ESTIMATING DIRECTIONAL CHANGES TREND REVERSAL IN FOREX USING MACHINE LEARNING

A THESIS SUBMITTED TO

THE UNIVERSITY OF KENT

IN THE SUBJECT OF COMPUTER SCIENCE

FOR THE DEGREE

OF PHD.

By

Adesola Tolulope Noah Adegboye

March 2022

Acknowledgements

First of all, I would like to thank my supervisors Dr Fernando Otero and Dr Michael Kampouridis for their indelible help, support and guidance along these 6 year journey. I would also like to say a big thank you to my wife Amanda and daughters Morayo and Ayomi for their emotional support and encouragement through my PhD journey, most especially in those days when it was overwhelming, and the journey seemed unending. In addition, I would like to thank my parents for their words of inspiration, prayers and teachings on perseverance and dedication which were fundamental for completion. Finally, I thank God for providing all the resources necessary for completing the PhD programme. It has been a very long journey and without these people I won't have managed to reach the end. Once again, thank you.

Abstract

Most forecasting algorithms use a physical time scale data to study price movement in financial markets by taking snapshots in fixed schedule, making the flow of time discontinuous. The use of a physical time scale can make traders oblivious to significant activities in the market, which poses risks. For example, currency risk, the risk that exchange rate will change. Directional changes is a different and newer approach of taking snapshot of the market, which uses an event-based time scale. This approach summarises data into alternating trends called upward directional change and downward directional change according to a change in price a trader considers to be significant, which is expressed as a threshold. The trends in the summary are split into directional change (DC) and overshoot (OS) events. In this work, we propose a novel DC-based framework, which uses machine learning algorithms to forecast when the next, alternate trend is expected to begin. First, we present a genetic programming (GP) algorithm that evolves equations that express linear and non-linear relationships between the length of DC and OS events in a given dataset. Awareness of DC event and OS event lengths provide traders with an idea of when DC trends are expected to reverse and thus take appropriate action to increase profit or mitigate risk. Second, DC trends can be categorised into two distinct types: (1) trends with OS events; and (2) trends without OS events(i.e. OS event length is 0).

Trends with OS events are those that continue beyond a period when they were first observed and trends without OS event are others that ends as soon as they were observed. To further improve trend reversal estimation accuracy, we identified these two categorises using classification techniques and estimated OS event length for trends that belong in the first category. We appraised whether this new knowledge could lead to an even greater excess return. Third, our novel trend reversal estimation approach was then used as part of a novel genetic algorithm (GA) based trading strategy. The strategy embedded an optimised trend reversal forecasting algorithm that was based on trend reversal point forecasted by multiple thresholds. We assessed the efficiency of our framework (i.e., a novel trend reversal approach and an optimised trading strategy) by performing an in-depth investigation. To assess our approach and evaluate the extent to which it could be generalised in Forex markets, we used five tailored thresholds to create 1000 DC datasets from 10, monthly 10-minute physical time data of 20 major Forex markets (i.e 5 thresholds * 10 months * 20 currency pairs). We compared our results to six benchmarks techniques, both DC and non-DC based, such as technical analysis and buy-and-hold. Our findings showed that our proposed approach can return a significantly higher profit at reduced risk, and statistically outperformed the other trading strategies compareds in a number of different performance metrics.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

After the collapse of the gold standard, the Bretton Woods system, a collective international currency exchange regime was established (Igwe 2018). The United States (US) dollar became the replacement for gold and other national currencies were pegged to the US dollar. With the Bretton Woods System in place, volatility of Forex rates was minimised. However, the US government was unable maintain international dollar liquidity as foreign claims on gold started to exceed US gold supply (Garber 2007). In the early 1970's, US government unilaterally decided to stop exchanging gold for the US Dollar and the Bretton Woods system was abolished. In 1973, there were changes in currency policies of the world major currencies and floating exchange rate regime emerged (Peng, Wang and Yeh 2020).

Financial forecasting in the Forex market are attempts to estimate future exchange rate or predict trend reversal through patterns discovery in historical price or traded volume (Yu, Wang and Lai 2007; Islam et al. 2020). Up to the early 1970s, exchange rate was mostly determined by the balance of payments of countries and their level of importation and exportation of goods and services (Chang and Huang

2014). The float rate regime introduced after 1973 has attracted more market participants across the globe which has led to significant increase in Forex market size to about $6.6 trillion in daily turnover as of April 2019 (Wooldridge 2019). The increase in participants coupled with local and international supply-demand factors, such as economical, political and psychological have made forecasting in the Forex market a challenging task. Some of these challenges include: 1) pronounce price fluctuation in the short term (Folger 2020); 2) low-profit margin in comparison to fixed income trading (Petropoulos et al. 2017); and 3) noise and chaotic signals, which make separation of uninteresting features from trends difficult (Abu-Mostafa and Atiya 1996; Kamruzzaman, Sarker and Ahmad 2003).

Despite these inherent challenges, Forex market presents a major opportunity for informed traders and algorithmic trading developers to make profit. This has motivated researchers both in academia and industry to investigate regularities in the Forex market. There are two main methods for performing this investigation, namely fundamental analysis and technical analysis.

In fundamental analysis approach, fundamentalists evaluate assets' primary characteristics and financial data, such as Interest Rates (IR), Employment Reports, Inflation, Gross Domestic Product (GDP), Consumer Price Index (CPI), Producer Price Index (PPI), Institute of Supply Management (ISM), Commodity Price Index (CPI), Industry Production Index (IPI), Retail sales Report, Trade Flow and Trade Balance, Balance of Payment (BOP), Purchasing Power Parity (PPP), central bank policies and geopolitical events (Dao, McGroarty and Urquhart 2019). On the other hand, technicians evaluate historical price values. The motivation for evaluating historical price value is based on the belief that all the fundamentals that cause a change in value of an asset have been factored into the current price and repeating trends are observable in historical price data(Cavalcante et al. 2016). Hence, traders are able to discover winning strategies from a series of historical price snapshots.

