0.0486	-0.0527	-0.0527	-0.0528
Median	0	0	0	0	0	0	-0.1751e-3	0	0	-1.2489e-3
Maximum	0.0729	0.0729	0.0729	0.0729	0.0682	0.0682	0.0729	0.0682	0.0682	0.0682
Skewness	0.4431	0.4431	0.4392	0.4392	0.3112	0.3112	0.0729	0.2682	0.2682	0.3210
Kurtosis	12.8800	12.8800	12.3600	12.3600	11.4300	11.4300	5.7300	11.0200	11.0200	5.6100
Panel B: 500 trading day daily returns when Wind = 1151										
Average return(e-3)	-0.1617	-0.4060	-0.1758	-0.3100	-0.4071	-0.7094	0.2211	-0.3762	-0.5736	1.0134
Standard deviation	0.0089	0.0081	0.0088	0.0080	0.0090	0.0083	0.0121	0.0089	0.0081	0.0120
t-Statistic	-0.4040	-1.1151	-0.4444	-0.8610	-1.0075	-1.8928	0.4091	-0.9429	-1.5693	0.1962
Minimum	-0.0485	-0.0424	-0.0527	-0.0514	-0.0485	-0.0424	-0.0486	-0.0527	-0.0514	-0.0528
Median	0	0	0	0	0	0	0.4195e-3	0	0	0.5345e-3
Maximum	0.0729	0.0473	0.0682	0.0472	0.0729	0.0473	0.0729	0.0682	0.0472	0.0682
Skewness	0.3128	0.1407	0.1678	0.0339	0.2933	0.0601	0.0517	0.1011	0.0115	0.0414
Kurtosis	13.3070	7.3180	13.5060	8.5550	12.8190	6.6150	3.9690	12.8000	7.9980	4.2840

Notes: 'L', 'LS', and 'BH' stand for 'long' strategy, 'long and short' strategy, and 'buy-and-hold' strategy.
The test set sizes for Panel A and Panel B are 250 days and 500 days, individually.
t test is one side test to check whether positive returns are significant.
Significant estimate for t test at 0.1% (***), 1% (**) 5% (*) and 10% (+) levels are marked correspondingly.

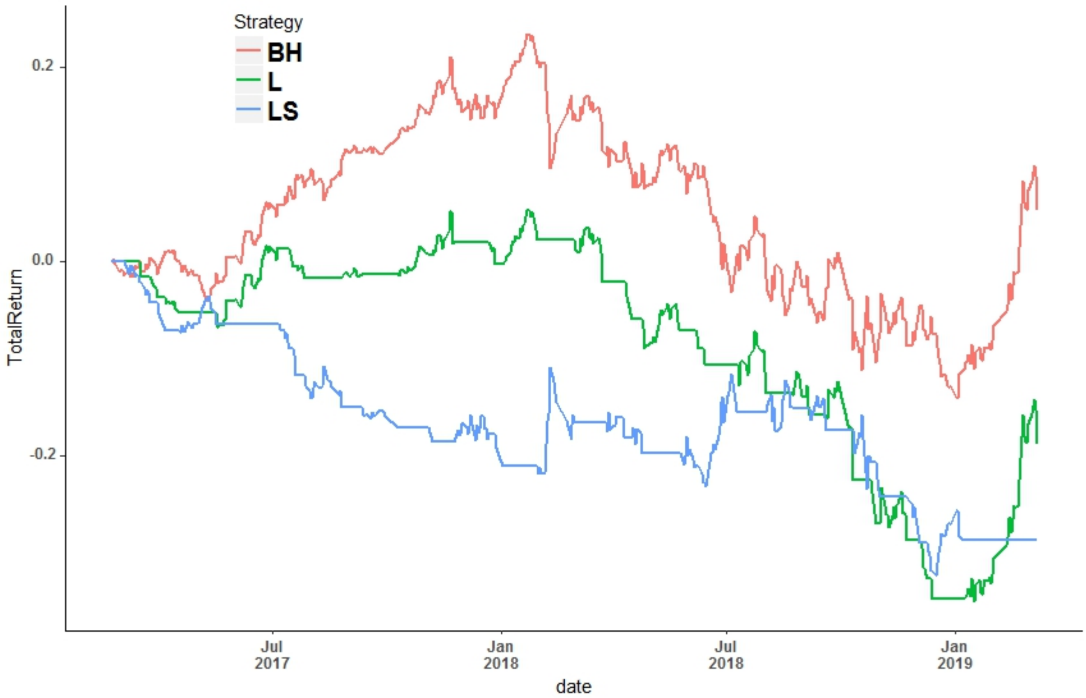
Table 2.4: Daily trading profits of trend following strategy.

strategies. When the testing set is established by the last 250 days' data, there is no difference between the 'L' and 'LS' trading strategies. However, when the testing set is reset to include 500 days' data, the trading profits of the 'L' and 'LS' trading strategies are negative in all cases, with no trend following strategy capable of outperforming the buy-and-hold strategy.

As the concern of most investors is the recent trading performance, only the cumulative returns of the chart pattern over the most recent 500 trading days are plotted for comparison purposes. The results of the trend following strategy in Figure 2.10 reveal that 'L' strategies outperform 'LS' strategies in all cases for both ETF50 and ETF300. Nevertheless, no positive cumulative returns are generated after transaction costs.



(a) ETF50 in 500 days



(b) ETF300 in 500 days

Figure 2.10: Cumulative returns (after transaction costs) for MACD trend following strategy in 500 trading days.

	Before transaction costs				After transaction costs					
	ETF50		ETF300		ETF50			ETF300		
	L	LS	L	LS	L	LS	BH	L	LS	BH
Panel A: 250 trading day annual returns when Wind = 1401										
Annual return	-0.2344	-0.2344	-0.3077	-0.3077	-0.1578	-0.1578	-0.0883	-0.2126	-0.2126	-0.1045
Excess return	-0.2519	-0.2519	-0.3252	-0.3252	-0.1753	-0.1753	-0.1058	-0.2301	-0.2301	-0.1220
Standard deviation	0.1772	0.1772	0.1796	0.1796	0.1815	0.1815	0.2311	0.1825	0.1825	0.2335
Downside deviation	0.1325	0.1325	0.1367	0.1367	0.1314	0.1314	0.1619	0.1345	0.1345	0.1635
Sharpe ratio	-1.4220	-1.4220	-1.8109	-1.8109	-0.9657	-0.9657	-0.4579	-1.2603	-1.2603	-0.5225
Sortino ratio	-1.9010	-1.9010	-2.3780	-2.3780	-1.3340	-1.3340	-0.6535	-1.7100	-1.7100	-0.7462
Maximum drawdown	0.5082	0.4828	0.4337	0.4986	0.4817	0.4739	0.2120	0.6125	0.5413	0.2120
Panel B: 500 trading day annual returns when Wind = 1151										
Annual return	-0.0407	-0.1023	-0.0443	-0.0781	-0.1025	-0.1787	0.0557	-0.0947	-0.1445	0.0264
Excess return	-0.0582	-0.1198	-0.0618	-0.0956	-0.1200	-0.1962	0.0382	-0.1122	-0.1620	0.0089
Standard deviation	0.1419	0.1291	0.1403	0.1277	0.1433	0.1329	0.1919	0.1415	0.1296	0.1897
Downside deviation	0.1024	0.0942	0.1009	0.0923	0.1056	0.1003	0.1352	0.1040	0.0965	0.1340
Sharpe ratio	-0.4105	-0.9280	-0.4406	-0.7489	-0.8381	-1.4768	0.1992	-0.7938	-1.2502	0.0470
Sortino ratio	-0.5685	-1.2711	-0.6123	-1.0350	-1.1368	-1.9556	0.2827	-1.0788	-1.6790	0.0666
Maximum drawdown	0.4022	0.4258	0.4377	0.4386	0.4831	0.4129	0.2895	0.5192	0.4561	0.2895

Notes: 'L', 'LS', and 'BH' stand for 'long' strategy, 'long and short' strategy, and 'buy-and-hold' strategy.
The test set sizes for Panel A and Panel B are 250 days and 500 days, individually.
Table 2.5 presents the annualised return converted from Table 2.4 with the assumption that one year contains 252 trading days.

Table 2.5: Annualised trading profits of trend following strategy.

2.4 Summary

This chapter has verified whether non-machine learning trading strategies yield significant positive excess returns when applied to two ETFs in China. Both the chart pattern and trend following trading strategies cannot generate continuous positive returns.

It is worth noting that the chart pattern strategy cannot generate positive returns with other possible cases. For example, when the holding period changes from 25 to 40 days, 40-day forecast horizon performs worse than 25 forecast horizon. Moreover, the use of other parameter combinations to develop the MACD trend following strategy cannot improve the current results.

To summerise, the chart pattern strategy cannot generate a positive risk-adjusted return. Neither Sharpe ratios nor Sortino ratios are above 1 after transaction costs. This indicates that the return is not be high enough to compensate for the risk. As previously mentioned, the success of the chart pattern is heavily dependent on experts' experience. However, algorithm trading for chart pattern strategies is a robust tool that can replace traders in monitoring trade and taking actions based on trading rules. Therefore, the chart pattern remains feasible for experts who have profitable trading strategies. MACD trend following cannot generate a positive Sharpe ratio after transaction costs for both ETF50 and ETF300. Thus, other technical indicators should perhaps be considered in the future when designing trend following strategies.

Chapter 3

Machine learning methods selected for predicting and trading asset returns

This chapter provides an overview of preceding research specific to predicting stock price movements and developing machine learning based trading strategies, and in the application of logistic regression, support vector machine, random forest and neural networks models to financial time series. I mathematically model the problem of financial market prediction in the study, and explain why these machine learning models are provided to solve the proposed problem. Machine learning based trading strategies is proposed in chapter 4 based on the predictions made in this chapter.

3.1 Literature review

This chapter reviews an abundance of existing literature pertaining to the field of trading strategies with machine learning models. Previous studies succeeded in establishing machine learning models that generate signals. One method of signal generation is to apply piecewise linear representation to discover turning points. In such research, the buy and sell signals generated from turning points are then denoted as the target of the proposed machine learning models (Chang et al., 2011; Luo and Chen, 2013; Luo et al., 2017; Chang et al., 2008). Possible trading signals may be generated by periodic occurrences. For instance, Booth et al. (2014) solely assess the weighted random forest ensemble when trading over seasonal events rather than daily trading. Further research models focus on the transaction probabilities of uptrend/downtrend to downtrend/uptrend to generate profitable trading strategies (Ładyżyński et al., 2013). Chavarnakul and Enke (2008) determine that neural networks can provide more appropriate trading signals based on two technical indicators, namely volume adjusted moving average and ease of movement. Meanwhile, Chang et al. (2009) identify that when training artificial neural networks, buy and sell signals are generated according to the experience of experts, but not the stock prices for each input. A recent study generates trading signals by news based data, and trains the model through random forest (Feuerriegel and Prendinger, 2016).

Numerous attempts have been made to devise a consistently profitable autonomous trading system assisted by single machine learning models, such as LSTM (Troiano et al., 2018), SVM (Choudhury et al., 2014; Chen and Hao, 2018) and neural network (Chenoweth et al., 2017; Chen and Leung, 2004). Multi-layer perceptron (MLP) is perceived to be capable of approximating arbitrary functions since artificial neural network is a highly elaborate analytical technique for modeling complex nonlinear functions (Principe et al., 2000).A considerable volume of work has already been published on the application of neural networks on stock market trading. For instance, Dunis and Williams (2002) conclude that neural network regression models can forecast EUR/USD returns. Moreover, according to empirical evidence from the Madrid Stock Market, in the absence of trading costs the technical trading rule is always superior to the buy-and-hold strategy for 'bear' and 'stable' market episodes (Fernandez-Rodrıguez et al., 2000). Chen et al. (2003) reveal that probabilistic neural network based investment strategies obtain higher returns than other investment strategies on the Taiwan Stock Index.

With exception to single machine learning models, numerous studies have been conducted on ensemble learning across many fields, including credit risk forecasts and price predictions on the oil market, the stock futures market, the gold market and the electricity market. A summary of the most commonly employed ensemble methods is provided.

Previous literature indicates that the simple combination of neural networks outperforms individual networks when trading S&P 500 future contracts (Trippi and DeSieno, 1992). Moreover, Paleologo et al. (2010) compare the subagging decision tree with SVM when contending with credit scoring problems. Yu (2008) perform crude oil forecasting through a three-step forecasting system called the EMD-based neural network ensemble learning paradigm. The first step employs empirical mode decomposition (EMD) to separate the crude oil price into certain intrinsic mode functions (IMFs). Then, each IMF is directed into a three-layer feed-forward neural network. Finally, an adaptive linear neural network totals the prediction outcomes. Tang et al. (2018) implement a similar system through decomposition by EMD, prediction through random vector functional link network, and ensemble by linear addition. Yu et al. (2008) establishes a more complex system, the multistage neural network ensemble learning approach, which consists of 6 stages. Zhao et al. (2017) first utilise bootstrapping to replicate the original sample, thus generating several sub-samples. Then, stacked denoising autoencoders are adopted to train, test and predict all sub-samples. Finally, the average of the sub-sample prediction results is selected to obtain a final prediction. Meanwhile, Xian et al. (2016) devise the Ensemble Empirical Model Decomposition and Independent Component Analysis method for gold price analysis, while Neupane et al. (2017) employ two types of ensemble model for comparison, namely the fixed weight method and varying weight method.

This chapter primarily focuses on literature concerned with price prediction on the stock market through ensemble learning approaches. The local weighted polynomial regression formulates a sophisticated ensemble model for exposing stock index (Chen et al., 2007). It has been determined that the multiple classifier is superior to the single classifier in terms of prediction accuracy for stock returns (Tsai et al., 2011). The 3-stage nonlinear ensemble model, proposed by Xiao et al. (2014), includes training through three neural network based models, optimisation by improved particle swarm optimisation, and prediction by SVM. The 3-layer expert trading system comprises of random forest prediction, ensemble of ensembles, and signal filtering alongside risk management (Booth et al., 2014). To predict stock prices, Ballings et al. (2015) establish an innovative idea to generate imbalanced data based on certain thresholds of annual stock price returns. Consequently, the results were unresponsive to the selected threshold. The research of Ballings et al. (2015) indicate that exponential smoothing is fundamental for predicting stock price movements through random forest.

To the best of our knowledge, no previous research develops trading strategies based on forecasts of price movements across multiple days. Thus, through the utilisation of one-day forecasts and multi-day forecasts generated by machine learning models, this research verifies and analyses the performance of machine learning techniques in making trading decisions, and draws comparison with several benchmark strategies including the buy-and-hold strategy, the time series strategy (AR model), the chart pattern strategy, and the trend following strategy.

3.2 Model setups for financial market prediction

Financial market prediction problem can be expressed as the attempt to explore the relationship between an output y and a set of D inputs $\mathbf{x}$ where $\mathbf{x} = \left\{x^1, x^2, ..., x^D\right\}$, i.e. y = F(x)[1]. If y represents the total of upward price movements in n days in the future, the function F could be learnt from in-sample training data so that when new unseen (out-of-sample) data is presented, a new prediction can be made. Both binary classification where y $\in \{0, 1\}$ and multi-class classification where y $\in \{0, 1, ..., n\}$, i.e. y denotes the total number of upward prices (p) across the n days, [2] are investigated in the next chapter. $\mathbf{x}$ could be composed of exogenous variables. However, $\mathbf{x}$ in this research is L lags of x so that:

$$y_{t+1} = F(x_t) \tag{3.1}$$

[1] F can be any machine learning models.
[2] Especially, n=1 for binary classification

where $\mathbf{x}_t = \left\{x_t^1, ..., x_t^D, x_{t-1}^1, ..., x_{t-1}^D, ..., x_{t-L}^1, ..., x_{t-L}^D\right\}$, and y_{t+1} is obtained by considering p_i from t+1 until t+n, $p_i \in \{0, 1\}$ [3].

The following sections in this chapter describe all machine learning models, including logistic regression, the support vector machine, artificial neural network, random forest and the majority vote employed for prediction.

3.3 Logistic Regression

Logistic regression is the first machine learning model to be investigated as it is the simplest and maybe the most effective classifier when dealing with classification problems. The advantage of logistic regression is that it can give the probability of a binary class by producing a simple probabilistic formula for the classification. Moreover, it does not require strict assumptions like normally distributed input variables. A linear relationship between the input and output variables is not required as well. Logistic regression is based on two assumptions: (1) it requires the dependent variable to be binary, with the groups being discrete, non-overlapping, and identifiable; and (2) it considers the cost of type I and type II error rates in the selection of the optimal cut-off probability (Ohlson, 1980).

Logistic regression, developed by statistician Cox (1958), is a classification algorithm utilised to assign observations to a discrete set of classes. Ng (2000) provides a basic structure of the logistic regression model (Appendix A.1). The optimisation solver of logistic regression that employed in this thesis is the limited-memory BFGS (LBFGS) algorithm as LBFGS can be applied straightforwardly to high-frequency data. Mathematically speaking, LBFGS never explicitly forms or stores the Hessian matrix, which can be rather expensive when the number of dimensions is increased (Appendix A.2). LBFGS has been selected because

[3] $y_{t+1} = p_{t+1}$ for binary classification

for high frequency data, and time efficiency is just as significant as prediction accuracy. To implement high frequency trading, predicting stock movement immediately and accurately enables the opportunity to defeat competitors.

3.4 Support Vector Machine

Support Vector Machine (SVM) is the second classifier to be utilised since it can produce a binary classifier - optimal separating hyperplanes - by an extremely non-linear mapping of the input vectors into a high-dimensional feature space (Pai et al., 2011). The idea of SVM is to transform the data that is non-linear separable in its original space to a higher dimensional space in which it can be separated by a simple hyperplane.

SVM, proposed by Vapnik (1998), is a means of generating a classification hyper-plane that separates two classes of data with a maximum margin, considering that aspect of the error is tolerated. In SVM, a linear model is adopted to estimate a decision function using non-linear class boundaries based on support vectors. If the data can be linearly separated, SVM trains linear machines for an optimal hyperplane that separates the data without error and into the maximum distance between the hyperplane and the closest training points. The training points that are closest to the optimal separating hyperplane are called support vectors (Kim and Sohn, 2010). Mathematically, more details about the basic structure of SVM can be found in Appendix A.2.

There are some advantages of SVM: (1) there are purely two free parameters to be chosen, namely, the upper bound and the kernel parameter; (2) the solution of SVM is unique, optimal, and global since the training of an SVM is conducted by solving a linearly constrained quadratic problem; (3) SVM is based on the structural risk minimization principle, therefore, this type of classifier minimizes the upper bound of the actual risk, whereas other classifiers minimize the empirical risk (Shin et al., 2005).

3.5 Artificial Neural Network

Multi-layer perceptron (MLP) is selected as the third classifier since it perceived to be capable of approximating arbitrary functions. Artificial neural network is a highly elaborate analytical technique for modeling complex nonlinear functions (Principe et al., 2000).

procedure MLPCLASSIFIER(X)
Initialization
  Initialize all weights to small random values
Training
while *learning* **do**
  for *each input vector* **do**
    Forwards Phase
      Compute the activation of each neuron in the hidden layer(s)
      Work through the network and calculate the output activations
    Backwards Phase
      Compute the error at the output
      Compute the error in the hidden layers
      Update the output layer weights
      Update the hidden layer weights
  end
  Randomize the order of the input vectors
end
Recall
Use the Forwards Phase described above

Algorithm 5: MLP Algorithm

With regard to MLP algorithm [4], the process of feed-forward neural network with back propagation is shown in Algorithm 5 (Rumelhart et al., 1985). Back-

[4]The basic structure of MLP from mathmatical perspective can be found in Appendix A.3.

propagation is a form of gradient decent where the gradient of the errors are calculated with respect to the weights of the network so that the weights are adjusted and the error function is minimised. As this differentiation cannot be done directly, the chain rule of differentiation is utilized. This produces an update function for each layer, which is applied backwards through the network.

3.6 Ensemble Learning

The machine learning algorithms previously discussed contain some parameters that require optimisation. As the 'No Free Lunch' theorem states, there is no single learning algorithm that consistently generates the most accurate learner in any domain (Wolpert, 1996). Moreover, each learning algorithm is based on a certain model that derives from a set of assumptions. This inductive bias results in an error if the assumptions do not hold for future data. Consequently, these issues increase motivation to explore models that are comprised of multiple base learners to solve the problem. The theory proposes that by suitably combining multiple base learners, overall accuracy will be improved. This general approach is termed the ensemble method, which has certain advantages when confronted with statistical, computational and representational problems (Ballings et al., 2015). Commonly used ensemble learning models such as decision tree, random forest, and majority vote are introduced in the next section.

3.6.1 Decision Tree

The principle of decision tree is to recursively divide data into subsets, until the state of the target predictable attribute in each subset is homogeneous. The decision tree classification technique consists of two phases, namely tree building and tree pruning. Tree building is a top-down process, which aims to recursively partition the tree until all data items are assembled into the same class label. Meanwhile, tree pruning is a bottom-up process. Thus, by reducing over-fitting, prediction and classification accuracy will be improved (Imandoust and Bolandraftar, 2014).

The decision tree commonly imitates the human mindset, deeming it straightforward to comprehend data and provide accurate interpretations. Decision trees even provide the logic for the data to interpret, unlike black box algorithms such as SVM and MLP.

3.6.2 Random Forest

The overall objective of random forest is to combine the predictions of multiple binary decision trees to establish more accurate predictions than individual models. One advantage of the random forest algorithm is its relative robustness when confronted with noisy data. According to Patel et al. (2015), the utilisation of sub-sampling and random decision trees often generates sounder predictive results than single decision trees. Moreover, random forest contains methods for balancing errors in unbalanced data sets of class population. The random forest algorithm minimises overall prediction errors by maintaining low error rates on larger classes whilst permitting high error rates on smaller classes (Breiman, 1999).