Both fundamental and technical analysis are complimentary. For example, assuming we are interested in purchasing a personal computer brand sometime in the future. To determine the expected price, we can choose to evaluate the different components that make up a computer such as hard-drive, memory, processor, monitor, mouse, keyboard, etc. In contrast, historical sales price of the computer brand in the recent past can be evaluated.

Previously, academics adopted fundamental analysis approach and discarded technical analysis because they concluded that it did not perform better than random walk (Malkiel 1999; Lo, Mamaysky and Wang 2000). However, there are works that recorded success in using technical analysis for forecasting. For example Lo and MacKinlay (1988) demonstrated that past prices can be used to predict future profit. This finding was also corroborated by Plastun (2017) who explained that the success of technical analysis is based on a phenomenon called "Behavioural Finance Market Hypotheses" (BFMH). BFMH describes the role investors' emotions and psychology play in their financial decision making. Investors evaluate disseminated information and news differently according to their experiences, culture and needs which technical analyst capitalise on, in the short term (Yildirim 2017). It is now common practice to use fundamental analysis for long-term prediction and technical analysis for short-term prediction (Cavalcante et al. 2016).

Majority of technical analysis studies utilise historical market data snapshots taken at fixed intervals (Aloud 2017). To generate the snapshots, investors decide how often to sample the data, then snapshots are taken at the chosen frequency forming an interval-based summary. A drawback of using interval-based summary is that it ignores market activities between snapshots that could be significant, exposing market participants to risks such as currency risk, interest-rate risks and so on. An alternative approach is to utilise snapshots of markets activities that are significant. To generate these snapshots, investors decide on what a significance market activity is, then snapshots are taken when the activity is observed.

In our study, we used directional changes (DC), an approach for taking snapshots of historical price values on an intrinsic time scale. In the directional changes approach, data is recorded when there is a change in price by a predetermined threshold θ. The threshold value is decided in advance by a trader according to his or her belief of what a significant upwards or downwards price change is. This concept provides traders with a new way of viewing historical data, allowing them to focus on key price movements, thereby, blurring out other price details which could be considered irrelevant or noise.

Furthermore, DC has enabled researchers to discover new statistical properties that were not previous captured from interval-based summaries (Glattfelder, Dupuis and Olsen 2011). Thus, these new properties give rise to novel opportunities for traders and open a whole new area for research.

There are two main types of techniques used for financial analysis, namely statistical modelling and machine learning (Wang et al. 2011). The main difference between the two approaches are their purpose. Statistical modelling is the use of statistics to develop a representation of data, perform inference on the relationships between data attributes to establish significance (Sidehabi, Tandungan et al. 2016). In Statistical modelling, a probabilistic model is explicitly specified and attributes of interest that has effect on the predictor attribute are identified (McCullagh 2002). On the other hand, machine learning, does not impose a model, it empirically discovers relationships between attributes for the purpose of prediction. Nonetheless, statistical models could also be used for prediction and to do that, an intermediate step such as model transformation or independent variable segmentation is performed

(Julier and Uhlmann 2004; Bach and Ghil 2022).

In this work, we aim to gain further insight into trend reversal forecasting under DC approach for prediction trends reversal with improved accuracy using machine learning. DC event is an activity in the market that has caused a change in price either upward or downward by a predefined threshold considered to be significant by a user. OS event can be described as the movement in price in the same direction as a preceding DC event from the point the trend is first confirmed to be occurring until the event that ceases to impact the market and a new DC event in the opposite direction starts. One of the most interesting regularities discovered in DC was a scale invariant linear relationship between DC and OS event lengths, which, when combined gave an estimate of trend reversal points Guillaume et al. (1997); Glattfelder, Dupuis and Olsen (2011). The regularity was that if on average a DC takes t amount of physical time to complete, the OS event will take an amount of $2t$. Although the regularity, through empirical tests as shown to holds across Forex financial time series and thresholds Glattfelder, Dupuis and Olsen (2011), it does not always express the richest relationships in DC event series. We therefore examined the regularity further, to discovered richer linear and non-linear relationships tailored to dataset under observation. We explore machine learning algorithms as tools for 1) discovering whether a relationship exists between DC and OS event lengths; and 2) discovering richer linear and or non-linear relationships when they exist. Our choice of machine learning algorithms is influenced by the characteristics of our dataset. Machine learning algorithms are well suited for exploring dataset where predictor attribute does not take a predetermined form but is constructed according to information derived from the dataset under observation. Assaad and Fayek (2021). Thus, we explore for richer models from DC event series by generating a dataset where the predictor attribute is the DC event length and the dependent attribute is the OS event length. Then used machine learning techniques to explore their relationship search space. The insight gained is then exploited in a proposed trading strategy algorithm that we tested in 20 major Forex markets. We aim to attain higher rate of return at minimum risk than other trading strategies embedded with earlier proposed DC trend reversal forecasting algorithms and other strategies that are based on physical time scale technical indicators.

1.1 Thesis Structure

The remainder of this thesis is organised as follows. Chapter 2 presents an overview of financial forecasting and detail two common approaches: fundamental and technical analysis. We also present an in-depth analysis of directional changes paradigm from a financial forecasting perspective. Chapter 3 presents an overview of machine learning, a well-employed approach for financial forecasting and for developing trading frameworks. Chapter 4 presents the first contribution of the thesis, a novel approach for estimating overshoot events' length using symbolic regression genetic programming. Chapter 5 presents the second contribution of this thesis, an improvement to our approach for forecasting trend reversal, which combined classification and regression techniques for estimating DC trend reversal. Chapter 6 presents the third contribution of this thesis, a novel trading framework that used a genetic algorithm to optimise predictions from multiple trend reversal forecasting models. Chapter 7 presents conclusions and final remarks of the thesis, as well as suggestions for future research.

1.2 Publications

The list of publications from the research described in this in thesis in Peer-Reviewed Journals are as follows:

- Adegboye, A. and Kampouridis, M. (2021). Machine learning classification and regression models for predicting directional changes trend reversal in FX markets. *Expert Systems with Applications*, 173, p. 114645, DOI:10.1016/j.eswa.2021.114645, Impact factor:5.452.

- Adegboye, A., Kampouridis, M. and Otero, F. (2021). Improving trend reversal estimation in forex markets under a directional changes paradigm with classification algorithms. *International Journal of Intelligent Systems*, DOI: 10.1002/int.22601, Impact factor: 8.709.

The list of publications from the research described in this in thesis in Conference Proceedings are as follows:

- Adegboye, A., Kampouridis, M. and Johnson, C. G. (2017). Regression genetic programming for estimating trend end in foreign exchange market. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, pp. 1–8.

- Kampouridis, M., Adegboye, A. and Johnson, C. (2017). Evolving directional changes trading strategies with a new event-based indicator. In *Asia-Pacific Conference on Simulated Evolution and Learning*, Springer, pp. 727–738

Chapter 2

Financial Forecasting

Financial forecasting can be defined as an attempt to predict future market events through patterns discovery in historical data (Yıldırım, Toroslu and Fiore 2021). In this work, we developed a Forex trading framework for making decisions according to certain trading rules and accurate trend reversal forecasting algorithm. The major challenge is the nature of the data which seem random because of the inherent noise, making trend reversal forecasting a hard problem (Walczak 2001).

There are two schools of thought concerning financial forecasting. One considers that the market is perfect and cannot be predicted as it is a summary of all available information (Zafar 2012). This idea is based on two hypotheses, namely (1) the efficient market hypothesis (EMH) (Fama 1970) and (2) random walk hypothesis (RWH) (Regnault 1863; Levy 1967; Fama 1995). The other considers that, in the short term, markets can be predicted by analysing patterns that may exist in historical market data (Edwards, Magee and Bassetti 2012; Antony 2020).

We expatiate on the two schools of thought in the remainder of the chapter. In Section 2.1 we present a high-level description of the random walk hypothesis. In

Section 2.2, we present an overview of the efficient market hypothesis and arguments supporting the unfeasibility of forecasting. In Section 2.3, we describe financial analysis, detailing two common approaches: fundamental analysis and technical analysis. These two approaches forecast future market movement using snapshots of data, taken at constant time intervals. In Section 2.4, we present event-based approach, a new way of summarising data, which is the technique used in this thesis.

2.1 Random walk hypothesis

The idea is that financial market cannot be predicted because price changes are stochastic and each price change is independent of the preceding and following price changes (Pesaran and Pick 2008). In essence, it was argued that old price values do not have the power to forecast future prices and the current publicly available price is considered the best prediction. Nevertheless, it was shown that underlying economic factors that determine price can be subjected to structural breaks[1] such as wars, major change in government policy, or any sudden event other than demand and supply activities generated by the market. It is sometime the case for these breaks, also known as “Random Walk with a Drift” to be cyclical and recorded in historical data (Krol 1992; Gabrielli and Valente 2011). These cyclical breaks introduces deterministic trends creating opportunity for forecasting (Pesaran and Pick 2008).

gabrielli2011random

[1] A sudden change in market volatility because of unexpected shift in economic factors (Stawiarski 2015)

2.2 Efficient Market Hypothesis (EMH)

Similarly to Random Market Hypothesis, EMH is also based on the consideration that the financial market cannot be predicted. However, the difference lies in the reason why the financial market is not predictable. The Efficient Market Hypothesis (EMH) is a proposition that asset-price incorporates and shows all publicly available information about its value, and it is impossible to earn a profit from forecasting future prices because the current price is the best price available adjusted for errors in previous price and future price will adjusts current price accordingly to emergence of new information (Fama 1970). EMH is based on the following assumptions (1) all relevant information is available to all investors; (2) investors are rational; (3) the market is rational; (4) information acquisition cost is the same for all participants;(5) there are no taxes; (6) there is no transaction costs; and (7) investors are insensitive to the currency of earnings (Zafar 2012). There are three types of EMH: weak, semi-strong, and strong (Eom et al. 2008).

Weak EMH is based on the assertion that excess return cannot be made from analysing historical prices because the current price includes all past information. Thus, an investor is unable to profit from publicly available information. Any future price adjustment would be the result of new information. The time of arrival of the new information is arbitrary, consequently, future price will also be arbitrary (Stasinakis and Sermpinis 2014).

Semi-strong EMH asserts that excess return cannot be made from analysing publicly available information because all investors have access to new information, which will rapidly reflect in the next future price. Publicly available information can be of financial nature, like historical prices, data reported in financial statements, dividend announcements and merger plan announcements. It can also be non-financial such as innovation plans, patent applications, etc. The assumption here is stronger than weak EMH, because the eventual price of an asset has already taken into consideration all publicly available information. Excess return can only be made if an investor is in possession of information that is unknown to the market (Stasinakis and Sermpinis 2014).

Strong EMH asserts that excess return cannot be made from analysing both publicly and privately available information. It is impossible to profit on privately available information for two reasons, (1) it is a crime and an illegal activity (Woody 2020), and (2) the excess demand/supply will cause under-priced or overpriced asset to adjust swiftly to the level supported by the new information (Stasinakis and Sermpinis 2014).

Earlier research works supported the claim that it is impossible to profit from financial forecasting (Fama 1970; Jensen 1978). However, more recent studies demonstrated that it is possible, rejecting the EMH (Fernandez-Rodrıguez, Gonzalez-Martel and Sosvilla-Rivero 2000; Kyriazis 2019). The rational supporting this claim includes predictable and deliberate human error in reasoning and processing information (Neely, Weller and Dittmar 1997). Human emotions also play a role when taking trading decisions, for example, some investors can be more reactive than others to new financial market related information (Zafar 2012). Additionally, if the market is perfectly efficient there would be no motivation for professionals and investors to trade (Grossman and Stiglitz 1980). Consequently, financial analysis is an important tool for understanding market behaviour and supporting trading activities. This topic will be discussed in the next section.