Bagging (see Algorithm 6) is applied when the goal is to reduce the variance of the decision tree. The objective is to establish several subsets of data from the training sample selected at random with replacement. The subset of data is employed to train the decision trees, and consequently, an ensemble of diverse models is produced. The average of all predictions from the various trees is

for m = *1 to M* **do**
    Bootstrap sample D_m of size N with replacement from the original training set D with equal weight
    Train a model $G_m(x)$ to the bootstrap sample D_m
end
for m = *1 to M* **do**
    Employ $G_m(x)$ to the testing set D_T
end
Classifier using $I(\sum_{i=1}^{i=M} G_i(x_i)/M > threshold) \in class1$

Algorithm 6: Bagging Algorithm

procedure RandomForestClassifier(X) [5]
$Forest$ = new Array[ ]
for i = *0 to B* **do**
    D_i = Bagging(D)
    T_i = new DecisionTree( )
    $features_i$ = RandomFeatureSelection(D_i)
    T_i.train(D_i,$features_i$)
    $forest$.add(T_i)
end
return $forest$

Algorithm 7: Random Forest Classifier Algorithm utilised to establish a more robust estimator than the single decision tree.

Alternatively, the random forest algorithm is an extension to bagging. The random forest algorithm not only considers a random subset of data, but also a random selection of features to develop the trees. Random forest is an ensemble that consists of multiple classification or regression trees, which produces a higher accuracy than any base classifier (Booth et al., 2014).

In the random forest algorithm(see Algorithm 7), B samples are first bootstrapped from the original dataset (Liaw et al., 2002). Then, for each bootstrap sample, every node is divided using the most effective predictor from the randomly selected predictors. Meanwhile, in the decision tree, each node is divided using the most appropriate split among all of the predictors. New data is then forecast by aggregating the predictions of the B number of trees, such as the majority votes for classification and the average for regression.

Finally, a training dataset score is obtained through an out-of-bag (OOB) estimate. An estimate of the error rate is calculated based on the following training sample (Liaw et al., 2002):

- At each bootstrap iteration, predict the data that is not in the bootstrap sample ('out-of-bag' (OOB) data) using the tree grown with the bootstrap sample.

- Aggregate the OOB predictions, calculate the error rate, and call it the OOB estimate of the error rate.

3.6.3 Voting-based ensemble classifier

The majority vote model (Vote) is the most straightforward means of combining multiple classifiers. The majority vote model is established through combining certain single classifiers such as logistic regression, support vector machine and random forest. In this study, the majority vote model is an ensemble of ensemble. Furthermore, in this instance, all base learners are granted equal weight to establish a final decision. The combination rule for the Vote strategy is that the information from the base classifiers includes the binary decisions in each instance. The majority vote produces the final decision from the common results of the base classifiers.

All machine learning models introduced in this chapter will be applied to predict the trading signals of the machine learning based trading strategies in the next chapter.

Chapter 4

Generating trading signals with machine learning

This chapter focuses on how machine learning methods such as logistic regression, support vector machine, random forest, and neural network can be incorporated into making an investment decision for use within stock market trading systems. It creates and follows a well defined methodology for developing trading systems which focuses on signal generation. Machine learning models are trained with technical indicators and the future price movements are predicted. One day and multi-day forecasts are then used to generate trading signals to make trading decisions. In voting integration system, the final trading signals are further confirmed by majority vote based on the predicted signals generated from single machine learning models. This chapter evaluates how well the investment decisions of each machine learning based strategy perform according to out-of-sample backtesting experiments. It verifies the robustness of machine learning based strategy with empirical results.

4.1 Introduction

Studies with regard to Chinese stock market anomalies are reviewed in this section. The presence of stock market anomalies defeats the basic premises of the Efficient Market Hypothesis (EMH). The definition of an efficient market is a market in which all available information is fully reflected by the price. Fama (1995) defines three levels of efficiency, namely weak form, semi-strong form and strong form.

- In a weak form efficient market, the past price movements, volume and earnings data do not affect a stock's price and can't be used to predict its future direction.

- In a semi-strong form efficient market, security prices have factored in publicly-available market and that price changes to new equilibrium levels are reflections of that information.

- In a strong form efficient market, all information in a market, whether public or private, is accounted for in a stock's price.

The classical evidence of equity market anomalies is calendar anomalies, such as weekend, day of the week, and January effects. The following research show calendar anomalies in the Chinese stock market. Luo et al. (2009) show that the day-of-the-week effects and monthly effects exist in the Chinese stock markets based on the evidence from Shanghai and Shenzhen A-shares Closing Index and Shanghai and Shenzhen B-shares Closing Value-Weighted Index. Cai et al. (2006) verify that the day-of-the-week effect in the Chinese market for both A-share and B-share indexes remains significant. Chukwuogor et al. (2006) conclude that the day-of-the-week effect exists in the Shanghai Stock Exchange (SSE), supported by Kruskal-Willis test and basic statistics for daily returns. More recently, Lu

Author (year)	Stock market	Test period	Machine learning methods	Transaction cost	Positive returns
Andrada-Félix and Fernández-Rodríguez (2008)	New York Stock Exchange Composite Index	1993 - 2002	Boosting	0.2% per stock	Yes
Booth et al. (2014)	Deutsche Borse Ag German Stock Index	2000 - 2010	Random forest	\$0.003 per stock	Yes
Chang et al. (2011)	3 US stocks and 3 Taiwan stocks	2008 - 2009	Neural network	\$40 per transaction	Yes
Chang et al. (2009)	700 Taiwan stocks	2005 - 2006	Neural network	No	Yes
Chavarnakul and Enke (2008)	$S\&P$ 500 Index	1998 - 2003	Neural network	No	Yes
Chen et al. (2003)	Taiwan Stock Index	1982 - 1992	Neural network	3 cases are 0.03%, 3%, and 6%	All cases are profitable
Chen and Hao (2018)	Shanghai and Shenzhen stock markets	2012 - 2014	SVM	No	Yes
Chenoweth et al. (2017)	$S\&P$ 500 Index	1982 - 1993	Neural network	break even cost is 0.35%	Yes
Fernandez-Rodrıguez et al. (2000)	Madrid Stock Index	1991 - 1997	Neural network	No	Yes
Feuerriegel and Prendinger (2016)	Deutsche Borse Ag German Stock Index	2014 - 2011	Natural language processing	0.1%-0.3%	Yes
Leigh et al. (2002b)	New York Stock Exchange Composite Index	1981 - 1996	Neural network	No	Yes
Luo and Chen (2013)	20 stocks in Shanghai stock market	2010 - 2011	SVM	0.35%	Some stocks are profitable
Sezer and Ozbayoglu (2018)	Dow 30 stocks and ETFs	2002 - 2017	Convolutional neural network	No	Most stocks are profitable

Table 4.1: Literature for machine learning based trading strategies reviewed in this thesis

et al. (2016) point out that holiday effects are significant in both Shanghai and Shenzhen Security Exchanges according to Wilcoxon Rank-Sum test. Monday anomalies are prominent in Shenzhen Component Index by calendar effect performance ratio with three sample interval cases of 500 days, 1000 days and 1500 days (Zhang et al., 2017). Xiong et al. (2018) prove the existence of four types of calendar effect - reversed Monday effect, January effect, TOTM effect,and CLNY effect- in SSE50, SSE180[1], CSI300, and ChiNext index[2]. Casalin (2018) lists that some empirical results for data including Shanghai and Shenzhen index, and suggests that the holiday effects are positive, significant, time-varying, with no signs of decline over time. To summarise, several studies covering Chinese stock indices anomalies are analysed. All studies obtained positive results supporting Chinese stock market calendar anomalies. It proves the existence of market inefficiencies in Chinese stock markets.

Lo (2004) propose Adaptive Markets Hypothesis (ADH) that assumes that evolutionary market dynamics, such as cycles, trend, and market inefficiency, are

[1]SSE180 consists of 180 stock with some of the large scale, good fluidity, and strong industry representative stock listed on the SSE.

[2]ChiNext index can comprehensively and objectively reflect the overall price movements of the Growth Enterprises Market stocks.

capable of trigger arbitrage opportunities. The studies in section 3.1 related to machine learning based strategies which tested for the stock market are discussed to verify whether technical indicators and calendar anomalies can be trained to generate profitable trading strategies. Other papers that apply machine learning in stock trading are not re-summerised as they purely provide performance measurements, such as 'Accuracy' for classification and 'Mean Square Error' for regression problem. Therefore, 13 papers are selected from section 3.1 to be presented in Table 4.1. The evidence from 13 papers supports that machine learning based trading strategies are profitable among different countries. This indicates that there may be huge investment opportunities to be explored by machine learning models in stock markets.

4.2 Problem statement

This section deals with the problem of verifying whether machine learning models can take trading opportunities in the Chinese stock market to make profitable investment decisions based on price predictions. The tasks for machine learning models are predicting one day and several day's future price movements instead of one day and several day's future values. Learning tasks, therefore, are forecasting upward prices in a single day and over several days, and these two tasks are solved by binary and multi-class classification models, separately.

For the first task related to one day forecast, the next day's stock price change is classified into two types: upward price and downward price.

With regard to multi-day (n) price movement problem, for each day, we still assume that the stock price either goes up or down. In this case, the target is classified into m classes[3] based on how many upward price movements are

[3] $n + 1$ cases can be obtained during the next n days. In other words, $m = n + 1$.

observed during the next n days. The second task deals with the multi-day forecasts with the aim of improving the risk-adjusted returns by reducing the number of transactions.

4.3 Data pre-processing and data description

The targets in the two tasks are described in detail. Daily close prices of ETF50, ETF300, and ETF500 from 28 May 2012 to 8 March 2019 are selected for experimental evaluation. Experiments are based on 7 years of historical data for these 3 ETFs. In more detail, the dataset for ETF50, ETF300 and ETF500 consists of 1650, 1650, and 1455 observations, 809, 811 and 777 of which belong to 'up', and the remaining 841, 839, and 678 belong to 'down'. Each observation has 7 features[4] and 1 class label. The class label is either '1' or '0' for the binary classification problem where '1' indicates an uptrend and '0' indicates price downfall. The class labels for the second task (multi-class classification problem) are '0', '1', '2', '3', and '4', in which '0' indicates no upward price movements during the next four days, '1' implies there is one upward price movement during four days from today, ..., and '4' denotes all price movements are upward in the four days, respectively. Under this setting, class '4', '3', '2', '1' and '0' imply a strong buy signal, a weak buy signal, a neutral signal, a weak sell signal and a strong sell signal, respectively.

The data is divided into three sets: training, cross validation and test sets. We set the length of the in-sample training window to three possible values: 1100 days, 1000 days and 800 days. In order to determine the best architecture of machine learning models, the subsequent 300-600 daily data is utilized to optimise the best parameters before making out-of-sample predictions on the cor-

[4]Refer to section 4.4

	ETF50							ETF300						
	One day		Multiple days					One day		Multiple days				
Day	1	0	4	3	2	1	0	1	0	4	3	2	1	0
250	119	131	8	61	100	65	16	113	137	6	57	95	70	22
500	255	245	31	134	191	116	28	255	245	31	133	193	114	29

Table 4.2: Comparison of classification observation for test sets in machine learning based trading strategies with one-day forecast and multi-day forecasts.

responding test sets. To provide robust results of model performance, three cross validation periods with 300, 400, and 600 days are checked in Table 4.4. Based on assessments of performance in the cross validation period, the specification of random forest type is selected for the use of out-of-sample testing. In Tables 4.7 to 4.13, the trading performances of the selected model are examined with two investment horizons of 250 days and 500 days.

Table 4.2 illustrates the number of observations for test sets in each class for both one day prediction and multiple days prediction problems. Balanced data can be observed in the binary classification setting for one day prediction. However, the observations per class in the multiple days prediction are imbalanced: more than a third of the data belongs to class '2' (38% to 40%) whereas class '3' or '1' has 22.8% to 28% data. Only 2.4% to 8.8% data belongs to '4' or '0'.

4.4 Features

Features for each observation are generated from the stock price information (open price, close price, low price, high price, and volume), which means that technical indicators are utilised as features.

Some scholars support that technical indicators can be employed to generate positive trading strategies (Kwon and Kish, 2002; Nam et al., 2005; Chong and Ng, 2008; Metghalchi et al., 2008; Marshall and Cahan, 2005; Conrad and Kaul,

Technical indicator	Description	Calculation
RSI	Relative Strength Index	100-100/(1+$\frac{SMA_t(p_n^{up},n1)}{SMA_t(p_n^{dn},n1)}$)
STCK	Fast Slow Stochastic Oscillator	$\%K = \frac{p^c - min(p_n^l)}{max(p_n^h) - min(p_n^l)} * 100$
STCD	Slow Stochastic Oscillator	$\%D = SMA(\%K, N)$
WILLS	Williams % R	$W\%R = \frac{p^c - HIGH_n}{HIGH_n - LOW_n} * 100$
MACD	Moving Average Convergence Divergence	$EMA_t(p^c, s) - EMA_t(p^c, f)$
ROCP	Rate of Change Percentage	$\frac{EMA_t(p^h - p^l, n)}{EMA_{t-n1}(p^h - p^l, n)} - 1$
OBV	On Balance Volume	$OBV_t = OBV_{t-1} \pm VOL_t$

Table 4.3: Description and calculation of feature in machine learning based trading strategies with one-day forecast and multi-day forecasts.

1998). Some papers report the mixed results of technical trading strategies over different trading periods (Schulmeister, 2009, 2008). Other scholars argue that technical trading rules are unprofitable after data snooping bias or transaction costs are taken into account (Marshall et al., 2008; Ready, 2002). The rest of my thesis intends to examine whether some technical indicators is capable of developing profitable machine learning based trading strategies.

Sullivan et al. (1999) divide trading rules into five families, namely, filter rules, moving averages, support and resistance rules, channel breakouts, and on-balance volume averages. In this thesis, some commonly used technical indicators such as relative strength index, stochastic line, williams, moving average convergence divergence, rate of change percentage, and on balance volume (see Table 4.3) are selected from the five categories of indicators.

Relative strength index (RSI) is a strength indicator computed as rise rate over fall rate in a time period[5]. It is usually used to identify an overbought or oversold state in the trading of an asset. *RSI* ranges from 0 to 100. If the *RSI* is below level 30, the asset is oversold; when *RSI* is above 70, the asset is overbought. Large surges and drops in the price of an asset will influence the *RSI* by creating false buy or sell signals. Therefore, the *RSI* is best used with other tools together to determine the trading signals (Wilder, 1978).

Stochastic line (STCK, STCD) is composed of two smooth moving average lines ($\%K$ and $\%D$) to judge the buy/sell points. The fast stochastic K represents a percent measure of the last close price related to the highest of the high price values (highest high) and the lowest of the low price values (lowest low) of the last n periods. The theory behind the *stochastic line* is that the price is likely to close near its high (or low) in an upward-trending (or downward-trending) market. Transaction signals occur when the $STCK$ crosses the $STCD$ (Basak et al., 2019).

William (WILLS) indicates stock overbuy or oversell. It can be used to discover the peak or trough of the stock price. If the difference between the lowest stock prices and close prices is large (or small), and the stock is overvalued (or undervalued), traders need wait for a short (or long) trading signal to sell (or buy) the stock (Chang et al., 2009). *WILLS* therefore is adopted to assist the turning point decision through examining stock is overvalued or undervalued.

Moving average convergence divergence (MACD) is to find the fluctuation of stock price via exponential moving average of the last n observations of series p^c[6]. When *MACD* is above 0, the short-term average is above the long-term average, which signals upward momentum. The opposite is true when the

[5] $SMA_t(p_n^{up}, n1)$ and $SMA_t(p_n^{dn}, n1)$ are the simple moving averages of the last n1 observations of p's up and down.

[6] $EMA(p^c, n) = \lambda \sum_{s=0}^{\infty} (1-\lambda)^s p_{t-s}^c$, where $\lambda = \frac{2}{n+1}$

MACD is below 0. The zero line (when the *MACD* is 0) usually acts as an area of support and resistance for the *MACD* indicator. A typical trading strategy using *MACD* is *MACD crossover*. When the *MACD* falls below the signal line, it is a bearish signal which indicates that it is the time to sell. On the contrary, when the *MACD* rises above the signal line, the indicator gives a bullish signal, which suggests that the price of the asset tends to experience an upward momentum (Appel, 2005).

Rate of change percentage (ROCP) of a time series p_t^c evaluates the breadth of the range between high and low prices. *ROCP* can be used to confirm price moves or detect divergences (Larson, 2012).

On balance volume (OBV) is using positive and negative volume flow to predict the change of stock price. It adds the volume when the close has increased and subtracts it when the close has decreased (Granville, 1976).

Due to the huge value difference among 7 features, pre-processing process i.e. normalisation (or standardisation) [7] is necessary to feed logistic regression, SVM, and MLP with a same range for each input.

4.5 Training, cross validation and test set

Cross-validation set is separated from test set as it plays an essential role in robustness check for model accuracy with different model parameters prior to trade on the out-of-sample test sets. In the cross-validation set, the optimal combination of parameters is determined before out-of-sample tests.

[7] Each feature is standardised independently. The standardised features have zero mean and unit variance.

4.5.1 K-fold cross-validation vs. sliding window cross-validation process

In most situations, the k-fold cross-validation or leave one out cross validation is widely used when the training and testing data are independent. Due to time series data have non-random time orders, k-fold cross-validation by random process may mix up past, present and future. Instead, all models are trained via a rolling horizon approach to make out-of-sample forecasts in cross-validation set. This sliding window process updates training set for every out-of-sample predictions and incorporates the latest observed information into the model. After one out-of-sample forecast is made, the training set moves forward for one day and the same training process is repeated.

4.5.2 Parameter optimisation in cross-validation phase

All cross-validation data is used for checking the out-of-sample prediction accuracy of machine learning models. The model accuracy[8] defined by the following equation:

$$\text{Accuracy} \; = 100 \times \frac{\text{Number of success (i.e. predicted equals real)}}{\text{Sample size}}, \tag{4.1}$$

All machine learning models mentioned in chapter 3 are examined by *Accuracy* in 3 cross-validation sets. As MLP is a heavily parametrised model, and the number of hidden nodes selected controls the complexity of the model, we use a simple 2 hidden layers with 3 nodes in the first layer and 2 nodes in the second layer to consider the basic version of MLP.

As an example to show the empirical results during the test set of 250 days,

[8]Specifically, the predicted classes are 2 and n+1 for binary classification and multi-classification (n), respectively.

	ETF50						ETF300					
	RF		DT	SVM	LR	MLP	RF		DT	SVM	LR	MLP
Wind, CV	N=5	N=10				nn(3,2)	N=5	N=10				nn(3,2)
(801, 600)	63.67	64.33	61.50	66.83	67.17	67.17	63.33	65.50	62.83	65.00	66.00	66.00
(1001, 400)	64.25	68.00	65.25	68.50	70.00	70.00	67.00	68.50	65.50	66.00	66.00	66.00
(1101, 300)	66.00	71.00	65.00	69.67	70.67	70.67	63.67	64.67	62.33	67.67	67.00	66.00

Notes: 'RF', 'DT' and 'LR' represent 'random forest', 'decision tree' and 'logistic regression', respectively.
N is the number of decision trees in the random forest.
$Wind$ is the length of sliding window (in days). CV is the size of cross validation set.
LR uses limited-memory bfgs (LBFGS) to implement regularized logistic regression.
The parameters in SVM are obtained by radial basis function kernel (RBF).
nn(3,2) is the neural nodes with 2 hidden layers, which is composed of 3 nodes in the first hidden layer and 2 nodes in the second hidden layer.

Table 4.4: Up/down prediction accuracy (%) in the cross validation set of machine learning based trading strategies with one-day ahead forecasts.

Table 4.4 tabulates the results of optimal estimations for all machine learning models in all 3 cases, in which training window size is 801 days, 1001 days and 1101 days, respectively. SVM, logistic regression and random forest have a better performance in all cross-validation sets, all with an accuracy of greater than 61.50%. Random forest classifier performs better than decision tree classifier considering all cases including different training window sizes and different stock indices. It means that a single tree is not accurate enough in determining the future stock price movement. Similarly, if we compare the performance of MLP and logistic regression with respect to both stock indices across different training window sizes, MLP does not perform as well as logistic regression in all cases. As a result, decision tree and MLP are not used in predicting one day price movement. As the *Accuracy* for random forest improves as $Wind$ increases, this indicates that a larger training window size often contributes to the prediction accuracy. Furthermore, random forest always results in a higher accuracy for $N = 10$ than for $N = 5$. Therefore, $N = 10$ is the pre-determined optimal parameter when we use random forest to predict one day forecast.

4.6 Trading strategies

This section presents the overall process of formulating the machine learning based trading strategies by algorithms and formulas. Three types of machine learning based trading strategies, namely one day prediction based trading strategies with single machine learning classifier, one day prediction based trading strategies with majority vote and multi-day forecast based trading strategies with machine learning are introduced. The corresponding trading rules for these three trading strategies are explained.