2.3 Overview of financial analysis

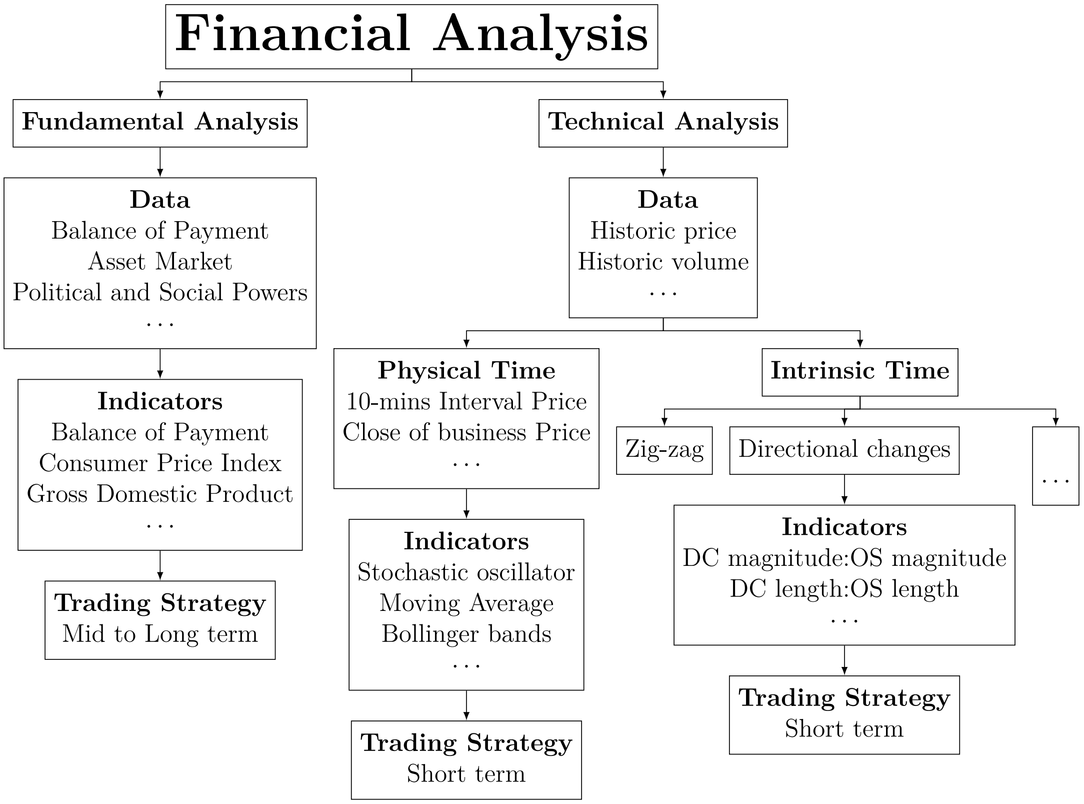


Figure 2.1: Classification of financial analysis.

The goal of financial analysis is to understand market behaviour and use the knowledge gained in making future decisions that can potentially provide beneficial outcomes to investors in maximising profit and reducing risk (Schneeweis 1983). Figure 2.1 illustrates the two approaches used in performing financial analysis, namely fundamental and technical analysis. Fundamental analysis is based on the study of economic factors that influence the demand and supply of an asset to determine its intrinsic value (Petrusheva and Jordanoski 2016). It is a technique primarily used by passive investors who do not seek immediate gain, instead they prefer to take a longer-term investment approach. They are willing to wait for the right moment as long as short losses are within a tolerable limit (Shiryaev, Xu and Zhou 2008).

Financial analysis is data driven, fundamental analysis is based on data generated from macroeconomic activities while technical analysis is based on historical market price and volume (Hu et al. 2015; Datta et al. 2021). Indicators are tools for finding signals of strengths or weaknesses. In both fundamental and technical analysis, indicators are applied to gain insight into and interpret changes in the macroeconomy (fundamental analysis) or historical market data (technical analysis). There are two categories of indicators, (1) lagging indicators explain changes in trends and (2) leading indicators are used to predict future market direction. Signals picked up using indicators are then used to develop a framework of rules and predefined criteria for making trading decisions such as, (1) what to buy and sell; (2) when to buy and sell it; and (3) what quantity to buy and sell (Hayes 2021; Balasubramaniam 2021).

In this work, the focus is on technical analysis. Notwithstanding, we present a brief description of fundamental analysis.

2.3.1 Fundamental Analysis

In fundamental analysis, investors evaluate information such as company revenue, expenses, asset and liabilities to determine performance and potential for future economic growth or contraction (Hu et al. 2015). The analysis is used to aid investors when deciding on 1) long term investment in undervalued assets or assets with growths prospect and 2) sale of overpriced assets or assets tending towards a decline in value. In Forex trading specifically, an investor analyses data to determine a rate that a currency should be exchanged for another. The rate reflects the balance of trade between two economies. However, forecasting the Forex market using fundamental analysis approach is challenging because data used are mostly publicly available and their fundamental value is known to a degree of certainty (Kaltwasser 2010). Hence, there is little competitive advantage from a face-value analysis of such data.

Data

Three types of data employed in analysing the Forex market from the fundamental analysis perspective are, (1) demand and supply of Forex, (2) asset market activities, and (3) political and social events (Lui and Mole 1998; Korczak, Hernes and Bac 2016).

Demand and supply of a currency in exchange for another is swayed by changes in the inflation rate, unemployment rate, interest rates and the balance of imports to exports. When disparity exists, investors, guided by the principles of interest rate parity[2] and carry trade[3] are lured into investing in the higher interest rate country. As a result, the demand for the currency of the higher interest rate country is increased, strengthening the currency. Export to import imbalance between two countries can also lead to the higher valuation of the exporting country's currency. However, it is not uncommon for countries to use external intervention to ensure that export remains competitive as currency valuation goes up. To analyse the impact of changes to interest rate and balance of trade to currency valuation, indicators are employed. Some of these indicators are Balance of Payment, Consumer Price Index, Gross Domestic Product and so on (Korczak, Hernes and Bac 2016).

[2] Interest rate parity is the difference between forward exchange rate of one currency and the spot rate of another currency.

[3] Carry trade is the sale of lower interest rate currency to purchase asset denominated in higher interest rate currency to benefit from greater yield on investment.

Asset market comprises of the stock and commodity markets which provide leading signs of the direction of the currency. These signs are sourced from media coverage and reports generated by participating firms in these markets. For example, a sudden sell-off in the stock market could be an indication of an eminent economic downturn which can affect the value of a currency. Also, in a commodity-based economy, an increase of exported commodity prices by a country could lead to appreciation of her currency. A common indicator for analysing these markets include Stock Market Index, Commodity Index, Consumer Price Index and many more (Degiannakis and Filis 2019).

Political and social landscape of a country also influence the strength of a currency. Fundamental analysts always keep abreast with such information to evaluate the impact on Forex rate. Example of events that they pay close attention to includes election outcomes, climatic stability, government debt, diseases outbreaks, conflicts, wars and so on (Attigeri et al. 2015; Remias 2021).

Fundamental Indicators

There are several fundamental indicators available to modern Forex traders (Nti, Adekoya and Weyori 2019). Their exhaustive listing and description are beyond the scope of this thesis, howbeit we describe some commonly used ones.

Balance of Payment (BOP) is the booking-keeping of all transactions (import, export, investments) between a country and the rest of the world during a specific period e.g., quarterly, annually, etc (Kenton 2021). Balance of trade, the total export value net total import value, is the main component of BOP. A positive balance of trade occurs when the total value of goods and services that domestic producers vend to foreign countries surpasses the total value of foreign goods and services that domestic consumers purchase and vice versa. BOP cannot be used as the only indicator to forecast exchange rate or trend reversal because it is common for policymaker, motivated by different goals for their currency to intervene through economic policies and tariffs(Bernanke 2017; Habib, Mileva and Stracca 2017). Therefore, BOP is interpreted in the context of macroeconomic policies in place around the same period (Wong et al. 2019).

Consumer Price Index measures the weighted average of prices of a basket of consumer goods and services. It is a frequently used measure for identifying periods of inflation or deflation which at extreme levels can influence exchange rates (Giannellis and Koukouritakis 2013).

Gross Domestic Product is the total market value of all the finished goods and services produced within a country in a given time period (Fernando 2021). It is a lagging indicator to confirm economists' assessments of long term trends. In Forex fundamental analysis, it can be used to assess the impact of currency fluctuation on domestic production.

buy-and-hold is a trading strategy that fundamental analysis traders employ (Du Plessis 2012; Tun 2020) and it is common to benchmark technical analysis based strategy against it.

Buy-and-hold

Buy-and-hold (BandH), a common benchmarking strategy for performing comparison test of trading strategies, is a forbearing investment strategy used by long-term investors (Yam, Yung and Zhou 2009). Investors buy an asset (e.g., US dollars in exchange for British pounds) and hold it for a long period of time, without being concerned about short-term price movements or market volatility (Stasinakis and Sermpinis 2014). The advantages of BandH include fewer fees, lesser commission and tax benefits, which can add up to net investment. BandH investors focus on building a portfolio of shares or currencies that will potentially grow over time using passive elements, such as dollar-cost averaging and index funds (Shiryaev, Xu and Zhou 2008). BandH is not a completely risk-free investment strategy, for example, an investor who bought RIM (Blackberry) in 2008 at its all-time high price would have lost 70% of its share price by 2012 and the shares never recovered. Researchers have performed comparison tests of different trading strategies, using BandH as the benchmark and their results show that BandH strategy can be outperformed. However, there have also been cases of BandH outperforming technical analysis strategies (Neely 2003), generating returns in excess of over 10% per annum in some cases (Stasinakis and Sermpinis 2014).

Fundamental analysis works well if all market participants' logical expectations are the same, i.e., wait for the release of fundamental data about an asset before taking decisions (Achelis 2001). In reality, decisions are made in a shorter time frame by hesitant or over-reactive investors who do not have complete information to correctly value assets but react tardily or prematurely (Critchley and Garfinkel 2018). Information will eventually reach all investors and the market will adjust in the direction of fundamental analysis. However, before this adjustment takes place, technical analysis can capitalise on market inefficiency to make short-term gains by evaluating market data using appropriate indicators to find repeating patterns.

2.3.2 Technical analysis

Technical analysis are sets of techniques for examining historical market data with the aim of identify repeating patterns that can be used in predicting future market trends in the short term (Picasso et al. 2019). Technical analysis was first introduced in the 1800s by Charles Dow (Murphy 1999) for studying market inefficiency in the stock market. Since the early 1970s, after the end of Bretton Woods system of currency valuation agreement, traders have used technical analysis for forecasting in Forex markets (Bordo 2019). Technical analysis approaches can either be charting or technical indicators. Charting is the identification of patterns and meaningful pictures from visual representation of the data to aid trading decision making (Schabacker 2005). Technical indicators are mathematical calculations applied to historical data to gauge price movement and empirically confirm existence of repeating patterns. In contrast, charting is an approach that relies on instinct and experience to interpret the visual representation of historical data for pattern discovery and future predictions (Stasinakis and Sermpinis 2014).

Data

Technical analysis looks to predict future changes in the financial market by searching for repeating patterns in historical market data, mainly price and volume.(Seth 2021). In technical analysis, first a frequency when snapshots of the market are to be taken is decided. On the stroke of the chosen frequency, snapshots are taken to create a physical time series. Technical indicators are then applied to the physical time series to identify repeating patterns.

There is evidence of higher success in finding repeating patterns when technical indicators are applied to data snapshots taken in high frequency (Hongguang and

Ping 2015). High-frequency data[4], has created new opportunities for traders to find more complex patterns to base their trading strategies (Caginalp and Balevonich 2003).

Technical Indicators

As aforementioned whilst presenting fundamental indicators, the exhaustive listing of technical indicators is also beyond the scope of this thesis. Instead, the readers is directed to "The encyclopedia of technical market indicators" by Colby (2003) for a more detailed look at indicators for technical analysis. Here we present two commonly used technical indicators named: (1)Moving Average, and (2) Bollinger bands.