4.6.1 Trading using the one-day ahead prediction with machine learning

The overall process of formulating the one day ahead prediction based trading strategies with machine learning is shown in this section.

```
Input: data, Wind, cost
Output: TotalReturn_L
Initial State S = 0
for i in test set do
    da(i) = data[i-Wind,i]
    y_predict(i) = AICLASSIFIER(da(i))
    if S = 0 AND y_predict(i) is TRUE then
        S=1 ;                                   // Open new long position
        Return = ln(P(i+1)/P(i))-cost
    if S = 1 AND y_predict(i) is TRUE then
        Return = ln(P(i+1)/ P(i)) ;        // Hold the existing position
    else
        S = 0
        Return = 0 ;                            // Close the long position
    end
end
TotalReturn_L = cumsum(Return)
```

Algorithm 8: Back testing for trading using single day machine learning prediction ('L')

```
Input: data, Wind, cost
Output: TotalReturn_LS
Initial State S = 0
for i in test set do
    da(i) = data[i-Wind,i]
    y_predict(i) = AICLASSIFIER(da(i))
    if S = 0 AND y_predict(i) is TRUE then
        S=1 ;                                   // Open a long position
        Return = ln(P(i+1)/P(i))-cost
    if S = 0 AND y_predict(i) is FALSE then
        S= -1 ;                                 // Open a short position
        Return = ln(P(i)/P(i+1))-cost
    if S = 1 AND y_predict(i) is TRUE then
        Return = ln(P(i+1)/ P(i)) ;             // Hold the long position
    if S = −1 AND y_predict(i) is FALSE then
        Return = ln(P(i)/ P(i+1)) ;             // Hold the short position
    else
        S = 0
        Return = 0 ;                            // Close the existing position
    end
end
TotalReturn_LS = cumsum(Return)
```

Algorithm 9: Back testing for trading using single day machine learning prediction ('LS')

The operational details of the machine learning based trading strategies are described. At the beginning of each trading period, the investor makes an asset allocation decision of whether to invest in the stock market or deposit it in the bank. The alternative to stock market investment is regarded as an opportunity cost. Hence, excess return is calculated by subtracting the deposit rate from the actual log return of trading strategies. If investors decide to trade in the stock market, there are two choices: buy-and-hold or active trading decision. Buy-and-hold is investing money in the stock fund and taking no action until the end of the test periods. Active trading strategies developed in this section include single machine learning based trading strategies, voting integration strategy and time series based strategy.

Algorithms for single machine learning based trading strategies and voting-based integration strategy (see Algorithms 8 to 13) have a common structure. The inputs are two ETFs: ETF50 and ETF300 (*data*), training window size ($Wind$) and transaction costs (*cost*). The out-of-sample forecasts ($y_{predict}(i)$) are produced by training the models using the previous $Wind$-day data information before i day (from $i - Wind$ to i). Based on the daily forecasts, three trading strategies with different trading rules are designed.

The first type of trading strategy is the single machine learning based trading strategy. Algorithm 8 illustrates the long only ('L') machine learning based strategy. As the name suggests, only opportunities to buy are considered. In this case, we open the position when the stock will go upward, and close the position when the stock is predicted to go into a downtrend. In Algorithm 9, long and short ('LS') strategy demonstrates that at initial state ($S = 0$), if it is predicted that stock will go up, we take a long position ($S = 1$) for the upcoming day whereas a short position is taken ($S = -1$) today if a downward prediction is obtained. Meanwhile, we keep a record of the previous one position (or signal[9]) as it is needed to determine the following decision later. However, we only record the previous one position of the trade since it is not necessary to keep the history of previous positions. The current position is determined by the last previous position and the predicted signal together. If the same signal is predicted today compared with yesterday's signal, the current position is the same as the last previous position. On the contrary, when an opposite signal is predicted, the current position is updated based on the new predicted signal, that is to say, we close the existing position and open a new position in the opposite direction. The trading process is terminated by closing out the existing position at the end of the testing period.

Voting-based strategy is a valid single machine learning based trading strategy. Three predictors used in voting based strategy are the random forest, logistic regression and SVM, respectively. For each predictor, we set $y^{M}_{predict} = 1$ if M-th machine learning classifier shows an 'upward' prediction for the next day, otherwise, we set $y^{M}_{predict} = 0$, where M = 1, 2, 3. Instead of using the single machine learning predictor to generate one trading signal in each trading day, another two machine learning predictors identify two separate signals on the same day. Finally, all three classifiers together determine the buy ($Vote > 1$) or sell ($Vote < 2$) decision if any two classifiers give the consistent prediction result. For the long - only ('L') strategy, if any two of the single machine learning predictors give a counter to the previous one position, we should close the existing long position (see Algorithm 12). For the long and short ('LS') strategy, when all machine learning predictors simultaneously show the opposite signals, we close the existing position and open a new position in the opposite direction (see Algorithm

[9]'Long position' means that the last predicted signal is 'buy signal' while 'short position' indicates that a 'sell signal' was predicted at the previous day.

13).

Finally, time series (AR model with lag = 2) based strategy uses the lagged stock price information to predict the stock price. On each trading day, the predicted price movement is obtained by comparing the one day ahead price prediction with the current price. That is, a buy signal is generated if the predicted price will go up compared with the current price, otherwise, a sell signal is obtained. The trading rule for AR model is the same as that for single machine learning model ('LS' strategy). The lagged return is included with the purpose to check whether the time series properties of the past information contain any useful information in forecasting the future returns.

4.6.2 Trading using the multi-day machine learning prediction

As previously discussed, in the machine learning trading strategies with binary labels, investment decisions are made based on the price information for the next day. However, under the formulation of trading strategies based on the next few days' forecasts, we consider the sequence of ups and downs over the next several days. To examine whether the information of multi-day forecasts is able to improve the performance of machine learning based trading strategies, we simply classify the target by counting the number of upward price movements in the multiple days sequence day by day.

There are several changes made by multi-day forecast based trading strategies with machine learning compared to the single day ahead forecast based trading strategies. First, the trading signal of the one day ahead forecast based trading system is either 'buy' or 'sell', so there are no neutral signals. In this case, when a reverse signal is forecasted, we close the previous position and open a new opposite

Input: $data, p, n, a, Wind, cost$
Output: $TotalReturn_{BH}, TotalReturn_L$
Data: Close, Open, High, Low, Volume
Calculate feature matrix X from data

$y_{all}[i]$ = sign (Close[i+1]-Close[i])

y[i] = sum all elements in (y_{all}[i:i+n]) ; `// Count n day movements`

Train $(X[i], y[i])$ in AICLASSIFIER

Initial State $S = 0$
for *i in test set* **do**
  da(i) = data[i-Wind,i]
  $y_{predict}(i)$ = AICLASSIFIER(da(i))
  if $S = 0$ *and* $y_{predict}(i) == p$ **then**
    Return= ln(P(i+1)/P(i))-cost;
    S = 1 ; `// Open long`
  if $S = 1$ *and* $y_{predict}(i) > a$ **then**
    Return= ln(P(i+1)/P(i)) ; `// Hold long`
  else
    Return = 0;
    S = 0 ; `// Close long`
  end
  ReturnBH = ln(P(i+1)/P(i))
end

TotalReturn = cumsum(Return)

TotalReturn$_{BH}$ = $cumsum(ReturnBH)$

Algorithm 10: Back testing for trading using multi-day machine learning prediction ('L')

```
Input: data, p, q, n, a, b, Wind, cost
Output: TotalReturn_BH, TotalReturn_LS
Data: Close, Open, High, Low, Volume
Calculate feature matrix X from data
y_all[i] = sign (Close[i+1]-Close[i])
y[i] = sum all elements in (y_all[i:i+n]) ;        // Count n day movements
Train (X[i], y[i]) in AICLASSIFIER
Initial State S = 0
for i in test set do
    da(i) = data[i-Wind,i]
    y_predict(i) = AICLASSIFIER(da(i))
    if S = 0 and y_predict(i) == p then
        Return= ln(P(i+1)/P(i))-cost;
        S = 1 ;                                     // Open long
    if S = 0 and y_predict(i) == q then
        Return= ln(P(i+1)/P(i))-cost;
        S = - 1 ;                                   // Open short
    if S = 1 and y_predict(i) > a then
        Return= ln(P(i+1)/P(i)) ;                   // Hold long
    if S = −1 and y_predict(i) < b then
        Return= ln(P(i)/P(i+1)) ;                   // Hold short
    if S = 1 and y_predict(i) == a then
        Return= 0 ;                                 // Close long
        S = 0
    if S = −1 and y_predict(i) == b then
        Return= 0 ;                                 // Close short
        S = 0
    else
        Return = 0
    end
    ReturnBH = ln(P(i+1)/P(i))
end
TotalReturn = cumsum(Return)
TotalReturn_BH = cumsum(ReturnBH)
```

Algorithm 11: Back testing for trading using multi-day machine learning prediction ('LS') position at the same time, according to 'LS' strategy. However, as for the trading system with multi-day forecasts, since we have more accurate signals considering the multiple day sequence for the trend, we can close the existing position and wait for the next trading signal (i.e. strong 'buy' or strong 'sell' signal) to open a new trading position, according to 'LS' strategy. Second, multi-day forecast based trading strategies take account of investors' risk-return preferences to set parameters (p, q, a, b), while the trading rule for one-day forecast only switches between long position and short position based on signals without considering investors' risk preference. For example, risk averse investors may close the long position as long as the buy signal disappears while risk loving investors exit the long position only when the weak sell signal is obtained. The difference between these two types of investor is how much risk they can bear if neither a buy signal nor a sell signal is generated. That is to say, risk averse investors cannot keep investing in the long position in any other cases unless all signals generated before are strong buy signals. Nevertheless, risk loving investors can hold the long position in all cases which has a buy opportunity. This is because risk loving investors prefer to take a risk to profit from holding the long position. More details about the parameters (p, q, a, b) used in this thesis are shown in the empirical results in section (4.7).

Algorithms 10 and 11 demonstrate the 'L' and 'LS' strategies with consideration of n-day ahead price movements to classify our target (or trading signal) y into n classes. Historic close, open, high, low prices and volume are passed into machine learning models to generate trading signals. Every day, we re-train updated information X to predict trading signals y, where y counts how many positive price movements are observed during the next n-days. In 'L' strategy, buy decisions are triggered whenever the trading signal is predicted to be p, where

p is any integer between $\left\lceil \frac{n}{2} \right\rceil$ and n. When p is large, the strong upward price movement is expected during the following n days. The long position is held by investors as long as a buy or neutral signal is forecasted in the following n days. Once y is predicted to be a, the buy position is closed. The value of a is chosen based on investor's risk preference and tolerance: risk averse investors may close the long position at the first time the trading signal reaches a neutral signal (a $= \left\lfloor \frac{n}{2} \right\rfloor$) while risk loving investors exit the long position when a is any number between 0 and $\left\lfloor \frac{n}{2} \right\rfloor$.

'LS' strategy discovers both buy and sell opportunities to trade. New short position is opened whenever a strong sell signal is predicted ($y_{predict} = q$, where q $= 0, ..., \left\lfloor \frac{n}{2} \right\rfloor$). When the short position has already been opened, and the uptrend price movement is predicted ($S = 0$ and $y_{predict} = b$), the long position is closed. The trading process then restarts from the initial state that no position has been taken.

4.7 Empirical results

This section evaluates different variants of machine learning based trading strategies over various time periods (or test sets). The trading performances are analysed and possible improvements are suggested. Starting with the one day forecast based strategy using single machine learning classifiers, the profitability of one-day prediction based trading strategy is discussed. In addition, the risk and return analysis for the majority vote trading strategies are verified to analyse whether the voting based one-day trading signals can be improved. Furthermore, the possible variants of multiple days forecasts based trading strategies and their performances are presented to overcome the shortcomings of the one day prediction based trading strategies. The performance of the best machine learning based trading strategy is compared with that of the trend following strategy with respect to the risk-adjusted returns. Finally, some empirical results for all the considered trading strategies are summarised.

4.7.1 Trading using the one day ahead machine learning prediction

This section presents the results of one day ahead forecasting strategy with machine learning over several trading periods, namely the most recent 1 week, 1 month, 3 months, 6 months, 1 year, and 2 years[10]. Evaluating the one day forecast based trading strategies using simple machine learning with ETF50 and ETF300 reveals the following findings (see Tables 4.5 to 4.8).

When the investment horizon is less than 1 year (i.e 1 week, 1 month, 3 months and 6 months), single day forecast based trading strategies may generate positive daily returns after transaction costs during the last 1 month and 3 months. Although SVM, logistic regression and the vote-based classifier can produce positive returns, none of the investigated machine learning models can beat the buy and hold trading strategies for both ETF50 and ETF300.

As can be observed in the last 250 trading days, single day forecast based trading strategies yield negative returns in most experiments, while we can only note positive returns in a few cases. Compared to ETF50, ETF300 has more opportunity to gain positive returns before transaction costs according to annualised excess returns (-0.13 on average) and risk-adjusted returns (-0.55 on average) in the last 250 days. Trade ETF300 by machine learning models can generate higher returns with an annualised rate of 4.55% on average after transaction costs. Although these results seem to be low, some risk adjusted returns are higher for

[10]The last 5 days, 21 days, 63 days, 125 days, 250 days, and 500 days, individually.

machine learning models compared with the buy-and-hold strategy. When comparing with the second benchmark, for the same period, AR model yields an excess return of -0.3151. In this period, both ETF50 and ETF300 are in downward trends. Neither active trading strategies nor passive trading strategy can generate positive returns after transaction costs[11].

With respect to 500 trading day returns, all machine learning models are profitable (0.023 for ETF50 and 0.064 for ETF300 on average) before transaction costs according to excess returns; however, no machine learning models are profitable (-0.195 for ETF50 and -0.124 for ETF300 on average) after transaction costs. This result is consistent with the observation from the cumulative returns in Figure 4.7, which indicates that the total return for random forest based strategy is positive before transaction costs but negative after transaction costs are included. Following the previous table (Table 2.1) in section 2.1.2, the average transaction costs for ETF50 and ETF300 per trade are 6.225e-3 and 4.824e-3, the result demonstrates that the annualized transaction costs in this case are 0.259 and 0.188, individually. This implies that the transaction costs in total are high for the trading strategies developed in this section. The possible reason is that the number of trades utilising the one-day ahead forecast is high according to Table 4.12, and the improved results of multi-day forecast based trading strategy are presented later in section 4.7.2.

Based on the evidence from the prediction of ETF50 price movements (Table 4.4), we find that the use of random forest (when N=10) improves the accuracy compared with the logistic regression. However, random forest yields less excess returns compared to logistic regression as shown in Panel C and Panel D of Table

[11]'Active trading strategies' in this section include all developed one day forecast based trading strategies and AR model based strategy. 'Passive trading strategy' is referred to as buy-and-hold in all sections.

ETF50	Before transaction costs								After transaction costs									
	RF		SVM		LR		Vote		RF		SVM		LR		Vote		BH	AR
	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS		
Panel A: 5 trading day daily returns when Wind = 1346 and CV = 300																		
Average return (e-3)	-1.9390	-6.5640	-2.8640	-8.4150	-2.081	-6.8480	-1.9390	-6.5640	-5.5880	-3.8660	-4.4750	-3.2730	-4.1200	-5.3540	-5.5880	-5.3540	-9.2770	-7.0700
Standard deviation	0.0085	0.0128	0.0078	0.0114	0.0084	0.0126	0.0085	0.0128	0.0068	0.0035	0.0073	0.0034	0.0076	0.0074	0.0068	0.0074	0.0171	0.0158
t-Statistic	-0.5080	-1.1397	-0.8140	-1.6430	-0.5474	-1.2065	-0.5080	-1.1397	-1.8210	-2.4216	-1.3700	-2.1419	-1.2011	-1.6010	-1.8210	-1.6010	-1.2100	-1.0000
Minimum	-0.0168	-0.0238	-0.0168	-0.0238	-0.0168	-0.0238	-0.0168	-0.0238	-0.0168	-0.0068	-0.0168	-0.0068	-0.0168	-0.0168	-0.0168	-0.0168	-0.0359	-0.0353
Median	0.0000	0.0007	0.0000	-0.0046	0.0000	-0.0007	0.0000	0.0007	-0.0055	-0.0055	0.0000	-0.0046	-0.0007	-0.0055	-0.0055	-0.0055	-0.0007	0.0000
Maximum	0.0046	0.0046	0.0024	0.0024	0.0046	0.0046	0.0046	0.0046	0.0000	0.0000	0.0000	0.0007	0.0024	0.0024	0.0000	0.0024	0.0046	0.0000
Skewness	-1.3055	-0.4843	-1.4270	-0.3998	-1.2679	-0.4560	-1.3055	-0.4843	-0.9200	0.3426	-1.1539	0.2574	-1.0209	-0.5819	-0.9200	-0.5819	-0.8063	-1.5000
Kurtosis	3.0140	1.4230	3.1650	1.5030	2.9650	1.4390	3.0140	1.4230	2.5090	1.1940	2.6730	1.3010	2.5510	2.1800	2.5090	2.1800	2.0780	3.2500
Panel B: 21 trading day daily returns when Wind = 1330 and CV = 300																		
Average return (e-3)	4.3209	2.4912	6.1556	6.1607	3.6564	1.1622	5.2021	4.2536	-1.7771	-3.4993	4.3238	4.8302	2.0831	0.8141	2.8409	3.1060	3.7190	0.5353
Standard deviation	0.0183	0.0215	0.0187	0.0207	0.0202	0.0216	0.0186	0.0212	0.0095	0.0084	0.0190	0.0186	0.0205	0.0203	0.0192	0.0191	0.0225	0.0183
t-Statistic	1.0810	0.5298	1.5069^{+}	1.3598^{+}	0.8258	0.2458	1.2799	0.9170	-0.8537	-1.8947	1.0389	1.1880	0.4645	0.1837	0.6765	0.7429	0.7548	0.1334
Minimum	-0.0220	-0.0238	-0.0220	-0.0238	-0.0317	-0.0317	-0.0220	-0.0238	-0.0220	-0.0220	-0.0220	-0.0220	-0.0317	-0.0317	-0.0220	-0.0220	-0.0359	-0.0374
Median	0.0000	0.0007	0.0000	0.0024	0.0000	-0.0007	0.0000	0.0007	0.0000	-0.0055	0.0000	0.0000	0.0000	-0.0007	0.0000	0.0000	0.0029	0.0000
Maximum	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0236	0.0236	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0673
Skewness	2.5288	1.6807	2.1081	1.4789	1.6968	1.5985	2.2898	1.5852	0.7425	0.9554	2.2455	2.3888	1.8535	2.0807	2.4081	2.4013	0.9843	1.8792
Kurtosis	10.7280	6.5600	8.9580	6.2740	7.8870	6.9110	9.6010	6.3030	5.1020	7.0830	9.1610	9.7070	8.0500	8.9330	9.5710	9.5990	5.6670	10.3020
Panel C: 63 trading day annual returns when Wind = 1288 and CV = 300																		
Average return (e-3)	1.7937	1.6410	2.1892	2.4321	1.3691	0.7920	1.6744	1.4025	-1.4339	-1.9819	1.4471	2.0683	0.8758	0.6856	0.4284	0.6061	1.2400	-1.1700
Standard deviation	0.0121	0.0144	0.0123	0.0143	0.0131	0.0145	0.0121	0.0144	0.0071	0.0071	0.0123	0.0128	0.0130	0.0136	0.0121	0.0126	0.0151	0.0144
t-Statistic	1.1744	0.9010	1.4077	1.3459	0.8276	0.4326	1.0958	0.7687	-1.5988	-2.2006	0.9315	1.2727	0.5332	0.3976	0.2806	0.3788	0.6490	-0.6437
Minimum	-0.0220	-0.0238	-0.0220	-0.0238	-0.0317	-0.0317	-0.0220	-0.0238	-0.0220	-0.0220	-0.0220	-0.0220	-0.0317	-0.0317	-0.0220	-0.0220	-0.0359	-0.0595
Median	0.0000	0.0000	0.0000	0.0012	0.0000	-0.0008	0.0000	0.0000	0.0000	-0.0008	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0008	0.0000
Maximum	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0236	0.0236	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0673
Skewness	3.2538	1.9009	3.0036	1.8286	2.4490	1.8272	3.2616	1.9202	0.6730	0.8233	3.1955	2.6697	2.6214	2.2619	3.5700	3.0162	1.3720	0.0729
Kurtosis	20.4330	10.8450	18.6180	10.7460	15.9590	11.1410	20.4890	10.9040	7.0290	6.1190	19.4230	15.7830	16.7980	14.0330	21.9530	18.1210	9.7730	14.8220
Panel D: 125 trading day annual returns when Wind = 1226 and CV = 300																		
Average return (e-3)	1.6510	2.6970	0.7740	0.9428	0.2911	-0.2291	0.5145	0.4239	-1.2820	-1.9800	-0.5028	-0.6077	-0.5107	-0.9951	-1.0160	-1.3970	0.3513	0.0868
Standard deviation	0.0128	0.0156	0.0120	0.0158	0.0119	0.0159	0.0119	0.0159	0.0071	0.0067	0.0110	0.0134	0.0114	0.0138	0.0108	0.0130	0.0162	0.0147
t-Statistic	1.4385^{+}	1.9225^{*}	0.7182	0.6634	0.2714	-0.0160	0.4830	0.2978	-1.9935	-3.2719	-0.5087	-0.5036	-0.5000	-0.8024	-1.0471	-1.1936	0.2419	0.0660
Minimum	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0220	-0.0220	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0743
Median	0.0000	0.0007	0.0000	-0.0004	0.0000	-0.0011	0.0000	-0.0008	0.0000	-0.0015	0.0000	0.0000	0.0000	-0.0007	0.0000	0.0000	0.0007	0.0000
Maximum	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0424	0.0250	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0673
Skewness	1.3750	0.7965	1.3225	0.5531	1.1952	0.5011	1.4100	0.6241	2.1815	1.1449	1.6076	0.7578	1.3677	0.7075	1.7904	0.9304	0.3998	-0.6961
Kurtosis	12.4880	6.2730	15.1030	6.4970	15.6080	6.6060	15.9090	6.5420	14.9490	6.6860	20.4980	10.8040	18.5760	10.0330	22.3120	12.3230	6.2620	13.7100

Table 4.5: Daily trading profits of machine learning based trading strategies with one-day forecast for ETF50 within 1 year.

ETF300	Before transaction costs								After transaction costs									
	RF		SVM		LR		Vote		RF		SVM		LR		Vote		BH	AR
	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS		
Panel A: 5 trading day daily returns when Wind = 1346 and CV = 300																		
Average return (e-3)	-0.8419	-8.6587	3.2118	-0.5514	3.2118	-0.5514	3.2118	-0.5514	-3.0354	-5.6926	-3.8684	-2.7624	-3.8684	-2.7624	-3.8684	-3.6167	-4.1622	-6.8441
Standard deviation	0.0064	0.0097	0.0114	0.0137	0.0114	0.0137	0.0114	0.0137	0.0048	0.0040	0.0045	0.0025	0.0045	0.0025	0.0045	0.0036	0.0194	0.0153
t-Statistic	-0.2940	-1.9802	0.6258	-0.0897	0.6258	-0.0897	0.6258	-0.0897	-1.4111	-3.1753	-1.9206	-2.3956	-1.9206	-2.3956	-1.9206	-2.2181	-0.4783	-1.0000
Minimum	-0.0110	-0.0202	-0.0110	-0.0110	-0.0110	-0.0110	-0.0110	-0.0110	-0.0110	-0.0103	-0.0110	-0.0054	-0.0110	-0.0054	-0.0110	-0.0084	-0.0354	-0.0342
Median	0.0000	-0.0103	0.0000	-0.0084	0.0000	-0.0084	0.0000	-0.0084	0.0000	-0.0054	-0.0041	-0.0041	-0.0041	-0.0041	-0.0041	-0.0041	0.0068	0.0000
Maximum	0.0068	0.0068	0.0202	0.0202	0.0202	0.0202	0.0202	0.0202	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0103	0.0000
Skewness	-0.6424	0.6320	0.3818	0.7360	0.3818	0.7360	0.3818	0.7360	-1.0676	0.2796	-0.7613	0.2695	-0.7613	0.2695	-0.7613	-0.1385	-0.9284	-1.5000
Kurtosis	2.6240	2.5770	2.2690	1.8910	2.2690	1.8910	2.2690	1.8910	2.5130	1.9300	2.3180	1.2650	2.3180	1.2650	2.3180	1.5810	2.2700	3.2500
Panel B: 21 trading day daily returns when Wind = 1330 and CV = 300																		
Average return (e-3)	5.2515	1.9890	6.6939	4.8737	5.7342	2.9544	6.6939	4.8737	-0.3244	-1.7649	5.5458	3.1341	4.6686	2.2569	5.5458	3.1341	6.3800	-0.6256
Standard deviation	0.0175	0.0208	0.0181	0.0203	0.0192	0.0207	0.0181	0.0203	0.0093	0.0080	0.0184	0.0190	0.0194	0.0198	0.0184	0.0190	0.0213	0.0180
t-Statistic	1.3750^+	0.4367	1.6879^+	1.0964	1.3678^+	0.6524	1.6879^+	1.0964	-0.1588	-1.0071	1.3742^+	0.7541	1.1011	0.5212	1.3742^+	0.7541	1.3700^+	-0.1587
Minimum	-0.0166	-0.0264	-0.0166	-0.0166	-0.0227	-0.0227	-0.0166	-0.0166	-0.0166	-0.0166	-0.0166	-0.0166	-0.0227	-0.0227	-0.0166	-0.0166	-0.0354	-0.0456
Median	0.0000	-0.0026	0.0000	-0.0019	0.0000	-0.0019	0.0000	-0.0019	0.0000	-0.0026	0.0000	-0.0019	0.0000	-0.0019	0.0000	-0.0019	0.0068	0.0000
Maximum	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0286	0.0286	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0413
Skewness	2.3819	1.5577	1.9828	1.5135	1.6511	1.5646	1.9828	1.5135	1.7136	2.4050	2.0495	2.1275	1.7513	1.9037	2.0495	2.1275	0.7589	-0.3968
Kurtosis	9.1930	5.8840	7.3380	5.3740	6.4880	5.6620	7.3380	5.3740	6.6760	11.4080	7.3540	7.6490	6.5830	6.9760	7.3540	7.6490	4.8900	4.5170
Panel C: 63 trading day annual returns when Wind = 1288 and CV = 300																		
Average return (e-3)	1.9090	1.1294	2.3340	1.9793	1.5137	0.3388	2.1357	1.5829	-1.3146	-1.2995	1.3137	0.9370	1.0874	-0.1075	1.2476	0.4864	1.9636	-1.2109
Standard deviation	0.0126	0.0148	0.0125	0.0147	0.0126	0.0148	0.0126	0.0147	0.0071	0.0075	0.0121	0.0132	0.0125	0.0138	0.0121	0.0131	0.0153	0.0134
t-Statistic	1.1971	0.6048	1.4806^+	1.0665	0.9515	0.1809	1.3403^+	0.8501	-1.4523	-1.3596	0.8567	0.5617	0.6892	-0.0614	0.8124	0.2942	1.0154	-0.7166
Minimum	-0.020	-0.026	-0.0204	-0.0252	-0.0227	-0.0256	-0.0204	-0.0252	-0.0204	-0.0186	-0.0204	-0.0252	-0.0227	-0.0256	-0.0204	-0.0252	-0.0354	-0.0500
Median	0.0000	-0.0019	0.0000	-0.0015	0.0000	-0.0019	0.0000	-0.0019	0.0000	-0.0019	0.0000	0.0000	0.0000	-0.0015	0.0000	0.0000	0.0009	0.0000
Maximum	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0286	0.0286	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0413
Skewness	2.5140	1.5330	2.6160	1.5200	2.4910	1.5090	2.5540	1.5650	1.2020	1.3680	2.9710	2.2480	2.6680	1.9660	2.9730	2.3940	1.1530	-1.1810
Kurtosis	13.7700	8.1120	14.0370	7.8990	14.2520	8.3850	13.6350	7.9720	9.2500	7.4740	16.4180	12.2560	15.0660	11.0670	16.3840	13.0170	7.2600	8.6700
Panel D: 125 trading day annual returns when Wind = 1226 and CV = 300																		
Average return (e-3)	1.2903	1.8756	0.8728	1.0405	-0.1051	-0.9154	0.5965	0.4879	-1.4218	-1.7518	-0.4081	-0.8843	-0.3433	-1.3495	-0.4341	-1.3315	0.4447	0.4594
Standard deviation	0.0130	0.0162	0.0117	0.0163	0.0114	0.0163	0.0119	0.0163	0.0058	0.0069	0.0105	0.0136	0.0106	0.0137	0.0105	0.0133	0.0166	0.0134
t-Statistic	1.1090	1.2880	0.8307	0.7112	-0.1030	-0.6254	0.5566	0.3330	-2.7262	-2.8132	-0.4311	-0.7224	-0.3620	-1.0979	-0.4583	-1.1174	0.2980	0.3807
Minimum	-0.0487	-0.0487	-0.0487	-0.0514	-0.0487	-0.0514	-0.0487	-0.0514	-0.0204	-0.0251	-0.0487	-0.0514	-0.0487	-0.0514	-0.0487	-0.0514	-0.0527	-0.0500
Median	0.0000	-0.0005	0.0000	-0.0015	0.0000	-0.0020	0.0000	-0.0020	0.0000	-0.0021	0.0000	-0.0019	0.0000	-0.0015	0.0000	-0.0020	0.0000	0.0000
Maximum	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0286	0.0286	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0472
Skewness	1.4041	0.8787	1.4577	0.5504	1.2554	0.4971	1.3750	0.6024	1.2599	1.0432	1.6189	0.5346	1.5712	0.5066	1.6233	0.6190	0.3169	0.1056
Kurtosis	11.3180	5.7650	14.3160	5.9720	15.3800	6.2190	13.4690	6.0210	11.4520	7.8480	19.9810	9.5110	19.8550	9.4820	19.9530	10.5650	5.7910	8.5340

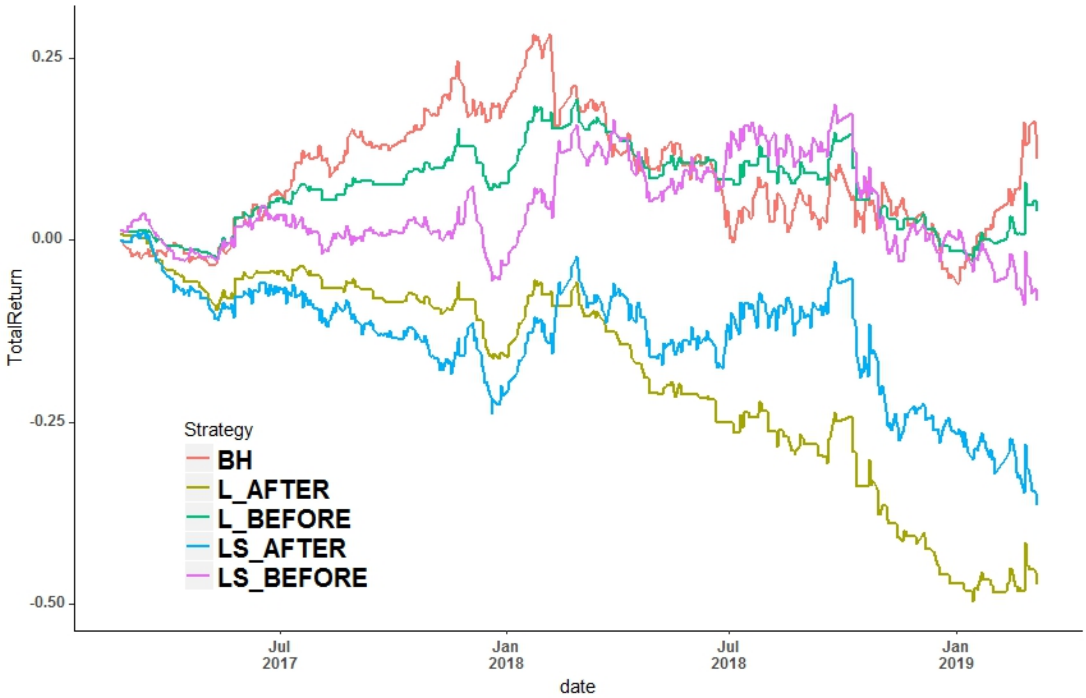
Table 4.6: Daily trading profits of machine learning based trading strategies with one-day forecast for ETF300 within 1 year.

ETF50	Before transaction costs								After transaction costs									
	RF		SVM		LR		Vote		RF		SVM		LR		Vote		BH	AR
	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS		
Panel A: 250 trading day daily returns when Wind = 1101 and CV = 300																		
Average return (e-3)	0.0803	-0.3625	-0.3270	-1.1770	0.2295	-0.0642	-0.2492	-0.2897	-0.8992	-1.9770	-0.8836	-2.2560	-0.3048	-0.8534	-0.8280	-2.1220	-0.3510	-1.1810
Standard deviation	0.0108	0.0184	0.0108	0.0184	0.0154	0.0184	0.0107	0.0121	0.0109	0.0189	0.0109	0.0187	0.0153	0.0183	0.0107	0.0188	0.0146	0.0129
t-Statistic	0.1176	-0.3106	-0.4770	-1.0105	0.2342	-0.0550	-0.3665	-0.3778	-1.3021	-1.6529	-1.2816	-1.8978	-0.3141	-0.7369	-1.2232	-1.7819	-0.3807	-1.4512
Minimum	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.1087
Median	0.0000	-0.0001	0.0000	-0.0005	0.0000	-0.0005	0.0000	0.0000	0.0000	-0.0018	0.0000	-0.0015	0.0000	-0.0017	0.0000	-0.0018	-0.0002	0.0000
Maximum	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0673	0.0673	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0673
Skewness	1.15350	-3.5171	0.9317	-3.5357	6.7710	4.0204	1.0308	0.7719	0.6482	-3.6050	0.7949	-3.6567	6.3912	3.8439	0.9150	-3.6058	0.3326	-2.5430
Kurtosis	14.0160	5.7470	14.1510	5.8680	16.3230	5.9150	14.6830	9.7730	11.6320	5.0450	13.8270	5.9190	16.0700	5.8940	14.7320	5.8480	5.7300	25.3100
Panel B: 500 trading day daily returns when Wind = 851 and CV = 300																		
Average return (e-3)	0.3225	0.0654	0.1969	0.1412	0.1590	0.0654	0.1749	0.1734	-0.6015	-1.0825	-0.3597	-0.8934	-0.4398	-0.7809	-0.4373	-1.0284	0.2211	-1.4690
Standard deviation	0.0083	0.0120	0.0088	0.0120	0.0087	0.0120	0.0088	0.0106	0.0083	0.0122	0.0088	0.0122	0.0088	0.0121	0.0087	0.0123	0.2311	0.0113
t-Statistic	0.8671	0.1217	0.5001	0.2627	0.4050	0.1217	0.4433	0.3645	-1.6122	-1.9716	-0.9066	-1.6360	-1.1074	-1.4317	-1.1114	-1.8668	0.1962	-2.8983
Minimum	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0486	-0.1087
Median	0.0000	0.0000	0.0000	-0.0004	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0016	0.0000	-0.0013	0.0000	-0.0008	0.0000	-0.0014	0.4195e-3	0.0000
Maximum	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0673	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0729	0.0673
Skewness	1.1932	0.5671	0.9171	0.5865	0.8894	0.5671	0.9258	0.8611	0.6226	0.6179	0.7423	0.6061	0.7609	0.5154	0.7699	0.6059	0.0517	-3.3550
Kurtosis	16.8720	3.9850	14.0720	3.9710	14.3150	3.9850	13.9390	7.1570	13.5190	3.8080	13.5110	3.9100	13.5000	3.9390	13.8990	3.7800	3.9690	32.6300
Panel C: 250 trading day annual returns when Wind = 1101 and CV = 300																		
Annual return	0.0202	-0.0913	-0.0823	-0.2966	0.0578	-0.0161	-0.0627	-0.0729	-0.2266	-0.4982	-0.2226	-0.5684	-0.0768	-0.2150	-0.2086	-0.5348	-0.0883	-0.2976
Excess return	0.0027	-0.1088	-0.0998	-0.3141	0.0403	-0.0336	-0.0802	-0.0904	-0.2441	-0.5157	-0.2401	-0.5859	-0.0943	-0.2325	-0.2261	-0.5523	-0.1058	-0.3151
Standard deviation	0.1715	0.2929	0.1720	0.2924	0.2459	0.2930	0.1706	0.1924	0.1733	0.3002	0.1730	0.2983	0.2436	0.2907	0.1699	0.2989	0.2311	0.2043
Downside deviation	0.1106	0.2361	0.1184	0.2430	0.1146	0.1631	0.1153	0.1317	0.1258	0.2532	0.1257	0.2552	0.1215	0.1695	0.1217	0.2531	0.1619	0.1733
Sharpe ratio	0.0160	-0.3715	-0.5807	-1.0744	0.1640	-0.1149	-0.4705	-0.4702	-1.4083	-1.7178	-1.3879	-1.9640	-0.3872	-0.8000	-1.3311	-1.8476	-0.4579	-1.5430
Sortino ratio	0.0248	-0.4611	-0.8438	-1.2926	0.3517	-0.2065	-0.6964	-0.6871	-1.9405	-2.0364	-1.9105	-2.2961	-0.7761	-1.3723	-1.8589	-2.1826	-0.6535	-1.8182
Maximum drawdown	0.2164	0.2961	0.2047	0.2047	0.2314	0.2331	0.2043	0.2304	0.5090	0.5980	0.3044	0.3027	0.2274	0.2654	0.3047	0.3447	0.2120	0.2594
Panel D: 500 trading day annual returns when Wind = 851 and CV = 300																		
Annual return	0.0812	0.0164	0.0496	0.0355	0.0400	0.0164	0.0440	0.0437	-0.1515	-0.2727	-0.0906	-0.2251	-0.1108	-0.1967	-0.1102	-0.2591	0.0557	-0.3701
Excess return	0.0637	-0.0010	0.0321	0.0180	0.0225	-0.0010	0.0265	0.0262	-0.1690	-0.2902	-0.1081	-0.2426	-0.1283	-0.2142	-0.1277	-0.2766	0.0382	-0.3876
Standard deviation	0.1320	0.1907	0.1397	0.1907	0.1394	0.1907	0.1401	0.1689	0.1324	0.1949	0.1408	0.1939	0.1410	0.1936	0.1397	0.1955	0.1919	0.1799
Downside deviation	0.0844	0.1283	0.0926	0.1270	0.0932	0.1283	0.0928	0.1102	0.0971	0.1395	0.0997	0.1379	0.1005	0.1385	0.0997	0.1400	0.1352	0.1616
Sharpe ratio	0.4830	-0.0053	0.2298	0.0948	0.1619	-0.0053	0.1898	0.1552	-1.2766	-1.4894	-0.7679	-1.2516	-0.9102	-1.1067	-0.9143	-1.4148	0.1992	-2.1188
Sortino ratio	0.7540	-0.0079	0.3465	0.1422	0.2419	-0.0079	0.2863	0.2377	-1.7411	-2.0798	-1.0840	-1.7586	-1.2770	-1.5467	-1.2805	-1.9756	0.2827	-2.3992
Maximum drawdown	0.2280	0.1927	0.2043	0.2443	0.3347	0.3043	0.2307	0.2045	0.5110	0.4230	0.3027	0.4041	0.2037	0.3042	0.2054	0.2047	0.2895	0.2310

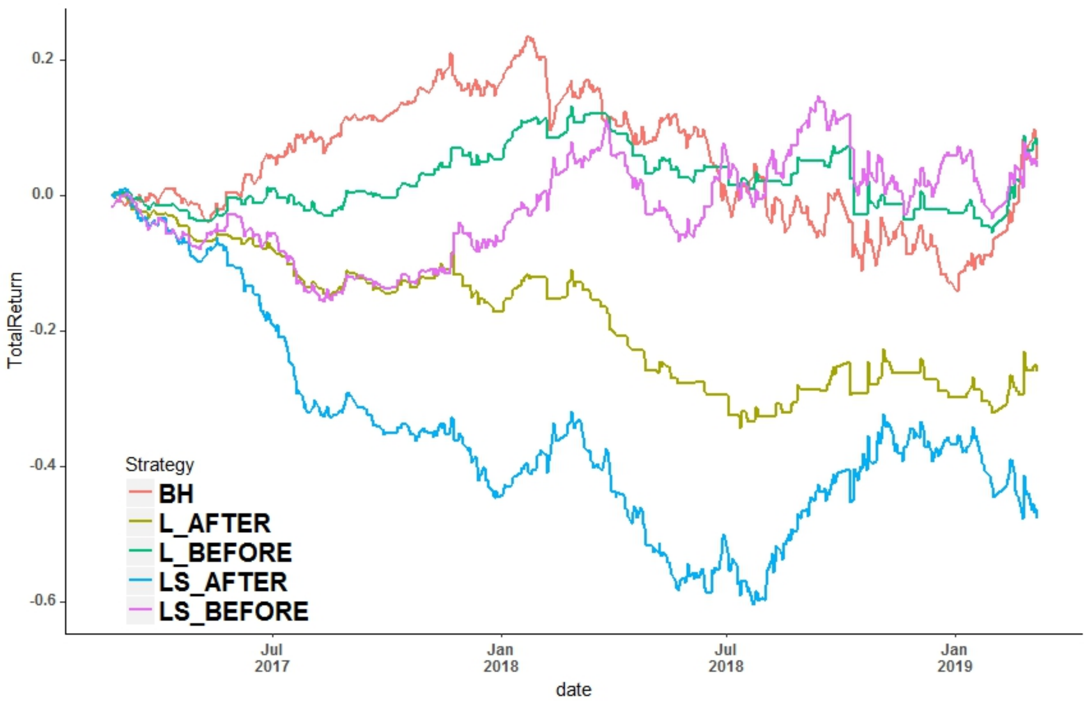
Table 4.7: Daily and annualised trading profits of machine learning based trading strategies with one-day forecast for ETF50 in one and two years.