Moving Average (MA) is a trend following indicator for evaluating a series of price averages to determine the direction of market trends. A rising MA indicates an uptrend while a declining MA indicates a downtrend. It forms the building block for many of the other technical indicators like simple moving average, Stochastic oscillator, Bollinger bands, Exponential Moving Average, moving average convergence/divergence and many more (Macedo, Godinho and Alves 2020).

Bollinger bands is a trend following indicator to signal price move within or outside pre-defined bands known as support and resistance level. The bands are set by adding and subtracting one or two standard deviation from the moving average, indicating levels that prices are considered too high or too low relative the average price. The lower band is called support i.e., a trough level where price is expected to revert upward towards the mean price and the upper band is called resistance, a peak level where high price is expected to revert downward towards the mean price.

[4] High frequency data is a series of data snapshots collected at an extremely fine scale to create a time-series

Price moves toward the direction of the bands are as a result of buy and sell interest. If price move reaches the lower band and reverts back, crossing the average price level, it is considered an uptrend. In uptrends, trend following investors tend to target the resistance level as the peak price . Similarly, if price move reaches the higher band and reverts, crossing the average price level, it is considered a downtrend and trend following investors tend to target the support level as the trough price. Additionally, it can be used to anticipate market volatility by observing the expansion and contraction of the bands. Contraction of the bands is customarily followed by a significant price level that lies beyond the bands which can be an indication that a volatile period is approaching (Macedo, Godinho and Alves 2020).

Most of the published work in fundamental and technical analysis use historical data sampled on a physical time scale. As already explained physical time scale data is generated by first deciding on a sampling interval, then successive data points at the decided interval are captured. However, sampling data at constant intervals has the possibility of omitting important details between adjacent data points. This is due to the assumption that important market events occur constantly in time which is not always the case. For instance, assuming it is decided to sample price using daily closing price, the flash crash which occurred across US stock indexes on the 6th of May 2010 from 2:32 pm EDT till 3:08 pm EDT would be ignored as prices rebounded shortly afterwards. An alternative approach to sampling physical time data at a predetermined constant interval is intrinsic time data sampling. In this approach, data is sampled when events considered to be significant occurs in the market, obfuscating noise and enabling traders to focus their strategies on important price moves.

2.4 Event-based approach

Mandelbrot and Taylor (1967) put forward the idea that physical time scale might not be the fundamental scale for analysing market movement and proposed an event-based approach as a potential alternative. Event based approach (EBA) captures important events in price movement. In EBA, data is sampled to model discontinuous movement in the financial market by summarising changes that are considered to be significant by an observer (Glattfelder, Dupuis and Olsen 2011).

There are many techniques for transforming physical time-series data into intrinsic time-series data, examples are Perceptual Important Points (Chung et al. 2001; Chen and Chen 2016), Zig-Zag (Raftopoulos 2003; Azzini, da Costa Pereira and Tettamanzi 2010), Turning Point (Bao and Yang 2008; Yin, Si and Gong 2011) and Directional Changes (DC) (Guillaume et al. 1997; Gypteau, Otero and Kampouridis 2015). To the best of our knowledge, DC approach is one of the actively researched EBA approaches (Ao 2018; Bakhach 2018; Petrov, Golub and Olsen 2019a,b; Chen and Tsang 2020; Petrov, Golub and Olsen 2020; Adegboye and Kampouridis 2021). It has also demonstrated the ability to yield profitable returns that outperforms state-of-the-art techniques that are based on physical time technical analysis indicators (Kampouridis, Adegboye and Johnson 2017; Aloud 2016b).

In directional changes approach, data summary is generated by recording key events in the market according to a threshold θ, expressed in percentage, and pre-determined by a trader according to his or her belief of what is a significant change. Also, to the best of our knowledge, directional changes is the only event based approach that has the concept of determine the occurrence of a trend whilst the trend is ongoing. This is an interesting feature because it has the potential for forecasting trend reversal without additional statistical measures.

2.4.1 Directional Changes (DC)

Directional changes is a technique employed in transforming physical time-scaled market data into an intrinsic time-series. The idea is to identify and capture important alternating events, also known as trends, while ignoring noise and irrelevant price fluctuations. DC framework is composed of different parts as can be seen in Figure 2.2.

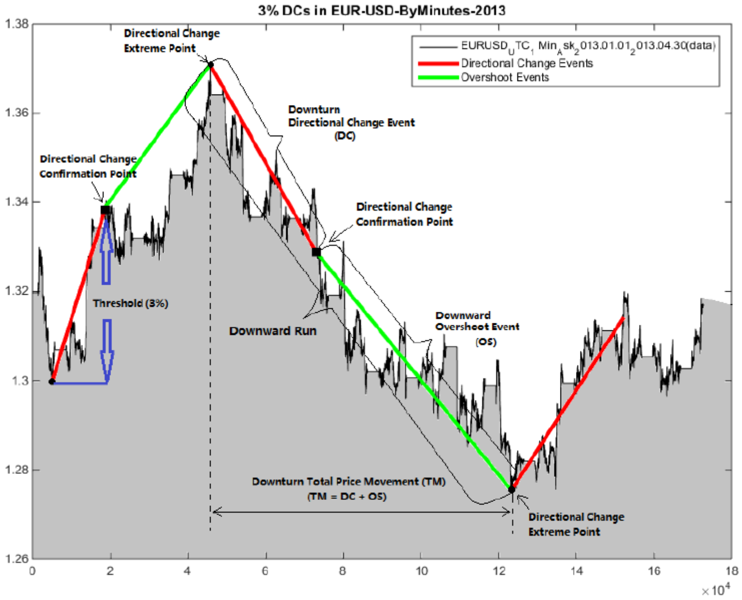


Figure 2.2: Projection of a DC events defined by a threshold $\theta = 3.0\%$. Source: (Tsang et al. 2017)