ETF300	Before transaction costs								After transaction costs									AR
	RF		SVM		LR		Vote		RF		SVM		LR		Vote		BH	
	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS		
Panel A: 250 trading day daily returns when Wind = 1101 and CV = 300																		
Average return (e-3)	-0.0959	0.2089	0.4283	0.6820	0.2531	0.3316	0.3442	0.1349	-0.8623	-1.5353	0.0118	-0.0845	-0.0452	-0.0661	-0.0889	-0.2860	-0.4150	-0.5127
Standard deviation	0.0120	0.0145	0.0122	0.0161	0.0121	0.0161	0.0121	0.0148	0.0121	0.0161	0.0120	0.0162	0.0118	0.0161	0.0119	0.0162	0.0147	0.0121
t-Statistic	-0.1254	0.2262	0.5550	0.6669	0.3305	0.3241	0.4497	0.1435	-1.1225	-1.5041	0.0155	-0.0824	-0.0600	-0.6461	-0.1179	-0.2786	-0.4458	-0.6707
Minimum	-0.0487	-0.0514	-0.0487	-0.0514	-0.0487	-0.0514	-0.0487	-0.0514	-0.0529	-0.0529	-0.0487	-0.0514	-0.0487	-0.0514	-0.0487	-0.0514	-0.0514	-0.1032
Median	0.0000	-0.0014	0.0000	0.0005	0.0000	0.0002	0.0000	0.0000	0.0000	-0.0028	0.0000	-0.0011	0.0000	-0.0016	0.0000	-0.0012	-0.0012	0.0000
Maximum	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0640	0.0640	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0641
Skewness	3.6830	1.3460	3.5490	1.4590	3.6460	1.4960	3.6320	1.9680	3.0190	1.4790	3.3850	1.3770	3.5150	1.3790	3.4530	1.3730	0.3212	-2.5930
Kurtosis	16.8480	5.7750	15.5010	5.6500	16.3000	5.6790	16.2050	8.1270	14.8070	5.4920	16.2780	5.5210	17.3810	5.6310	16.9510	5.5150	5.6050	27.6100
Panel B: 500 trading day daily returns when Wind = 851 and CV = 300																		
Average return (e-3)	0.0002	0.0004	0.0001	0.0003	0.0003	0.0006	0.0001	0.0005	-0.0005	-0.2644	-0.0002	-0.0005	-0.0002	-0.0002	-0.0003	-0.0006	0.1049	-0.9879
Standard deviation	0.0080	0.0125	0.0083	0.0125	0.0082	0.0125	0.0082	0.0109	0.0080	0.0126	0.0083	0.0126	0.0081	0.0126	0.0082	0.0126	0.0120	0.0102
t-Statistic	0.6320	0.8281	0.3816	0.5223	0.7749	1.0367	0.3140	0.9525	-1.4408	-1.3850	-0.7304	-0.9005	-0.5459	-0.3968	-0.7960	-1.0306	0.1962	-2.1659
Minimum	-0.0487	-0.0487	-0.0487	-0.0514	-0.0487	-0.0514	-0.0487	-0.0487	-0.0487	-0.0487	-0.0487	-0.0514	-0.0487	-0.0975	-0.0487	-0.0514	-0.0528	-0.1032
Median	0.0000	-0.0005	0.0000	-0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0015	0.0000	-0.0008	0.0000	-0.0005	0.0000	-0.0009	0.0005	0.0000
Maximum	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0640	0.0640	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0682	0.0641
Skewness	1.3552	1.3460	0.7035	1.0630	0.8509	1.0995	0.7490	1.8738	0.8649	1.1860	0.5932	0.9272	0.7751	-0.5103	0.6749	0.9556	0.0414	-3.1880
Kurtosis	18.7500	8.5300	13.9420	8.6160	14.5010	8.5390	14.9980	15.2170	15.8350	6.9710	14.1050	7.3510	15.7290	9.7060	15.3540	7.3330	4.2840	32.7800
Panel C: 250 trading day annual returns when Wind = 1101 and CV = 300																		
Annual return	-0.0241	0.0526	0.1079	0.1718	0.0637	0.0835	0.0867	0.0339	-0.2172	-0.3869	0.0029	-0.0213	-0.0113	-0.1666	-0.0224	-0.0720	-0.1045	-0.1292
Excess return	-0.0416	0.0351	0.0904	0.1543	0.0462	0.0660	0.0692	0.0165	-0.2348	-0.4044	-0.0145	-0.0388	-0.0288	-0.1841	-0.0399	-0.0895	-0.1220	-0.1467
Standard deviation	0.1919	0.2313	0.1937	0.2566	0.1922	0.2568	0.1921	0.2360	0.1928	0.2562	0.1907	0.2572	0.1887	0.2569	0.1893	0.2577	0.2335	0.1918
Downside deviation	0.1075	0.1641	0.1043	0.1576	0.1048	0.1601	0.1040	0.1439	0.1215	0.1737	0.1075	0.1644	0.1059	0.1699	0.1077	0.1668	0.1635	0.1580
Sharpe ratio	-0.2172	0.1519	0.4668	0.6014	0.2408	0.2572	0.3604	0.0699	-1.2178	-1.5784	-0.0762	-0.1508	-0.1530	-0.7168	-0.2108	-0.3476	-0.5225	-0.7646
Sortino ratio	-0.3876	0.2238	0.8670	0.9795	0.4417	0.4127	0.6655	0.1147	-1.9327	-2.3280	-0.1351	-0.2361	-0.2727	-1.0838	-0.3707	-0.5369	-0.7462	-0.9286
Maximum drawdown	0.3169	0.2371	0.2607	0.2633	0.3607	0.3633	0.2612	0.2663	0.3607	0.3207	0.3565	0.2592	0.2565	0.3532	0.2565	0.4592	0.2120	0.2594
Panel D: 500 trading day annual returns when Wind = 851 and CV = 300																		
Annual return	0.0574	0.1169	0.0358	0.0738	0.0721	0.1463	0.0290	0.1170	-0.1314	-0.1968	-0.0691	-0.1282	-0.0501	-0.0565	-0.0738	-0.1468	0.0264	-0.2490
Excess return	0.0399	0.0994	0.0183	0.0563	0.0546	0.1288	0.0115	0.0995	-0.1489	-0.2143	-0.0866	-0.1457	-0.0676	-0.0740	-0.0913	-0.1643	0.0089	-0.2665
Standard deviation	0.1280	0.1990	0.1323	0.1991	0.1311	0.1989	0.1302	0.1731	0.1286	0.2002	0.1333	0.2007	0.1294	0.2008	0.1307	0.2006	0.1897	0.1619
Downside deviation	0.0812	0.1217	0.0894	0.1284	0.0860	0.1248	0.0880	0.1032	0.0914	0.1346	0.0944	0.1377	0.0900	0.1459	0.0922	0.1377	0.1340	0.1415
Sharpe ratio	0.3119	0.4999	0.1386	0.2829	0.4166	0.6480	0.0884	0.5751	-1.1589	-1.0706	-0.6498	-0.7265	-0.5227	-0.3688	-0.6990	-0.8189	0.0470	-1.6457
Sortino ratio	0.4913	0.8174	0.2051	0.4384	0.6353	1.0327	0.1309	0.9643	-1.6285	-1.5920	-0.9168	-1.0581	-0.7516	-0.5075	-0.9906	-1.1930	0.0666	-1.8833
Maximum drawdown	0.1047	0.2508	0.3410	0.2434	0.3920	0.2434	0.3910	0.3402	0.3168	0.3354	0.2310	0.2380	0.2912	0.2032	0.3230	0.2480	0.2895	0.2379

Table 4.8: Daily and annualised trading profits of machine learning based trading strategies with one-day forecast for ETF300 in one and two years.



(a) ETF50 in 500 days



(b) ETF300 in 500 days

Figure 4.1: Cumulative returns (after transaction costs) for one day random forest forecast based strategy in the last 500 trading days.

4.7. This finding implies that a high '*Accuracy*' in cross-validation set does not necessarily result in a high trading profit in daily and annual returns.

It is worthwhile to point out that majority vote performs even worse than single machine learning models since the Sharpe ratio is reduced from -0.8726 to -0.9481 on average. Therefore, the simple majority vote method does not provide a more accurate signal for making trading decisions compared to single machine learning models, under the same experimental settings.

Viewing all the results, we conclude that it is possible for the developed strategies to generate profits based on one day prediction over 500 trading days. The major limitation of machine learning based trading strategy is that these trading strategies are profitable before transaction costs but unprofitable after transaction costs during the period. The possible explanation is that if the trading signals of the first task are volatile, we shift between long position and short position frequently. Then, the total expenditure on transaction costs is high. In some trades, profits generated in each trade are insufficient for trading to be profitable if transaction costs are taken into account. Therefore, the improvement of multi-day forecast based trading strategy is proposed by predicting the next few days' price movement instead of one day price movement to avoid this type of unprofitable trades.

4.7.2 Trading using the multi-day machine learning prediction

The profitability of multi-day forecast based trading strategy is verified in this section. The trading rules which are parameterised as (p, q, a, b) to carry out purchase and sell decisions are defined below.

The idea of trading rules using the multi-day machine learning prediction is to reduce the number of trades. We provide a possible way here to reasonably select parameters, and our final determination is $p \geq 3$, $a = 1$, $b = 3$ and $q = 0$ according to the following reasons. Since p is chosen from the label to trigger buy signals, it should be larger than 2 to confirm there is a buy signal in the next 4 days. Therefore, we follow the rule of opening the long position when $m \geq 3$ if no positions have been opened already. It is important to stress that the determination of a and b will directly affect the number of trades. Once the long position is opened, we keep the long position when it shows a buy signal or a neutral signal, and we only close the long position when a weak sell signal ($a = 1$) occurs to reduce the number of trades.

Tables 4.9 and 4.10 depict the trading performances of daily returns over 6 trading periods with the parameters N[12]$= 100$ for random forest, and $nn(3, 2)$ for MLP. As illustrated, the trading performance, that measured by average daily return, has been improved by the multi-day random forest and MLP strategies compared to single day machine learning strategies (see Tables 4.5 and 4.6). Moreover, over the 1 year and 2 years trading period, 'LS' trading strategies always yield higher average returns compared with 'L' trading strategies for both random forest and MLP based strategies before transaction costs. This result is also supported by machine learning based trading strategies even after transaction costs. It is consistent with the result that the cumulative returns for 'LS' strategy are higher than that for 'L' strategy shown in Figures 4.2 and 4.5.

If considering the risk, the standard deviation for 'L' strategies is usually smaller than that for 'LS' strategies no matter which test period, asset, or classifier used. We perform a one-sided t-test with the null hypothesis test that the mean of the return is above 0. Random forest yields a positive daily return at the 10%

[12]For the multi-class classification prediction instance, to obtain steady trading results, $N = 100$ is the default value in scikit-learn in Python (in version 0.22) for random forest.

significance level for both the 250 days and the 500 days test periods. Therefore, both indices have an above-zero mean of daily returns. Median returns appear to be 0% as there are no frequent trades. The magnitude of the maximum daily returns is higher than that of the minimum daily returns, which indicates that the extreme profits have great chances to cover the extreme losses. None of the daily returns show normality: skewness and kurtosis statistics of the returns for each day are generally above 0 and 3 respectively in all active trading strategies. Similar to other financial markets, the daily returns have fat tailed distribution. Both skewness and kurtosis show that positive daily returns can be generated in machine learning based trading strategies. The non-normality of the trading returns results in looking for an alternative to measure the risk of the strategy: the maximum drawdown. It shows that the smaller maximum drawdown with 0.1 is observed for both stock indices, while buy-and-hold strategy has a maximum drawdown of 0.25. Therefore, in all cases the investor would obtain a favourable risk-adjusted return using random forest by analysing the final return and the maximum drawdown.

Apart from t-statistics, we also find other positive evidence in favour of the trading signals generated by random forest from the perspective of risk-adjusted return analysis. Random forest is a better classifier since its Sharpe ratios range from 1.0480 to 2.6300, which are always above 1 after transaction costs. On the contrary, MLP cannot generate more risk-adjusted returns especially during 250 trading days when the market shows downward trends in price. Random forest is better than the other algorithm as it can deal with highly imbalanced data by some re-sampling methods. It is the case that the target only contains a few samples in some classes (see Table 4.2).

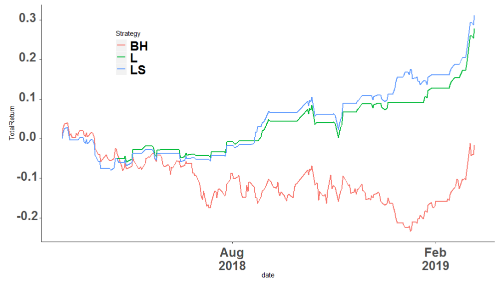
On the perspective of the total return of investment, multiple days random

	Before transaction costs								After transaction costs										
	ETF50				ETF300				ETF50					ETF300				
	RF		MLP		RF		MLP		RF		MLP		BH	RF		MLP		BH
	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS		L	LS	L	LS	
Panel A: 5 trading day daily returns when Wind = 1646																		
Average return (e-3)	0.5857	4.7673	8.8173	8.8173	0.0000	0.0000	0.0000	0.0000	-1.1132	3.6541	7.7041	7.7041	-9.2772	0.0000	0.0000	0.0000	0.0000	-6.8441
Standard deviation	0.0013	0.0106	0.0385	0.0385	0.0000	0.0000	0.0000	0.0000	0.0024	0.0115	0.0362	0.0362	0.0171	0.0000	0.0000	0.0000	0.0000	0.0153
t-Statistic	1.0000	1.0000	0.5115	0.5115	NaN	NaN	NaN	NaN	-1.0000	0.7082	0.4753	0.4753	-1.2104	NaN	NaN	NaN	NaN	-1.0000
Minimum	0.0000	0.0000	-0.0317	-0.0317	0.0000	0.0000	0.0000	0.0000	-0.0055	-0.0055	-0.0317	-0.0317	-0.0359	0.0000	0.0000	0.0000	0.0000	-0.0342
Median	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0007	0.0000	0.0000	0.0000	0.0000	0.0000
Maximum	0.0029	0.0238	0.0729	0.0729	0.0000	0.0000	0.0000	0.0000	0.0000	0.0238	0.0673	0.0673	0.0046	0.0000	0.0000	0.0000	0.0000	0.0000
Skewness	1.5000	1.5000	0.9513	0.9513	NaN	NaN	NaN	NaN	-1.5000	1.3271	0.8786	0.8786	-0.8063	NaN	NaN	NaN	NaN	-1.5000
Kurtosis	3.2500	3.2500	2.7730	2.7730	NaN	NaN	NaN	NaN	3.2500	3.0640	2.7310	2.7310	2.0780	NaN	NaN	NaN	NaN	3.2500
Panel B: 21 trading day daily returns when Wind = 1630																		
Average return (e-3)	5.2162	4.5540	3.8185	3.3570	0.9989	-1.3343	7.5528	7.7751	7.4588	3.9612	3.0233	2.8269	3.7190	0.4918	-1.5953	6.9578	7.5768	-0.6256
Standard deviation	0.0162	0.0185	0.0076	0.0117	0.0033	0.0044	0.0161	0.0177	0.0209	0.0186	0.0072	0.0112	0.0225	0.0016	0.0048	0.0158	0.0177	0.0183
t-Statistic	1.4732^{+}	1.1235	2.2736	1.3122	1.3799^{+}	-1.3623	2.1390*	2.0081*	1.6327	0.9730	1.9137*	1.1504	0.7548	1.3358^{+}	-1.5161	2.0130*	1.9541*	-0.1587
Minimum	-0.0012	-0.0317	-0.0042	-0.0317	0.0000	-0.0164	-0.0065	-0.0166	0.0000	-0.0317	-0.0055	-0.0317	-0.0359	0.0000	-0.0164	-0.0065	-0.0166	-0.0456
Median	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0029	0.0000	0.0000	0.0000	0.0000	0.0000
Maximum	0.0729	0.0729	0.0236	0.0236	0.0137	0.0065	0.0682	0.0682	0.0729	0.0729	0.0200	0.0220	0.0729	0.0072	0.0065	0.0682	0.0682	0.0413
Skewness	3.7309	2.2382	1.5794	-0.6663	3.2122	-1.9710	2.7200	1.9680	2.7279	2.2976	1.4673	-0.7939	0.9843	3.4492	-1.6944	2.9502	1.9931	-0.3968
Kurtosis	15.9510	10.4730	4.1590	5.4100	12.0600	7.7870	10.5720	7.5090	8.5420	10.5880	3.8870	5.7900	5.6670	13.7880	5.9580	11.8750	7.5560	4.5170
Panel C: 63 trading day annual returns when Wind = 1588																		
Average return (e-3)	0.6378	-0.2005	0.4626	1.5228	0.2000	1.2508	2.5187	2.6007	0.3170	-0.3771	0.4626	0.9443	1.2398	0.1496	0.9654	2.0559	2.5346	-1.2109
Standard deviation	0.0100	0.0084	0.0126	0.0127	0.0055	0.0123	0.0113	0.0132	0.0106	0.0083	0.0126	0.0133	0.0151	0.0063	0.0123	0.0114	0.0132	0.0134
t-Statistic	0.5056	-0.1880	0.2914	0.9479	0.2884	0.8070	1.7605^{+}	1.5566 $^{+}$	0.2367	-0.3586	0.2914	0.5610	0.6490	0.1869	0.6215	1.4214^{+}	1.5152^{+}	-0.7166
Minimum	-0.0187	-0.0317	-0.0317	-0.0317	-0.0175	-0.0256	-0.0227	-0.0227	-0.0178	-0.0317	-0.0317	-0.0317	-0.0359	-0.0228	-0.0256	-0.0227	-0.0227	-0.0500
Median	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0008	0.0000	0.0000	0.0000	0.0000	0.0000
Maximum	0.0729	0.0238	0.0729	0.0729	0.0205	0.0682	0.0682	0.0682	0.0729	0.0238	0.0729	0.0729	0.0729	0.0205	0.0682	0.0682	0.0682	0.0413
Skewness	5.9629	-0.1009	2.8425	2.6353	1.0537	2.5833	3.2968	2.0256	5.0534	-0.2095	2.8425	2.2881	1.3720	0.1667	2.5567	3.1597	2.0326	-1.1810
Kurtosis	44.8680	6.8770	19.3410	17.5610	9.0360	15.6810	19.7930	11.1210	36.1780	6.8560	19.3410	15.2020	9.7730	8.0120	15.8740	19.3480	11.1100	8.6700
Panel D: 125 trading day annual returns when Wind = 1526																		
Average return (e-3)	-1.010	0.6068	-0.2803	-0.1413	0.2815	-0.0458	-0.0879	-0.1362	-2.3740	0.5177	-0.9037	-0.3194	0.3513	-0.5758	-0.1563	-0.7211	-0.3800	0.4594
Standard deviation	0.0059	0.0123	0.0126	0.0134	0.0054	0.0106	0.0086	0.0116	0.0077	0.0123	0.0124	0.0135	0.0162	0.0055	0.0105	0.0087	0.0116	0.0134
t-Statistic	-1.8928	0.5485	-0.2486	-0.1174	0.5757	-0.0483	-0.1133	-0.1309	-3.4428	0.4683	-0.8114	-0.2636	0.2419	-1.1600	-0.1658	-0.9234	-0.3639	0.3807
Minimum	-0.0344	-0.0425	-0.0485	-0.0485	-0.0222	-0.0342	-0.0342	-0.0342	-0.0481	-0.0425	-0.0485	-0.0485	-0.0485	-0.0222	-0.0342	-0.0342	-0.0342	-0.0500
Median	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0007	0.0000	0.0000	0.0000	0.0000	0.0000
Maximum	0.0139	0.0729	0.0729	0.0729	0.0205	0.0286	0.0514	0.0514	0.0128	0.0729	0.0673	0.0729	0.0729	0.0186	0.0286	0.0472	0.0514	0.0472
Skewness	-3.2270	1.2520	1.0081	0.8006	0.2011	-0.2784	1.3036	1.0953	-3.5393	1.2459	0.5768	0.7658	0.3998	-0.9790	-0.2659	0.8038	1.0877	0.1056
Kurtosis	18.3810	13.3040	14.0430	10.9930	9.1660	4.9020	15.3450	8.3650	18.1110	13.3580	12.4700	10.8130	6.2620	9.4310	4.9950	13.0140	8.3740	8.5340

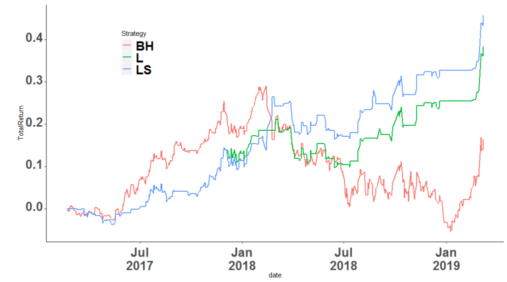
Table 4.9: Daily trading profits of machine learning based trading strategies with multi-day forecast within one year

	Before transaction costs								After transaction costs									
	ETF50				ETF300				ETF50					ETF300				
	RF		MLP		RF		MLP		RF		MLP		BH	RF		MLP		BH
	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS		L	LS	L	LS	
Panel A: 250 trading day daily returns when Wind = 1401																		
Average return (e-3)	1.1441	1.4995	-0.2479	0.1024	1.4154	1.9973	-0.1717	-0.5291	1.1102	1.2438	-0.4880	-0.2560	-0.3510	0.5873	0.7409	-0.4004	-0.7578	-0.4150
Standard deviation	0.0075	0.0090	0.0133	0.0137	0.0105	0.0116	0.0131	0.0137	0.0092	0.0097	0.0133	0.0136	0.0146	0.0078	0.0079	0.0130	0.0137	0.0147
t-Statistic	2.4030**	2.6130*	-0.2964	0.119	2.1290*	2.7140**	-0.2087	-0.6127	1.8940*	2.0260*	-0.5841	-0.2986	-0.3807	1.1840	1.4800+	-0.4883	-0.8804	-0.4458
Minimum	-0.0485	-0.0485	-0.0485	-0.0485	-0.0527	-0.0301	-0.0527	-0.0527	-0.0485	-0.0485	-0.0485	-0.0485	-0.0485	-0.0527	-0.0342	-0.0527	-0.0527	-0.0514
Median	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0001	0.0000	0.0000	0.0000	0.0000	-0.0012
Maximum	0.0729	0.0729	0.0729	0.0729	0.0682	0.0682	0.0682	0.0682	0.0729	0.0729	0.0729	0.0729	0.0729	0.0682	0.0682	0.0682	0.0682	0.0682
Skewness	0.1615	0.1301	0.3713	0.3217	0.4216	0.8057	0.3253	0.2089	0.1557	0.1159	0.3829	0.3378	0.3326	0.4035	0.7358	0.3284	0.2152	0.3212
Kurtosis	2.9810	2.8370	4.3530	3.7070	4.5870	3.9680	4.6310	3.7730	2.9790	2.9080	4.3960	3.7620	5.7300	4.5660	4.0740	4.7270	3.8530	27.6100
Panel B: 500 trading day daily returns when Wind = 1151																		
Average return (e-3)	1.2360	1.3650	0.3570	0.3570	1.1140	1.5400	0.1001	0.0794	0.7689	0.9148	0.3457	0.3457	0.2211	0.7562	1.1357	0.0664	0.0457	0.1049
Standard deviation	0.0075	0.0082	0.0120	0.0120	0.0072	0.0083	0.0113	0.0117	0.0076	0.0082	0.0120	0.0120	0.0111	0.0073	0.0082	0.0117	0.0117	0.0120
t-Statistic	2.5880**	2.4930**	0.6606	0.6606	3.7770***	2.9640**	0.1898	0.1503	2.2540*	2.4660**	0.6397	0.6397	0.1962	2.3130*	3.0740**	0.1261	0.0866	0.1962
Minimum	-0.0425	-0.0425	-0.0485	-0.0485	-0.0527	-0.0527	-0.0527	-0.0527	-0.0481	-0.0481	-0.0485	-0.0485	-0.0486	-0.0527	-0.0527	-0.0527	-0.0527	-0.0528
Median	0.0000	0.0000	0.0004	0.0004	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0004	0.0004	0.0000	0.0000	0.0000	0.0000	0.0005
Maximum	0.0729	0.0729	0.0729	0.0729	0.0682	0.0682	0.0682	0.0682	0.0729	0.0729	0.0729	0.0729	0.0729	0.0682	0.0682	0.0682	0.0682	0.0682
Skewness	0.4834	0.5217	0.0817	0.0817	0.1475	0.1677	0.0872	0.0908	0.4257	0.4726	0.0844	0.0844	0.0517	0.1156	0.0991	0.0907	0.0943	0.0414
Kurtosis	6.1100	5.4040	4.0540	4.0540	6.0440	5.4630	4.7380	4.6960	6.3530	5.4410	4.0520	4.0520	3.9690	6.1770	5.6330	4.7870	4.7440	4.2840
Panel C: 250 trading day annual returns when Wind = 1401																		
Annual return	0.2883	0.3779	-0.0624	0.0258	0.3567	0.5033	-0.0432	-0.1333	0.2798	0.3134	-0.1229	-0.0645	-0.0883	0.1480	0.1867	-0.1008	-0.1909	-0.1045
Excess return	0.2708	0.3604	-0.0799	0.0083	0.3392	0.4858	-0.0607	-0.1508	0.2623	0.2959	-0.1404	-0.0820	-0.1058	0.1305	0.1692	-0.1183	-0.2084	-0.1220
Standard deviation	0.1195	0.1440	0.2120	0.2177	0.1669	0.1847	0.2085	0.2189	0.1471	0.1541	0.2118	0.2173	0.2311	0.1245	0.1257	0.2078	0.2182	0.2335
Downside deviation	0.0492	0.0702	0.1488	0.1500	0.0980	0.1100	0.1445	0.1567	0.0859	0.0906	0.1507	0.1528	0.1619	0.0843	0.0830	0.1460	0.1581	0.1635
Sharpe ratio	2.2660	2.5020	-0.3772	0.0381	2.0330	2.6300	-0.2915	-0.6890	1.7820	1.9200	-0.6633	-0.3773	-0.4579	1.0480	1.3460	-0.5696	-0.9554	-0.5225
Sortino ratio	5.4950	5.1300	-0.5373	0.0553	3.4600	4.4150	-0.4206	-0.9624	3.0520	3.2630	-0.9320	-0.5368	-0.6535	1.5470	2.0370	-0.8108	-1.3182	-0.7462
Maximum drawdown	0.1047	0.1047	0.1047	0.1047	0.0910	0.0910	0.0910	0.0923	0.1047	0.1047	0.1047	0.1047	0.2120	0.0910	0.0910	0.0923	0.1089	0.2453
Panel D: 500 trading day annual returns when Wind = 1151																		
Annual return	0.2883	0.3678	0.0899	0.0899	0.2808	0.3882	0.0252	0.0200	0.1938	0.2305	0.0871	0.0871	0.0557	0.1906	0.2862	0.0167	0.0115	0.0264
Excess return	0.2708	0.3504	0.0724	0.0724	0.2633	0.3707	0.0077	0.0025	0.1763	0.2130	0.0696	0.0696	0.0382	0.1731	0.2687	-0.0007	-0.0059	0.0089
Standard deviation	0.1200	0.1309	0.1906	0.1906	0.1156	0.1319	0.1861	0.1864	0.1210	0.1316	0.1907	0.1907	0.1919	0.1160	0.1311	0.1858	0.1860	0.1897
Downside deviation	0.0649	0.0691	0.1328	0.1328	0.0625	0.0682	0.1309	0.1312	0.0712	0.0744	0.1328	0.1328	0.1352	0.0672	0.0730	0.1309	0.1312	0.1340
Sharpe ratio	2.4510	2.4940	0.3800	0.3800	2.2770	2.8100	0.0415	0.0134	1.4560	1.6180	0.3651	0.3651	0.1992	1.4910	2.0490	-0.0041	-0.0321	0.0470
Sortino ratio	4.5250	4.7240	0.5455	0.5455	4.2090	5.4370	0.0590	0.0191	2.4740	2.8630	0.5240	0.5240	0.2827	2.5740	3.6810	-0.0058	-0.0456	0.0666
Maximum drawdown	0.1047	0.1047	0.1047	0.1047	0.0910	0.0910	0.0910	0.0910	0.1047	0.1047	0.1047	0.1047	0.2895	0.0910	0.0910	0.0910	0.0910	0.2232

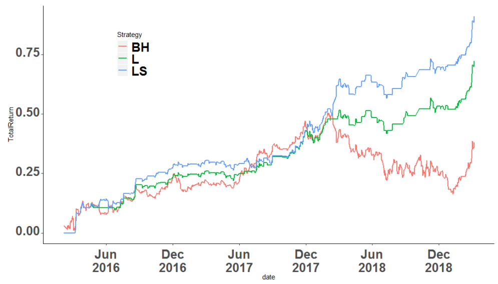
Table 4.10: Daily and annualised trading profits of machine learning based trading strategies with multi-day forecast in one and two years



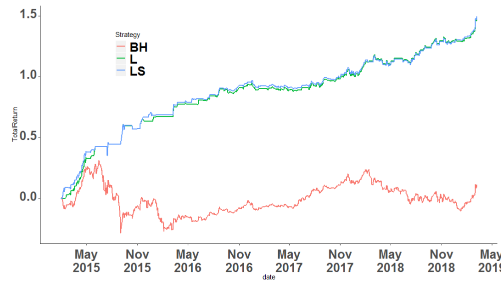
(a) ETF50 in 250 days



(b) ETF50 in 500 days



(c) ETF50 in 750 days



(d) ETF50 in 1000 days

Figure 4.2: Cumulative returns (after transaction costs) for multiple days random forest based strategy with forecasts for ETF50.

forest forecast based strategies are further evaluated in the following Figure 4.2. Some results are found below.

Stable and continuous positive returns can be generated by random forest. Random forest based trading strategies produce superior results in terms of profitability even when stock markets are in bear periods, e.g. the period after Jan. 2018[13]. In the same period, all the other trading strategies developed in this thesis cannot generate positive trading performances.

As a benchmark for comparison, the corresponding performance for buy-and-hold strategy is also included. During the period from 21 February 2017 until 8 March 2019, ETF50 and ETF300 increased by 11.69% and 8.36%, which correspond to 5.57% and 2.64% at an annualised rate. This implies that buy-and-hold strategy is profitable in the latest 500 days. By comparison of trading strategies

[13]This is the period of the last 250 days.

using our algorithms (see Figures 4.2(b) and 4.5(b)) in the same period, the total return on 8 March 2019 for ETF50 reaches 40% while 60% for ETF300. That is to say, the result for ETF300 shows a return 5 times greater than the one obtained by the buy-and-hold strategy.

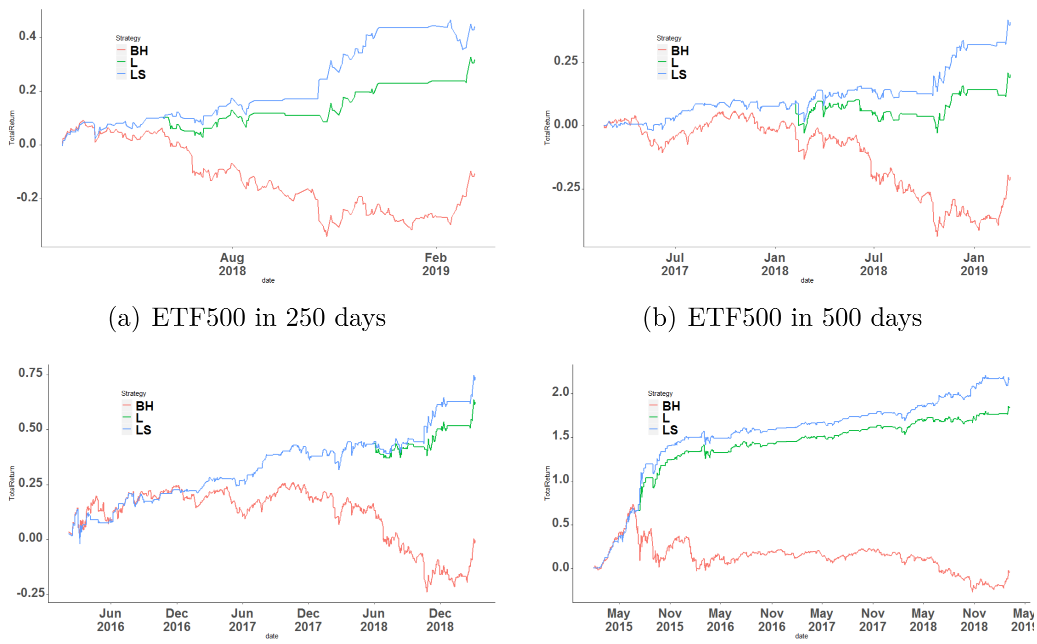
4.7.3 Robustness check for trading using the multi-day random forest prediction

To test if multi-day random forest prediction based trading strategies are able to obtain consistently profitable returns for a new asset for all time, we expand the assets and trading time periods used before. Specifically, the performances of the cumulative returns after transaction costs on ETF500 are checked under the same trading strategies used for the task of multi-day ahead prediction. The trading performances of all three indices are checked over another two time periods, the most recent 3 and 4 years[14].

As done in the performance analysis of ETF50 and ETF300, ETF500 is tested based on using multi-day forecasts version. According to the results for the cumulative return in Figure 4.3, when compared with the red curve (buy-and-hold strategy), the blue curve ('LS' strategy) and green curve ('L' strategy) gain significantly greater profits. Especially, 'LS' strategy outperforms 'L' strategy when the stock price is in a downtrend. These results are consistent with the observed results from ETF50 and ETF300 in Figures 4.2 and 4.5.

In addition, break-even analysis is used to further validate the profitability of multi-day random forest forecast based trading strategies in instances of assets and time periods. Break-even costs are computed based on the most recent 250, 500, 750 and 1000 out-of-sample forecasts in the test sets, respectively. Under

[14]The most recent 750 and 1000 days, separately.



(a) ETF500 in 250 days (b) ETF500 in 500 days

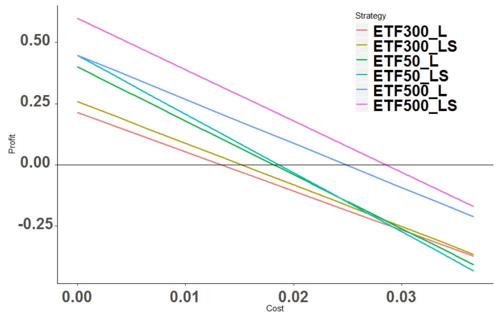
(c) ETF500 in 750 days (d) ETF500 in 1000 days

Figure 4.3: Cumulative returns (after transaction costs) for multiple days random forest forecast based trading strategies for ETF500.

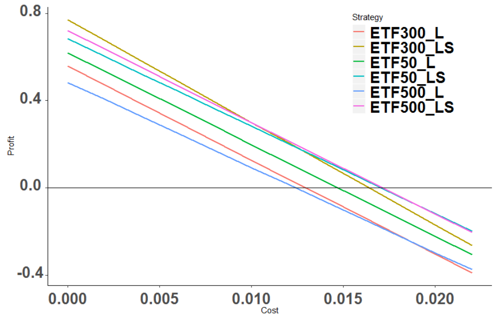
the assumption that the transaction costs for opening a long position and a short position are the same, the break-even transaction costs in all considered cases are around 0.01500- 0.04000 per trade in Figure 4.4. In reality, for all China ETFs, the average transaction costs on each trade are 0.00564 for long position and 0.00696 for short position, separately. The transaction costs in the real world are smaller than the break-even costs, and this implies that the profitability of 4 day ahead random forest forecast based strategies is high over the last 4 years.

Figures 4.2 to 4.5 show the results for profit after transaction costs by applying random forest across all forecast horizons, i.e. the last 250, 500, 750 and 1000 days. For all indices, two periods come to our attention. First, we observe strong returns between June 2015 and February 2016, i.e., the period of 'Chinese stock market turbulence'[15]. Second, the year of 2018 witnessed high returns, and

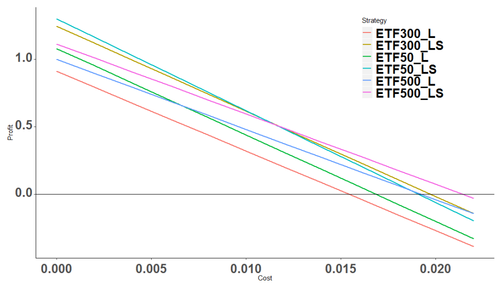
[15]The 'Chinese stock market turbulence' can be found in Bloomberg, refer to:



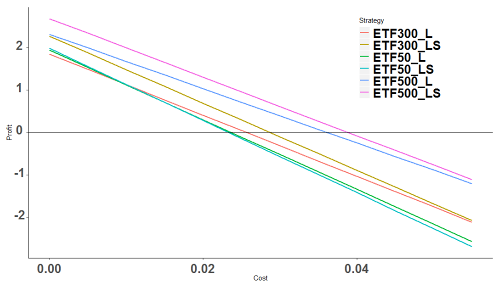
(a) 250 days



(b) 500 days

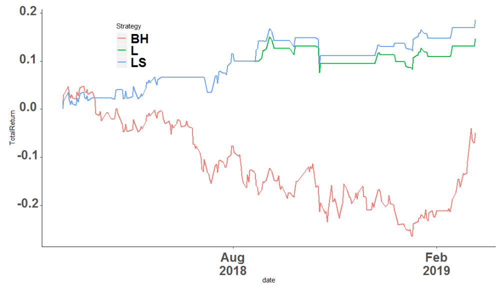


(c) 750 days

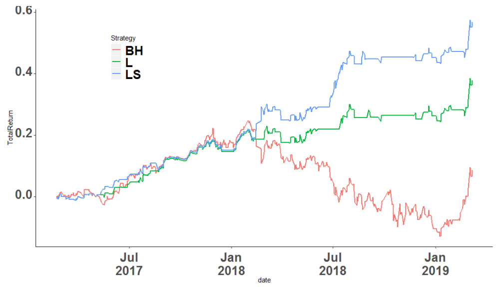


(d) 1000 days

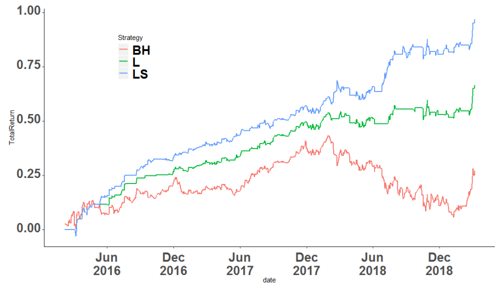
Figure 4.4: Break-even costs for cumulative returns (after transaction costs) for multi-day random forest forecast based trading strategies for ETF50, ETF300 and ETF500 during the last 250, 500, 750 and 1000 days.



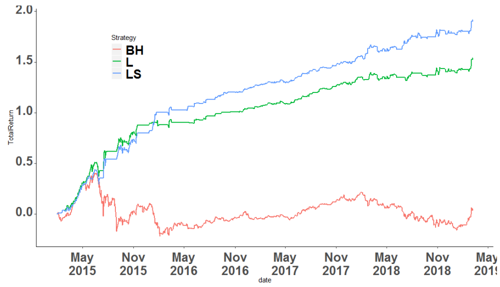
(a) ETF300 in 250 days



(b) ETF300 in 500 days



(c) ETF300 in 750 days



(d) ETF300 in 1000 days

Figure 4.5: Cumulative returns (after transaction costs) for multiple days random forest forecast based trading strategies for ETF300.

coincides with the time of US trade war, when all major Chinese stock indices lost more than 25%[16]. Both these results verify that multi-day random forest predictions based trading strategies yield high excess returns during a financial crisis.

In general, all results for total returns in this section show positive and upward trends. Therefore, for investors who select the tested trading rule with parameters setting $p \geq 3$, $a = 1$, $b = 3$ and $q = 0$, the proposed 4-day forecast based strategies with random forest are robust in the presence of costs and are robust over asset and time.

https://www.bloomberg.com/news/articles/2015-07-27/chinese-stock-index-futures-drop-before-industrial-profits

[16]For more details, refer to CNBN news in 2018 with link: https://www.cnbc.com/2018/12/31/china-markets-2018-performance-was-worst-in-a-decade.html

4.7.4 Variations of the multi-day machine learning prediction based trading

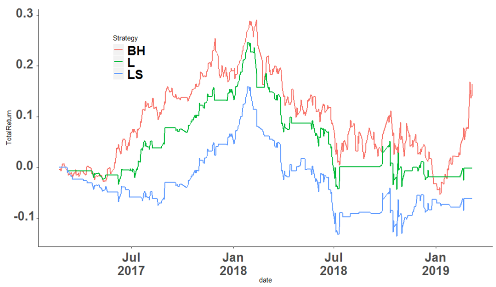
By showing some possible trading system variations, this section presents some evidence to support the view that 4-day random forest prediction based trading stra is the best trading strategy. Three relevant changes in the previous trading system are described below. First, time horizon in class label version, i.e. class label 2 and 3 are used as opposed to 4 class label version which has already been explored in the previous analysis. New trading rules are redefined accordingly. However, the aim of the new trading rules is still reducing the number of trades. The second relevant change deals with the use of another algorithm - MLP. The third relevant change consisted of introducing another trading rule to further reduce the number of trades by trading only when strong buy and sell signals are generated.

In this section, for the first change, the possible cases for time horizon less than 4 but more than 1 are analysed. Since there are many possible cases for the second and third change, only one possible case for each of the last two changes is shown as an example.

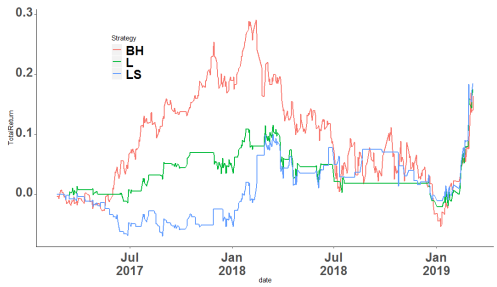
Time horizon in class label

One practical issue with multi-day forecasts based trading is how many future observations should be used in the trading system. The number of future observations used in the estimation is referred to as the time horizon in the class labels. In the first variation, we consider the time horizon of 2 and 3 days.

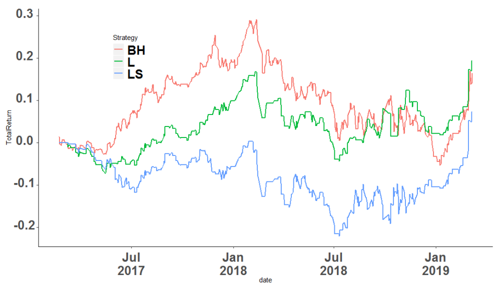
When the next 2 days' price movements are considered (n=2), the possible class label (y) is 2 for a buy signal, 1 for a neutral signal and 0 for a sell signal. The new trading rules in this case is described below. The long position is entered



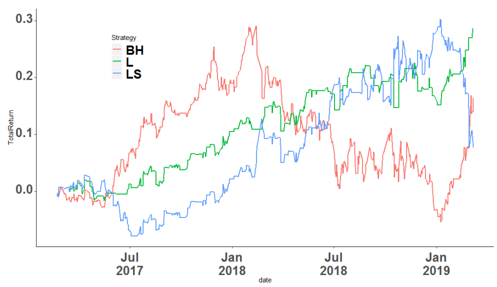
(a) ETF50 for 2 classes



(b) ETF300 for 2 classes



(c) ETF50 for 3 classes



(d) ETF300 for 3 classes

Figure 4.6: Cumulative returns (after transaction costs) for multiple days random forest forecast based trading strategies with different class labels in the last 500 days.

when we predict $y_{predict} = 2$, and the long position is held till $y_{predict} = 0$ at the first time. The short position is triggered when $y_{predict} = 0$, and the short position is closed if $y_{predict} = 1$.

Similarly, the case when $n = 3$ is a 3 day price movement prediction with 4 classes - 3 for a strong buy signal, 2 for a weak buy signal, 1 for a weak sell signal, and 0 for a strong sell signal. Following the 'L' strategies, we purchase the stock when $y_{predict} > 1$ and sell the stock when $y_{predict} = 1$. In 'LS' strategy, the short position is opened if $y_{predict} = 0$ and is closed when $y_{predict} = 1$.

The performance of the multi-day forecast based strategy under different classification settings is shown in Figure 4.6.

All random forest based trading strategies when $n = 2$ in Figures 4.6(a) and 4.6(b) cannot beat buy-and-hold strategy. Therefore, we cannot choose $n = 2$.

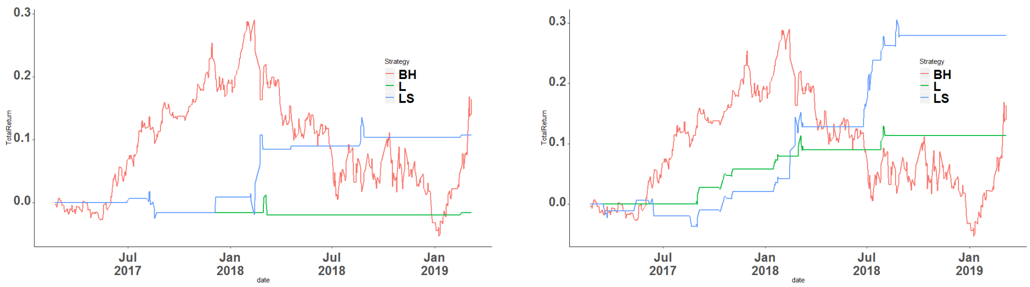
Although the final returns for 'LS' strategies are around 0.3 when $n = 3$ in Figures 4.6(c) and 4.6(d), these results still cannot beat the case with the final return 0.6 when $n = 4$ in Figures 4.2. It's worth mentioning that the trading rules for 3-day forecast based trading system ($n = 3$) and 4-day forecast based trading system ($n = 4$) are same, and the only difference is that 4-day forecast based trading strategy has one more class. In other words, with a longer time horizon in class label, 4-day forecast based trading strategy enhances the total returns, compared to 3-day forecast based trading strategy. From those empirical results, it may suggest that large n improves the trading performance in terms of the cumulative return. However, if n is very large, more classes are created, and the data will become more imbalanced as it is difficult to observe an example with all upward movements in 5 consecutive days.