DC event

A directional changes event highlighted with red lines in Figure 2.2 is characterised by a scalable threshold that price needs to exceed to be considered significant. The threshold is a value expressed in percentages and specified by investors according to their belief of what a significant price change is. When price reaches or surpasses the threshold, the start and end of DC event are recorded, while fluctuations between these two points are ignored. The start point of a DC event is called extreme point (EXT) and the end is called is called directional changes confirmation point (DCC). There are two types of DC events, known as upward DC event and downward DC event. To confirm an upward DC event, the current price level P_t is set as the last low price P_l . As price moves, if a price lower than P_l is recorded, P_l is updated with the lower price. Upward DC event occurs when the difference between the P_t and P_l is equal to or greater than the size of the specified threshold calculated using Equation 2.1. Similarly, in downward DC, the current price P_t is confirmed as the last high price P_h. As price moves, if a price higher than P_h is recorded, P_h is updated with the higher price. Downward DC event occurs when the difference between P_t and price P_h is equal to or lower than the size of the specified threshold and can be calculated using Equation 2.2.

$$P_t \geq P_l \ \times (1 + \theta) \tag{2.1}$$

$$P_t \leq P_h \ \times (1 - \theta) \tag{2.2}$$

Overshoot Event

Overshoot event (OS), highlighted with light green lines in Figure 2.2, is a period between two alternating DC events. It indicates the continuous impact a prior DC event has on price movement beyond the point when it was first observed. For example in 10 mins high frequency data, if there are two DC events 13:00-13:40 and 15:00-15:15. The period between 13:40 and 15:00 is considered as the OS event of the preceding DC event. Two types of OS events exist, downward OS event that follows a downward DC event, and an upward OS event that follows an upward DC event. We discovered that not all DC trends are composed of DC and OS events and it was possible to have DC event followed by another DC event in the opposite direction. In these cases, which we discuss in Chapter 5, immediately after the DC trend is observed, the trend is reversed and OS event can be considered to have a length of zero or not to exist.

Algorithm 2.1 Pseudocode for generating directional changes events given threshold θ.

```
Require: Initialise variables (event is Upturn event, $p^h = p^l = \mathrm{p}(t_0)$, $\Delta x_{dc}(Fixed) \geq$
    $0, t_0^{dc} = t_1^{dc} = t_0^{os} = t_1^{os} == t_0$ )

if event is Upturn Event then
    if $p(t) \leq p^h \times (1-\theta)$ then
        $event \leftarrow Downturn\ Event$
        $P^l \leftarrow p(t)$ //Price at end time for a Downturn Event
        $t_1^{dc} \leftarrow t$            //End time for a Downturn Event
        $t_0^{os} \leftarrow t+1$        //Start time for a Downward Overshoot Event
    else
        if $p^h < p(t)$ then
            $p^h \leftarrow p(t)$ //Price at start of Downturn event
            $t_0^{dc} \leftarrow t$            //Start time for Downturn event
            $t_1^{os} \leftarrow t-1$        //End time for a Upturn Overshoot Event
        end if
    end if
else
    if $p(t) \geq p^l \times (1+\theta)$ then
        $event \leftarrow Upturn\ Event$
        $P^h \leftarrow p(t)$ //Price at end time for upturn event
        $t_1^{dc} \leftarrow t$            //End time for a Upturn Event
        $t_0^{os} \leftarrow t+1$        //Start time for a Upturn Overshoot Event
    else
        if $p^l > p(t)$ then
            $p^l \leftarrow p(t)$ //Price at start time for upturn event
            $t_0^{dc} \leftarrow t$            //Start time for a Upturn Event
            $t_1^{os} \leftarrow t-1$        //End time for a Downturn Overshoot Event
        end if
    end if
end if
```

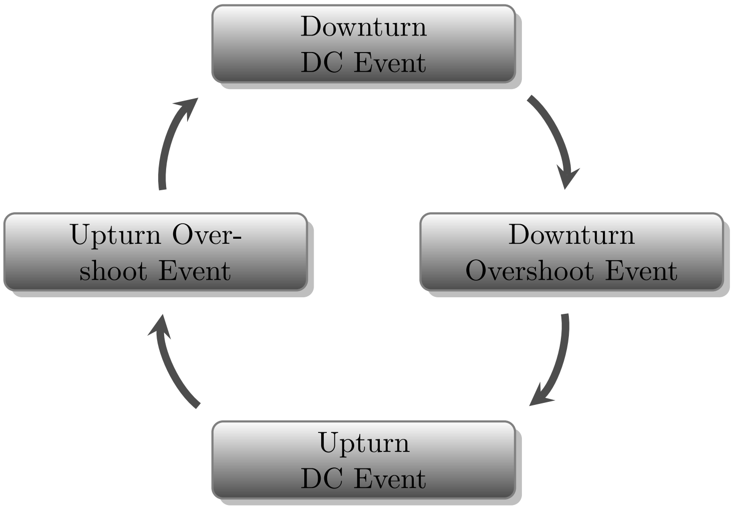


Figure 2.3: DC framework summarising price movement in a four-event cycle.

Directional Changes Run

Directional changes trend (DCT) is an epoch between two successive DC extreme points. As can be seen in Figure 2.2, it is the sum of DC and OS event lengths shown with adjacent red and green lines. A DCT can either be downward or upward. Algorithm 2.1 shows the pseudocode of how DC trends are recorded from physical time-series. The algorithm uses Equations 2.1 and 2.2 to record successive, alternating DC/OS events from the physical time-series. Figure 2.3 is a flowchart that depicts the sequence of how the events are captured. DCT is detected in hindsight at the DC confirmation point after it has already started. Therefore, Algorithm 2.1 determines the end of the previous DCT only after the next DCT is confirmed. After the next DCT is confirmed, the previous DCT's OS event region is determined also in hindsight.