In conclusion, for investors with the trading rule that $p \geq 3$, $a = 1$, $b = 3$ and $q = 0$, the trading performance of multi-day random forest prediction based strategy is sensitive to the choice of time horizon in the class label. According to the previous comparisons, applying the new trading rules on 2 classes and 3 classes obtain poor results, we, therefore, select $n = 4$ for multi-day forecast based trading strategy for investors who set $p \geq 3$, $a = 1$, $b = 3$ and $q = 0$.

Algorithms chosen

As shown in Table 4.10, random forest is a better model relative to MLP from the perspective of t-Statistics, Sharpe ratio and Sortino ratio.

In almost all cases, the t-Statistics for random forest are statistically significant while none of t-Statistics for MLP are statistically significant. Although in some cases, for ETF50 during 500 trading days, MLP can beat buy-and-hold strategy considering transaction costs, the Sharpe ratios in this case are still less



(a) ETF50 in 500 days (b) ETF300 in 500 days

Figure 4.7: Cumulative returns (after transaction costs) for multiple days random forest forecast based trading strategy with an alternative trading rule in the last 500 trading days.

than 1. In contrast, the Sharpe ratios for random forest in all cases are above 1. This implies that random forest can generate high risk-adjusted returns. In majority cases including ETF50 and ETF300, MLP cannot generate positive Sortino ratios after transaction costs during the forecast comparison periods (250 and 500 days). For ETF50 and ETF300, random forest based trading strategies can obtain Sortino ratios with values larger than 1.5 after transaction costs during the same periods. Therefore, random forest outperforms MLP when generating a signal of the next 4 days' price movements in the case of $p \geq 3$, $a = 1$, $b = 3$ and $q = 0$.

Alternative trading rule

We also consider an alternative trading rule for risk averse investors to take multi-day prediction based strategy. As before, $a = 1$, $b = 3$, but $p = 4$ instead of $p \geq 3$. In this strategy, we only trade when strong buy signals are generated.

Figure 4.7 demonstrates that almost all trades are profitable. However, from the perspective of total return, the majority of active trading strategies cannot beat buy-and-hold strategy as there are few trades. Consequently, there is a trade-off between profits for each trade and the number of trades.

(Wind, Test)	ETF50		ETF300	
	RF_{one}	$\mathrm{RF}_{multiple}$	RF_{one}	$\mathrm{RF}_{multiple}$
(1401,250)	0.536	0.580	0.568	0.544
(1351,300)	0.560	0.587	0.480	0.553
(1151,500)	0.558	0.562	0.570	0.570

Notes: 'RF_{one}' and '$RF_{multiple}$' represent 'one-day prediction based random forest model' and 'multi-day prediction based random forest model'.
$N = 10$ for RF_{one} while $N = 100$ for $RF_{multiple}$.
The multi-day predictions based model considers 5 classes while one-day prediction based model takes account 2 classes.

Table 4.11: Comparison of the model accuracy in random forest based trading strategies with one-day forecast and multi-day forecasts.

Therefore, the obtained results in this section confirm that 4 day prediction based trading strategy by random forest generates a positive return continuously after transaction costs. Moreover, under two tested trading rules-Option 1 is $p \geq 3$, $a = 1$, $b = 3$ and $q = 0$; Option 2 is - $p = 4$, $p = 1$, $b = 3$, and $q = 0$, using the 4 day random forest prediction based trading strategy is robust in terms of assets and time periods.

4.7.5 Performance comparison

Trading with one day forecast vs. Trading with multiple days forecast

As previously mentioned, before transaction costs, one day prediction based strategies are unprofitable during the last 250 days but profitable over the last 500 days. When incorporating transaction costs into strategies, one-day prediction based strategies are not able to gain profits in all cases[17]. As for multi-day prediction based strategies, random forest based trading strategies are profitable after transaction costs during both test periods i.e. 250 days and 500 days[18]. It can be concluded that for the random forest model, using multi-day forecasts generates a higher risk-adjusted return than the one day ahead forecast for all the assets and different trading time periods.

[17]Refer to section 4.7.1 for more details
[18]Refer to section 4.7.2 for more details

	ETF50				ETF300			
	RF_{one}		$RF_{multiple}$		RF_{one}		$RF_{multiple}$	
Day	L	LS	L	LS	L	LS	L	LS
250	38	140	22	24	38	141	17	16
500	92	133	42	40	87	130	43	47

Notes: 'RF_{one}' and '$RF_{multiple}$' represent 'RF based trading strategies using one day ahead forecast' and 'RF based trading strategies using multiple days ahead forecast'.
'L', 'LS', and 'BH' stand for 'Long' strategies, 'Long and Short' strategies, and 'Buy and Hold' strategies.

Table 4.12: Comparison of the number of trades in random forest based trading strategies with one-day forecast and multi-day forecasts.

The random forest models for one-day prediction and multi-day predictions with three test sets (250, 300, and 500 days) are summarized in Table 4.11. Nearly all multi-day prediction based random forest model has a higher accuracy than one-day prediction based random forest model in all three cases.

The advantage of trading strategies with multi-day forecasts compared to that with one day forecast is that the number of trades in multi-day forecast strategy is reduced by half for 'L' strategy and by more than two-thirds for 'LS' strategy according to Table 4.12. As a result, transaction costs for multi-day prediction strategy during the trading periods should be highly reduced.

The three results of this section are that multi-day random forest forecast based strategy is better than one-day random forest forecast based strategy from the perspective of risk-adjusted returns and model accuracy; the number of trades for multi-day random forest forecast based strategy is highly reduced compared with the strategy using one day forecast. These are encouraging results and it may suggest that the multiple days random forest based strategies are able to reduce some losing trades.

Machine learning based trading strategies vs. MACD trend following trading strategies

This section compares the benchmark strategy (MACD trend following strategy) to the best machine learning based trading strategies to demonstrate the performance of machine learning model used in the trading system to improve the investment decisions (see Table 4.13).

	ETF50						ETF300					
	MACD		RF_{one}		$RF_{multiple}$		MACD		RF_{one}		$RF_{multiple}$	
	L	LS	L	LS	L	LS	L	LS	L	LS	L	LS
Panel A: 250 trading day annual returns												
Annual return	-0.1578	-0.15787	-0.2266	-0.4982	0.2798	0.3134	-0.2126	-0.2126	-0.2172	-0.3869	0.1480	0.1867
Excess return	-0.1753	-0.1753	-0.2441	-0.5157	0.2623	0.2959	-0.2301	-0.2301	-0.2348	-0.44044	0.1305	0.1692
Standard deviation	0.1815	0.1815	0.1733	0.3002	0.2140	0.2203	0.1471	0.1541	0.1928	0.2562	0.1245	0.1257
Downside deviation	0.1314	0.1314	0.1258	0.2532	0.0859	0.0906	0.1345	0.1345	0.1215	0.1737	0.0843	0.0830
Sharpe ratio	-0.9657	-0.9657	-1.4083	-1.7178	1.7820	1.9200	-1.2603	-1.2603	-1.2178	-1.5784	1.0480	1.3460
Sortino ratio	-1.3340	-1.3340	-1.9405	-2.0364	3.0520	3.2630	-1.7100	-1.7100	-1.9327	-2.3280	1.5470	2.0370
Maximum drawdown	0.4817	0.4739	0.5090	0.5980	0.1047	0.1047	0.6125	0.5413	0.3607	0.3207	0.0910	0.0910
Panel B: 500 trading day annual returns												
Annual return	-0.1025	-0.1787	-0.1515	-0.4982	0.1938	0.2305	-0.0947	-0.1445	-0.1314	-0.1968	0.1906	0.2862
Excess return	-0.1200	-0.1962	-0.1690	-0.2902	0.1763	0.2130	-0.1122	-0.1620	-0.1498	-0.2143	0.1731	0.2687
Standard deviation	0.1433	0.1329	0.1324	0.1949	0.1210	0.1316	0.1415	0.1296	0.1286	0.2002	0.1160	0.1311
Downside deviation	0.1056	0.1003	0.0971	0.1395	0.0712	0.0744	0.1040	0.0965	0.0914	0.1346	0.0672	0.0730
Sharpe ratio	-0.8381	-1.4768	-1.2766	-1.4894	1.4560	1.6180	-0.7938	-1.2502	-1.1589	-1.0706	1.4910	2.0490
Sortino ratio	-1.1368	-1.9556	-1.7411	-2.0798	2.4740	2.8630	-1.0788	-1.6790	-1.6285	-1.5920	2.5740	3.6810
Maximum drawdown	0.4831	0.4129	0.5110	0.4230	0.1047	0.1047	0.5192	0.4561	0.3168	0.3354	0.0910	0.0910

Notes: 'RF_{one}' and '$RF_{multiple}$' represent 'random forest based trading strategies using one day ahead forecast' and 'random forest based trading strategies using multiple days ahead forecast'.
'L', 'LS', and 'BH' stand for 'Long' strategies, 'Long and Short' strategies, and 'buy-and-hold' strategies.
The sizes for test set in Panel A and Panel B are 300 days and 600 days, individually.
Both Panel A and Panel B analyse the parameter settings with n = 10 for RF_{one} and n = 100 for $RF_{multiple}$.
We assumption that one year contains 252 trading days.
Maximum drawdown is defined as the peak-to-tough decline over the duration of a strategy's test period.

Table 4.13: Comparison of annualised trading profits of active trading strategies for ETF50 and ETF300 after transaction costs.

The results show that MACD trend following strategy is unprofitable in terms of Sharpe ratio (ranges from -1.4768 to -0.7938 after transaction costs) and Sortino ratio, ranges from -1.9556 to -1.0788 after transaction costs. One day random forest forecast based strategy cannot improve the Sharpe ratio (ranges from -1.7178 to -1.0706) and Sortino ratio (ranges from -2.3280 to -1.5920), compared to MACD strategy.

As expected, by incorporating multi-day prediction information into random forest based trading strategy, it leads to the improvements in the results as indicated by Table 4.13. Multi-day forecast based trading system produces better performances than one-day forecast based trading system in terms of Sharpe ratio (ranges from 1.0480 to 2.0490) and Sortino ratio (ranges from 1.5470 to 3.6810). It can be implied that using multiple days forecast is capable of gaining the best results of Sharpe ratio and Sortino ratio.

In the case of maximum drawdown, the lower measures of maximum drawdown also identify that the amount of risk incurred by the multi-day forecast based trading strategy becomes smaller than the other strategies.

Therefore, the empirical results show that multi-day random forest forecast based strategy is superior compare to the other methods, since it generates the highest excess returns, around 21.11% on average, the highest Sharpe and Sortino ratios, as well as the lowest maximum drawdown.

4.8 Discussion

This section summarises the major empirical results that have been observed during the different experiments and back-testings.

A comparison between trend following strategies and machine learning based trading strategies is presented. Comparing the cumulative return plots for the MACD trend following strategy (see Figure 2.10) with the multiple days random forest prediction based strategy (see Figures 4.3 to 4.5), we find that multi-day random forest forecast based strategy outperforms the trend following strategies. According to the results of our previous experiments, which are not presented here, not only the MACD trend following strategies, but also other trend following strategies, such as moving average and relative strength index, fail to outperform the random forest based trading strategies.

There are at least two reasons why trend following strategies fail to beat machine learning based strategies. Trend following strategies solely consider one technical indicator. Using single technical indicator to judge the price movement is risky, and it may generate wrong signals since the single technical indicator may be easily influenced by the noisy information. In contrast, machine learning based trading strategies consider other technical indicators. The second reason is that trend following strategies trade based on lagging technical indicators while machine learning based trading strategy is a forecasting system to predict the future. That is to say, trend following strategy can be considered as a rule engine automating trading decisions only based on the technical signals. However, machine learning based trading strategy is another rule engine that uses technical signals and predictive signals mapping into future price movements and hence producing signals.

The comparisons of different machine learning models are discussed. In this study, Tables 4.7, 4.8 and 4.10 demonstrate the performances of machine learning classifiers used in this thesis. After transaction costs, SVM cannot generate positive Sharpe ratios in all cases, logistic regression and MLP can obtain positive Sharpe ratios in some cases. These results may be due to the disadvantages of each machine learning model. Different classifiers may fit different datasets. SVM doesn't perform well when the dataset has more noise, which means that if data is not properly separated and the target classes overlap, SVM performs poorly. Moreover, decision trees are also affected by the noise. Logistic regression may require a fairly large training set before it can make accurate predictions outside of the training set. MLP and LSTM cannot predict one-day ahead price movement with high accuracy according to the experiments conducted but not included in this thesis. Both MLP and LSTM contain a large number of parameters, and it is difficult to get a robust accuracy under all possible combinations of parameters.

In practical applications, we recommend using the 4-day random forest prediction based trading strategies to trade China ETFs. One potential challenge is the use of close-to-close log returns, however, this implies that prediction and order execution should be very fast or almost simultaneously. It is the case that the same profits of the developed trading strategies are not guaranteed in a one-to-one fashion. Therefore, slippage costs have been considered in our strategies to approximate the potential costs. Moreover, there is still a great potential to make profits since the break-even costs for multi-day random forest forecast based trading strategies on average is high, refer to Figure 4.4. In general, our 4-day random forest prediction based trading strategies are capable of generating positive returns when the total transaction costs are below 0.015 - 0.040 per trade.

In this chapter, the performances of machine learning based trading strategies were examined. After the robustness check and model comparisons, it demonstrates that the most successful trading strategy in this study is 4-day forecast based trading strategies using random forest algorithm.

Chapter 5

Conclusion

This chapter summarises the conclusions related to three aims. Furthermore, other machine learning based trading strategies can be developed in the future based on this study.

The primary aim of this study was to develop profitable trading strategies taking into account transaction costs. Chart pattern, trend following and machine learning based trading strategies[1] are developed as potential candidates to achieve the aim. The results show that only multi-day random forest forecast based trading strategies yield positive risk-adjusted returns after transaction costs. It is worthwhile to point out that the presence of profitability in multi-day random forest forecast based trading strategies is robust with respect to different assets[2] and trading periods[3]. Therefore, multi-day random forest forecast based trading strategies are the only profitable and robust strategies.

The second aim was to compare machine learning based trading strategies with chart pattern strategies and MACD trend following strategies. As previ-

[1] Machine learning based trading strategies are composed of one-day machine learning forecast based trading and multi-day machine learning forecast based trading.

[2] Assets include China ETFs, i.e. ETF50, ETF300 and ETF500.

[3] Trading periods consist of 250, 500, 750 and 1000 days.

ously mentioned, the limitation of chart pattern strategy is that it heavily relies on the experts' experiences. If only one single technical indicator such as MACD is considered to generate trading signals, it may not obtain positive risk-adjusted returns. Without considering transaction costs, the machine learning based trading strategies with one day prediction yields positive returns. However, the one day prediction based trading strategies considered in this thesis fail to generate positive returns after transaction costs. As for robustness check with respect to trading period, chart pattern, MACD trend following and machine learning with one day forecast are not profitable during market recessions. After analysing the limitations of all trading strategies mentioned above, an improved trading strategy - multi-day machine learning forecast based trading strategy - is developed. The result shows that multi-day machine learning forecast based trading strategies are profitable in terms of risk-adjusted returns. It is verified that the superiority of 4-day forecast based trading strategies with random forest over chart pattern and trend following strategies is most striking when transaction costs and trading periods are taken into account. In conclusion, not all machine learning trading strategies are profitable, and the trading profits generated by machine learning based strategies depend on the way the target variable is derived. That is, multi-day ahead forecasting techniques should be considered for profit generation.

The third aim of this thesis was to compare the developed machine learning based trading strategies with two benchmarks, namely buy-and-hold strategy and AR model. The recent performance of the best machine learning based trading strategies shows that it is superior to the buy-and-hold strategy and time series (AR) based strategy even when the stock market indices are declining in a recession, as measured by the Sharpe ratios and cumulative returns. It appears that passive trading and solely adopting historical stock price information to generate trading signals do not constitute robust investment decisions. However, this research verifies that machine learning method has superiority in predicting trading signals based on the multi-day forecasting techniques. With proper training of the models, machine learning is able to successfully assist investors to navigate the decision-making activities.

Overall, this thesis focuses on selecting the best investment times to trade on China ETFs. The new insight into the trading signal generation is that the machine learning models are trained with the time sequence of the multi-day price movements. It turns out that the multi-day machine learning based strategies have improved the trading performances during the last 4 years of backtesting period. The main contribution to the existing literature is the design of the multi-day target outputs for the machine learning based trading strategies, which offer smoother returns than the methods based on a single day predictions considered in the existing literature.

Future research directions can be considered as follows. Future studies can focus on a performance-weighted ensemble since the simple majority voting does not perform well. To design an effective ensemble classifier, the base classifiers will be selected by some restrictions such as diversity and accuracy to obtain final predictions. This thesis can be part of a larger algorithmic trading system which may include risk management systems in place.

One of the challenges of forecasting stock price movement is that of the information quality of features utilisation in the model. It is possible to improve the forecast of asset returns by incorporating a broader set of predictors. Related predictors include data sources from Baidu search trends and online news. With the help of natural language processing models, events and news can be added as features into machine learning models.

Bibliography

Andrada-Félix, J., Fernández-Rodríguez, F., 2008. Improving moving average trading rules with boosting and statistical learning methods. Journal of Forecasting 27 (5), 433–449.

Appel, G., 2005. Technical analysis: power tools for active investors.

Arévalo, R., García, J., Guijarro, F., Peris, A., 2017. A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting. Expert Systems with Applications 81, 177–192.

Ballings, M., Van den Poel, D., Hespeels, N., Gryp, R., 2015. Evaluating multiple classifiers for stock price direction prediction. Expert Systems with Applications 42 (20), 7046–7056.

Basak, S., Kar, S., Saha, S., Khaidem, L., Dey, S. R., 2019. Predicting the direction of stock market prices using tree-based classifiers. The North American Journal of Economics and Finance 47, 552–567.

Beyaz, E., Tekiner, F., Zeng, X.-j., Keane, J., 2018. Comparing technical and fundamental indicators in stock price forecasting. 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 1607–1613.

Bo, L., Linyan, S., Mweene, R., 2005. Empirical study of trading rule discovery in china stock market. Expert Systems with Applications 28 (3), 531–535.

Booth, A., Gerding, E., Mcgroarty, F., 2014. Automated trading with performance weighted random forests and seasonality. Expert Systems with Applications 41 (8), 3651–3661.

Bowen, D., Hutchinson, M. C., O'Sullivan, N., 2010. High frequency equity pairs trading: transaction costs, speed of execution and patterns in returns. Journal of Trading 5 (3), 31–38.

Breiman, L., 1999. 1 random forests–random features.

Brock, W., Lakonishok, J., LeBaron, B., 1992. Simple technical trading rules and the stochastic properties of stock returns. The Journal of finance 47 (5), 1731–1764.

Cai, J., Li, Y., Qi, Y., 2006. The day-of-the-week effect: New evidence from the chinese stock market. Chinese Economy 39 (2), 71–88.

Casalin, F., 2018. Determinants of holiday effects in mainland chinese and hongkong markets. China Economic Review 49, 45–67.

Cervelló-Royo, R., Guijarro, F., Michniuk, K., 2015. Stock market trading rule based on pattern recognition and technical analysis: Forecasting the djia index with intraday data. Expert systems with Applications 42 (14), 5963–5975.

Chang, P.-C., Fan, C.-Y., Liu, C.-H., 2008. Integrating a piecewise linear representation method and a neural network model for stock trading points prediction. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 39 (1), 80–92.

Chang, P.-C., Liao, T. W., Lin, J.-J., Fan, C.-Y., 2011. A dynamic threshold decision system for stock trading signal detection. Applied Soft Computing 11 (5), 3998–4010.

Chang, P.-C., Liu, C.-H., Lin, J.-L., Fan, C.-Y., Ng, C. S., 2009. A neural network with a case based dynamic window for stock trading prediction. Expert Systems with Applications 36 (3), 6889–6898.

Chavarnakul, T., Enke, D., 2008. Intelligent technical analysis based equivolume charting for stock trading using neural networks. Expert Systems with Applications 34 (2), 1004–1017.

Chen, A.-S., Leung, M. T., 2004. Regression neural network for error correction in foreign exchange forecasting and trading. Computers & Operations Research 31 (7), 1049–1068.

Chen, A.-S., Leung, M. T., Daouk, H., 2003. Application of neural networks to an emerging financial market: forecasting and trading the taiwan stock index. Computers & Operations Research 30 (6), 901–923.

Chen, T.-l., Chen, F.-y., 2016. An intelligent pattern recognition model for supporting investment decisions in stock market. Information Sciences 346, 261–274.

Chen, Y., Hao, Y., 2018. Integrating principle component analysis and weighted support vector machine for stock trading signals prediction. Neurocomputing 321, 381–402.

Chen, Y., Yang, B., Abraham, A., 2007. Flexible neural trees ensemble for stock index modeling. Neurocomputing 70 (4-6), 697–703.

Chenoweth, T., ObradoviĆ, Z., Lee, S. S., 2017. Embedding technical analysis into neural network based trading systems. Artificial Intelligence Applications on Wall Street, 523–541.

Chong, T. T.-L., Ng, W.-K., 2008. Technical analysis and the london stock exchange: testing the macd and rsi rules using the ft30. Applied Economics Letters 15 (14), 1111–1114.

Choudhury, S., Ghosh, S., Bhattacharya, A., Fernandes, K. J., Tiwari, M. K., 2014. A real time clustering and svm based price-volatility prediction for optimal trading strategy. Neurocomputing 131, 419–426.

Chukwuogor, C. N., et al., 2006. An econometric investigation of the day-of-the-week effect and returns volatility in fifteen asia pacific financial markets (1998-2003). Applied Econometrics and International Development 6 (1).

Conrad, J., Kaul, G., 1998. An anatomy of trading strategies. The Review of Financial Studies 11 (3), 489–519.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine learning 20 (3), 273–297.

Cox, D. R., 1958. The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological) 20 (2), 215–232.

Demsetz, H., 1968. The cost of transacting. The quarterly journal of economics 82 (1), 33–53.

Downes, J., Goodman, J., 2014. Dictionary of finance and investment terms. Simon and Schuster.

Duda, R. O., Hart, P. E., Stork, D. G., 1973. Pattern classification and scene analysis. Vol. 3. Wiley New York.

Eric, D., Andjelic, G., Redzepagic, S., 2009. Application of macd and rvi indicators as functions of investment strategy optimization on the financial market. Zbornik radova Ekonomskog fakulteta u Rijeci: časopis za ekonomsku teoriju i praksu 27 (1), 171–196.

Fama, E. F., 1995. Random walks in stock market prices. Financial analysts journal 51 (1), 75–80.

Fernández-Blanco, P., Bodas-Sagi, D. J., Soltero, F. J., Hidalgo, J. I., 2008. Technical market indicators optimization using evolutionary algorithms. Proceedings of the 10th annual conference companion on Genetic and evolutionary computation, 1851–1858.

Fernandez-Rodrıguez, F., Gonzalez-Martel, C., Sosvilla-Rivero, S., 2000. On the profitability of technical trading rules based on artificial neural networks: Evidence from the madrid stock market. Economics letters 69 (1), 89–94.

Feuerriegel, S., Prendinger, H., 2016. News-based trading strategies. Decision Support Systems 90, 65–74.

Fong, S., Si, Y.-W., Tai, J., 2012. Trend following algorithms in automated derivatives market trading. Expert systems with applications 39 (13), 11378–11390.

Fong, S., Tai, J., Si, Y. W., 2011. Trend following algorithms for technical trading in stock market. Journal of Emerging Technologies in Web Intelligence 3 (2).

Fong, W. M., Yong, L. H., 2005. Chasing trends: recursive moving average trading rules and internet stocks. Journal of Empirical Finance 12 (1), 43–76.

Fu, T.-c., Chung, F.-l., Luk, R., Ng, C.-m., 2007. Stock time series pattern matching: Template-based vs. rule-based approaches. Engineering Applications of Artificial Intelligence 20 (3), 347–364.

Fuertes, A.-M., Miffre, J., Rallis, G., 2010. Tactical allocation in commodity futures markets: Combining momentum and term structure signals. Journal of Banking & Finance 34 (10), 2530–2548.

Gorgulho, A., Neves, R., Horta, N., 2011. Applying a ga kernel on optimizing technical analysis rules for stock picking and portfolio composition. Expert systems with Applications 38 (11), 14072–14085.

Granville, J. E., 1976. Granville's new strategy of daily stock market timing for maximum profit.

Han, Y., Zhou, G., Zhu, Y., 2016. A trend factor: Any economic gains from using information over investment horizons? Journal of Financial Economics 122 (2), 352–375.

Hendershott, T., Jones, C. M., Menkveld, A. J., 2011. Does algorithmic trading improve liquidity? The Journal of Finance 66 (1), 1–33.

Hu, Y., Feng, B., Zhang, X., Ngai, E., Liu, M., 2015a. Stock trading rule discovery with an evolutionary trend following model. Expert Systems with Applications 42 (1), 212–222.

Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E., Liu, M., 2015b. Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. Applied Soft Computing 36, 534–551.

Imandoust, S. B., Bolandraftar, M., 2014. Forecasting the direction of stock market index movement using three data mining techniques: the case of tehran stock exchange. International Journal of Engineering Research and Applications 4 (6), 106–117.

Kim, H. S., Sohn, S. Y., 2010. Support vector machines for default prediction of smes based on technology credit. European Journal of Operational Research 201 (3), 838–846.

Korajczyk, R. A., Sadka, R., 2004. Are momentum profits robust to trading costs? The Journal of Finance 59 (3), 1039–1082.

Kwon, K.-Y., Kish, R. J., 2002. Technical trading strategies and return predictability: Nyse. Applied Financial Economics 12 (9), 639–653.

Ładyżyński, P., Żbikowski, K., Grzegorzewski, P., 2013. Stock trading with random forests, trend detection tests and force index volume indicators. International Conference on Artificial Intelligence and Soft Computing, 441–452.

Larson, M., 2012. 12 simple technical indicators: That really work 69.

Leigh, W., Modani, N., Purvis, R., Roberts, T., 2002a. Stock market trading rule discovery using technical charting heuristics. Expert Systems with Applications 23 (2), 155–159.

Leigh, W., Purvis, R., Ragusa, J. M., 2002b. Forecasting the nyse composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. Decision support systems 32 (4), 361–377.

Levy, R. A., 1971. The predictive significance of five-point chart patterns. Journal of Business, 316–323.

Liaw, A., Wiener, M., et al., 2002. Classification and regression by randomforest. R news 2 (3), 18–22.

Lo, A. W., 2004. The adaptive markets hypothesis. The Journal of Portfolio Management 30 (5), 15–29.

Lo, A. W., Mamaysky, H., Wang, J., 2000. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. The journal of finance 55 (4), 1705–1765.

Lu, X., Mehran, J., Gao, H., 2016. Holiday trading in china: Before and during the financial crisis. Journal of Applied Finance and Banking 6 (2), 117.

Luo, J., Gan, C., Hu, B., Kao, T., 2009. An empirical analysis of chinese stock price anomalies and volatility. Investment Management and Financial Innovations 6 (1), 1–18.

Luo, L., Chen, X., 2013. Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction. Applied Soft Computing 13 (2), 806–816.

Luo, L., You, S., Xu, Y., Peng, H., 2017. Improving the integration of piece wise linear representation and weighted support vector machine for stock trading signal prediction. Applied Soft Computing 56, 199–216.

Mabu, S., Hirasawa, K., Obayashi, M., Kuremoto, T., 2013. Enhanced decision making mechanism of rule-based genetic network programming for creating stock trading signals. Expert Systems with Applications 40 (16), 6311–6320.

Marshall, B. R., Cahan, R. H., Cahan, J. M., 2008. Does intraday technical analysis in the us equity market have value? Journal of Empirical Finance 15 (2), 199–210.

Marshall, B. R., Cahan, R. M., 2005. Is the 52-week high momentum strategy profitable outside the us? Applied Financial Economics 15 (18), 1259–1267.

Metghalchi, M., Chang, Y.-H., Marcucci, J., 2008. Is the swedish stock market efficient? evidence from some simple trading rules. International Review of Financial Analysis 17 (3), 475–490.

Michniuk, K., 2017. Pattern recognition applied to chart analysis. evidence from intraday international stock markets. Ph.D. thesis.

Nam, K., Washer, K. M., Chu, Q. C., 2005. Asymmetric return dynamics and technical trading strategies. Journal of Banking & Finance 29 (2), 391–418.

Narang, R. K., 2009. Inside the Black Box: The Simple Truth about Quantitative Trading.-Description Based on Print Version Record. J. Wiley et Sons.

Narayan, P. K., Ahmed, H. A., Narayan, S., 2015. Do momentum-based trading strategies work in the commodity futures markets? Journal of Futures Markets 35 (9), 868–891.

Neupane, B., Woon, W., Aung, Z., 2017. Ensemble prediction model with expert selection for electricity price forecasting. Energies 10 (1), 77.

Ng, A., 2000. Cs229 lecture notes. CS229 Lecture notes 1 (1), 1–3.

Ohlson, J. A., 1980. Financial ratios and the probabilistic prediction of bankruptcy. Journal of accounting research, 109–131.

Pai, P.-F., Hsu, M.-F., Wang, M.-C., 2011. A support vector machine-based model for detecting top management fraud. Knowledge-Based Systems 24 (2), 314–321.

Paleologo, G., Elisseeff, A., Antonini, G., 2010. Subagging for credit scoring models. European Journal of Operational Research 201 (2), 490–499.

Patel, J., Shah, S., Thakkar, P., Kotecha, K., 2015. Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications 42 (4), 2162–2172.

Pesaran, M. H., Timmermann, A., 1994. Forecasting stock returns an examination of stock market trading in the presence of transaction costs. Journal of forecasting 13 (4), 335–367.

Principe, J. C., Euliano, N. R., Lefebvre, W. C., 2000. Neural and adaptive systems: fundamentals through simulations. Vol. 672. Wiley New York.

Ready, M. J., 2002. Profits from technical trading rules. Financial Management, 43–61.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1985. Learning internal representations by error propagation.

Schulmeister, S., 2008. Components of the profitability of technical currency trading. Applied Financial Economics 18 (11), 917–930.

Schulmeister, S., 2009. Profitability of technical stock trading: Has it moved from daily to intraday data? Review of Financial Economics 18 (4), 190–201.

Sezer, O. B., Ozbayoglu, A. M., 2018. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Applied Soft Computing 70, 525–538.

Shin, K.-S., Lee, T. S., Kim, H.-j., 2005. An application of support vector machines in bankruptcy prediction model. Expert systems with applications 28 (1), 127–135.

Sullivan, R., Timmermann, A., White, H., 1999. Data-snooping, technical trading rule performance, and the bootstrap. The journal of Finance 54 (5), 1647–1691.

Szakmary, A. C., Shen, Q., Sharma, S. C., 2010. Trend-following trading strategies in commodity futures: A re-examination. Journal of Banking & Finance 34 (2), 409–426.

Tang, L., Wu, Y., Yu, L., 2018. A non-iterative decomposition-ensemble learning paradigm using rvfl network for crude oil price forecasting. Applied Soft Computing 70, 1097–1108.

Trippi, R. R., DeSieno, D., 1992. Trading equity index futures with a neural network. Journal of Portfolio Management 19, 27–27.

Troiano, L., Villa, E. M., Loia, V., 2018. Replicating a trading strategy by means of lstm for financial industry applications. IEEE transactions on industrial informatics 14 (7), 3226–3234.

Tsai, C.-F., Lin, Y.-C., Yen, D. C., Chen, Y.-M., 2011. Predicting stock returns by classifier ensembles. Applied Soft Computing 11 (2), 2452–2459.

Vapnik, V., 1998. Statistical learning theory. john wiley&sons. Inc., New York.

Wang, J.-L., Chan, S.-H., 2009. Trading rule discovery in the us stock market: An empirical study. Expert Systems with Applications 36 (3), 5450–5455.

Wang, J.-L. e. a., 2007. Stock market trading rule discovery using pattern recognition and technical analysis. Expert Systems with Applications 33 (2), 304–315.

Wilder, J. W., 1978. New concepts in technical trading systems.

Wolpert, D. H., 1996. The lack of a priori distinctions between learning algorithms. Neural computation 8 (7), 1341–1390.

Xian, L., He, K., Lai, K. K., 2016. Gold price analysis based on ensemble empirical model decomposition and independent component analysis. Physica A: Statistical Mechanics and its Applications 454, 11–23.

Xiao, Y., Xiao, J., Lu, F., Wang, S., 2014. Ensemble anns-pso-ga approach for day-ahead stock e-exchange prices forecasting. International Journal of Computational Intelligence Systems 7 (2), 272–290.

Xiong, X., Meng, Y., Li, X., Shen, D., 2018. An empirical analysis of the adaptive market hypothesis with calendar effects: Evidence from china. Finance Research Letters.

Yu, L., Wang, S., Lai, K. K., 2008. Credit risk assessment with a multistage neural network ensemble learning approach. Expert systems with applications 34 (2), 1434–1444.

Yu, L. e. a., 2008. Forecasting crude oil price with an emd-based neural network ensemble learning paradigm. Energy Economics 30 (5), 2623–2635.

Zapranis, A., Tsinaslanidis, P. E., 2012. A novel, rule-based technical pattern identification mechanism: Identifying and evaluating saucers and resistant levels in the us stock market. Expert Systems with Applications 39 (7), 6301–6308.

Zhang, J., Lai, Y., Lin, J., 2017. The day-of-the-week effects of stock markets in different countries. Finance Research Letters 20, 47–62.

Zhao, Y., Li, J., Yu, L., 2017. A deep learning ensemble approach for crude oil price forecasting. Energy Economics 66, 9–16.

Appendix A

Machine learning models

A.1 Basic structure of logistic regression

For a classification problem, the target variable (output) y can only have discrete values for a given set of features (inputs) x.

The hypothesis[1] for logistic regression model is

$$h_\theta(x) = g(\theta^T x) \tag{A.1}$$

where $g(z) = \frac{1}{1+e^{-z}}$, g is called the logistic function or the sigmoid function.

The conditional probabilities $\mathbf{P}$ for 2 labels (0 and 1) for $i-th$ observation are defined as

$$\begin{gathered} \mathbf{P}(y^{(i)} = 1|x^{(i)};\theta) = h(x^{(i)}), \\ \mathbf{P}(y^{(i)} = 0|x^{(i)};\theta) = 1 - h(x^{(i)}) \end{gathered} \tag{A.2}$$

The cost function (J) is

$$J(\theta) = -\tfrac{1}{m}[\textstyle\sum_{i=1}^{m} y^{(i)} \ln{(h_\theta(x^{(i)}))} + (1-y^{(i)}) \ln{(1-h_\theta(x^{(i)}))}] \tag{A.3}$$

[1]The formula we use for calculating h(x) is called a hypothesis.

To minimise J with respect to $\theta_j \in \theta$ (j-th feature), we repeat

$$h_{\theta_j}(x) = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2 \tag{A.4}$$

simultaneously update for all θ_j, where α is called learning rate, is the penalty parameter.

A.2 Basic structure of support vector machine

SVM represents the relationship between an output y and a set of inputs **x** in the form (Cortes and Vapnik, 1995):

$$f(x) = \sum_{i=1}^{N} w_i \phi(x) + b \tag{A.5}$$

where $\phi(x)$ represents a non-linear mapping of **x** into a higher dimensional feature space, i.e. a basis function, and **w** and b are parameters learnt from the N instances of training data.

In classification, these parameters are found by using Quadratic Programming (QP) optimisation to first find the α_i which maximise:

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i) \cdot \phi(x_j) \tag{A.6}$$

where $\alpha_i > 0$ $\forall_i$, $\sum_{i,j}^{N} \alpha_i y_i = 0$. The α_i are then used to find **w**:

$$w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \tag{A.7}$$

The set of Support Vectors S is then found by finding the indices i where $\alpha_i > 0$. b can then be calculated:

$$b = \frac{1}{N_s} \sum_{m \in S} (y_s - \sum_{s \in S} \alpha_m y_m \phi(x_m) \cdot \phi(x_s)) \tag{A.8}$$

The mapping x $\rightarrow \phi(x)$ is intended to make the data linearly separable in the feature space, and the kernel $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. The Radial Basis Function (RBF): $k(x_i, x_j) = e^{-(\frac{||x_i - x_j||^2}{2\sigma^2})}$ and the Linear Kernel: $k(x_i, x_j) = x_i x_j^T$ are commonly used. In this study, the RBF kernel is utilised for SVM.

A.3 Basic structure of artificial neural network

Figure A.1 depicts the structure of the multi-layer perceptron (MLP). Specifically, the input layer on the left-hand side of the figure corresponds with the vector of independent variables. The two hidden layers, where data transformations occur, are presented in the centre of the figure. Finally, the output layer on the right-hand side of the figure generates predictions of the dependent variable.

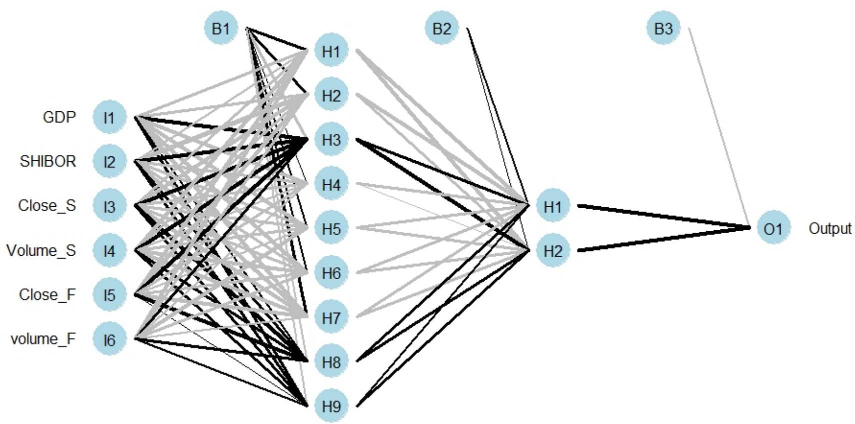


Figure A.1: The structure of the feed forward neural networks

More specifically, the input, hidden and output layers are noted as I, H, O, respectively. Let w$^{(L)}$ denote the weights at level L, L=1, 2, 3. Let $(I_p, O_p), p = 1, 2, \ldots, P$ represent the input and output vectors for training the neural network. For each input $I_{pi}(i = 0, 1, \ldots, m)$, the neurons H_{pj}, H_{pk} and O_{pl} are calculated as follows:

$$H_{pj}(w^{(1)}) = f(\textstyle\sum_{n=0}^{m} I_{pi} w_{ij}^{(1)}), j = 1, 2, \ldots, n \tag{A.9}$$

$$H_{pk}(w^{(1)}, w^{(2)}) = f(\textstyle\sum_{j=0}^{n} H_{pj} w^{(1)} \cdot w_{jk}^{(2)}), k = 1, 2, \ldots, K \tag{A.10}$$

$$O_{pl}(\mathrm{w}^{(2)}, \mathrm{w}^{(3)}) = \textstyle\sum_{k=0}^{K} H_{pj} w^{(2)} \cdot w_{jl}^{(3)}, l = 1, 2, \ldots, L \tag{A.11}$$

The activation function for the hidden layer can be sigmoid, tanh and Relu function. The training of the neural network is executed by minimising the following objective function with L2 penalty (regularization term) term:

$$Cost(w) = \tfrac{1}{P \times K \times L} \textstyle\sum_{p=1}^{n} \sum_{k=1}^{K} \sum_{l=1}^{L} (O_{pl} - Y_{pl})^2 + \alpha ||w||_2 \tag{A.12}$$

where Y_{pl} is the true value of the fitted O_{pl}, α is the L2 penalty (regularization term) parameter[2], and $\mathrm{w} = w^{(1)} \cup w^{(2)} \cup w^{(3)}$.

[2]In our model, the penalty parameter $\alpha = 0.01$.

A.4 LBFGS Optimiser

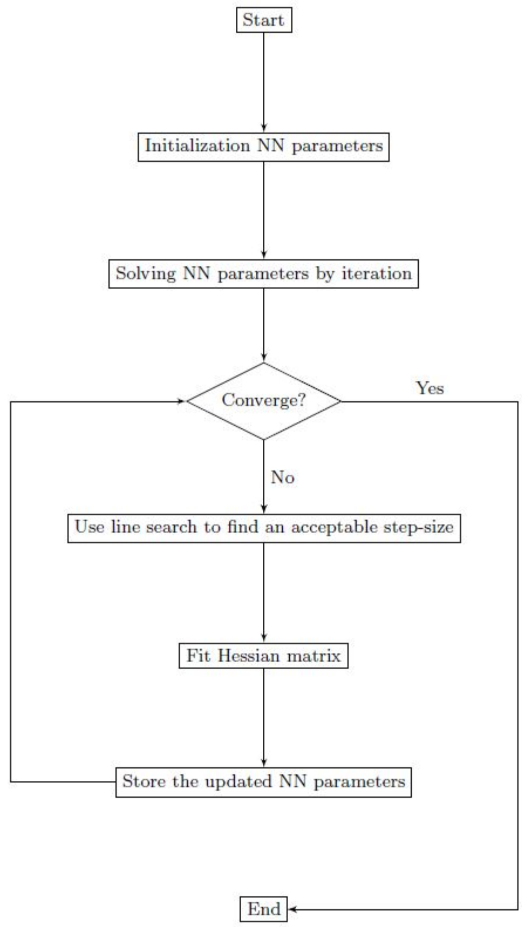


Figure A.2: The structure of LBFGS

To minimize the cost function efficiently, limited-memory BFGS (LBFGS) algorithm is employed in this research (see Figure A.2 in Appendix).

Appendix B

Backtesting Algorithms

B.1 Algorithm for Vote ('L')

Input: $data, Wind, cost$
Output: $TotalReturn_L$

Initial State $S = 0$
for *i in test set* **do**
    da(i) = data[i-Wind,i]
    $y_{predict}(i)$ = AICLASSIFIER(da(i))
    $Vote = SUM(AICLASSIFIER(da(i)))$
    if $S = 0$ *AND* $Vote > 1$ **then**
        S=1 ; `// Open "buy"`
        Return = ln(P(i+1)/P(i))-cost
    if $S = 1$ *AND* $Vote > 1$ **then**
        Return = ln(P(i+1)/ P(i)) ; `// Hold "buy"`
    else
        S = 0
        Return = 0 ; `// Close "buy"`
    end
end

TotalReturn$_L$ = $cumsum(Return)$

Algorithm 12: Back testing for trading using majority vote prediction ('L')

B.2 Algorithm for Vote ('LS')

Input: $data, Wind, cost$
Output: $TotalReturn_{LS}$

Initial State $S = 0$
for *i in test set* **do**
  da(i) = data[i-Wind,i]
  $y_{predict}(i)$ = AICLASSIFIER(da(i))
  $Vote = SUM(AICLASSIFIER(da(i)))$
  if $S = 0$ AND $Vote > 1$ **then**
    S=1 ; `// Open "buy"`
    Return = ln(P(i+1)/P(i))
  if $S = 0$ AND $Vote < 2$ **then**
    S= -1 ; `// Open "sell"`
    Return = ln(P(i)/P(i+1))
  if $S = 1$ AND $Vote > 1$ **then**
    Return = ln(P(i+1)/ P(i)) ; `// Hold "buy"`
  if $S = -1$ AND $Vote < 2$ **then**
    Return = ln(P(i)/ P(i+1)) ; `// Hold "sell"`
  else
    S = 0
    Return = 0 ; `// Close`
  end
end

TotalReturn$_{LS}$ = $cumsum(Return)$

Algorithm 13: Back testing for trading using majority vote prediction ('LS')