2.4.2 Directional Changes scaling laws

Scaling laws, refers to properties of an object that does not change even if certain variables that describes the object are scaled up or down. This concept, already established in the fields of physics, and mathematics was first pioneered in the financial market by (Mandelbrot 1967; Mandelbrot and Taylor 1967). The financial market is a complex system composed of multitude of attributes that influence price movement. The exact impact of an individual attribute's influence on price movement is still unknown and simple deterministic models are unable to reproduce them (Cont, Potters and Bouchaud 1997). However, certain properties have been empirically discovered from historical price movement and accepted as truth due to their statistical consistency across different snapshot size of historical data and different financial market types. A number of scaling laws have been discovered in DC event series specifically (Sewell 2011; Tsang, Tao and Ma 2015; Tsang et al. 2017). These laws, 46 in total and described in Section 2.5.1, are used in building a profile of general DC price evolution.

$$OS_t = 2 \times DC_t \tag{2.3}$$

$$OS_m = DC_m \tag{2.4}$$

In one of the scaling laws, it was observed that if a DC event is snapshot with threshold θ, the magnitude m of DC event is on average equal to magnitude m of the following OS event (Glattfelder, Dupuis and Olsen 2011). Similarly, it was also observed that if a DC event takes t amount of physical time to complete, the corresponding OS event on average takes twice the amount of time ($2t$) (Glattfelder, Dupuis and Olsen 2011). These two observations shown in Equation 2.3 and 2.4 are critical in forecasting expected trend reversal points. In this work, we explore Equation 2.3 further by investigating for equations that better express the relationships between DC and OS event lengths.

2.5 Related works in DC

We divide this section into two categories: (1) related research works on DC scaling laws, and (2) related research works on the application of DC scaling laws in forecasting DC trend reversal, monitor volatility and develop trading strategies.

2.5.1 Scaling Laws Discovery

Scaling laws have been discovered in DC event time series. Some of these laws include the average directional change tick count as a function of the directional change threshold, the average length of an overshoot after a DC event is confirmed, the average magnitude of overshoot after a DC is confirmed, and many more (Glattfelder, Dupuis and Olsen 2008). These properties are used by traders to empirically monitor volatility and trends and forecast trend reversal to achieve their financial goals. There has been advancement in the discovery of scaling laws in DC approach. To be best of our knowledge, there are 46 DC scaling laws and Table 2.1 presents them in chronological order.

Table 2.1: DC scaling laws discovery

Author and year	Number of Scaling laws
Glattfelder, Dupuis and Olsen (2008)	17
Glattfelder, Dupuis and Olsen (2011)	12
Bisig et al. (2012)	1
Aloud and Fasli (2013)	4
Aloud (2016c)	5
Tsang et al. (2017)	4
Ma et al. (2017)	1
Tsang and Chen (2018)	1
Wang and Wang (2021)	1

2.5.2 DC Trend Reversal Estimations and Trading

A trend is the perceived tendency of financial markets to move in a particular direction over time (Fontanills and Gentile 2002). Trend reversal is a change in the direction from upwards to downwards or vice-versa. A successful trading framework is expected to identify trading opportunities that minimise risk and maximise profit. A trend reversal estimation algorithm is a crucial component of the framework. Table 2.2 presents a summary of DC works that used discovered scaling laws in trading strategies or trend reversal forecasting algorithms. The work by Aloud, Tsang and Olsen (2014) focused on gaining insight into market activities through the understanding of the dynamics of how a human trader makes trading decisions. The behaviours of interest included traders' profit objective, risk appetite and limit order criteria. They modelled this behaviour into artificial agents that emulated human traders. One of the challenges encountered by Aloud, Tsang and Olsen (2014) is the complexity involved in emulating human trader interactions in the Forex market. They argued that this was because of the heterogeneous nature of human traders' behaviour and the asynchronous nature of the Forex market. Aloud, Tsang and Olsen (2014) therefore used individual traders' historical transactions to model agent's behaviour. To trade, the agents employed a strategy called ZI-DCT0. The characteristics of ZI-DCT0 are, (1) random choice by the agent of either trend following or contrarian trading strategies, and (2) random choice of thresholds to create event series for trading by the agents. They reported to have successfully constructed an agent-based model of the Forex market. However, comparison result to similar work was not presented. In a subsequent work, a trading strategy called ZI-DCT1 was proposed as improvement to ZI-DCTO (Aloud 2016a). The new strategy incorporated a model to dynamically select tailored DC threshold that captures the most significant events in a physical time series. Comparative trading results between ZI-DCT0 and ZI-DCT1 showed that ZI-DCT1 was more profitable. To further improve profitability and accuracy at forecasting DC trend reversal, a trading framework (DCT2) with adapting threshold capability was proposed by Aloud (2016b). In this approach, they separated DC event series in upward trend event series and downward event series. This way, separate thresholds can be selected for the subgroups. The strategy adapted to market conditions to remain profitable by updating threshold as the trading session progressed. The average return on investment of ZI-DCT, ZI-DCT1 and DCT2 were 4%, 30% and 58% respectively.

Bakhach et al. (2016) proposed a dynamic DC-based trading strategy (contrarian) called 'DBA'. In the approach, they arbitrarily selected a threshold for transforming one minute physical time series in a DC event series. Two parameters DBA_up and DBA_down were introduced at uptrend and downtrend respectively. DBA combined these parameters with the scaling law on the ratio between DC event length and OS event length to anticipate trend reversal. To trade, positions are opened within the OS event region if the magnitude of price change is greater than the parameter value and less than estimated end of the OS event. Open position are then closes at the confirmation point of the following DC event. To select the best real values for these parameters, 100 different values were experimented from 0.01 to 1.00, with a step size of 0.01. The forecasting algorithm was experimented in three Forex markets, EUR/CHF, GBP/CHF and EUR/USD. Positive returns were reported in the markets and buy-and-hold, a comparative strategy yielded negative returns. As a follow-up work to 'DBA', a strategy called "Intelligent Dynamic Backlash Agent" (IDBA) was proposed by Bakhach et al. (2018). IDBA adopted the same trading rules as DBA. It incorporated learning modules to manage risk (maximum drawdown) and moderate quantity traded per transaction. In addition, to overcome the limitation in DBA, a preliminary threshold selection step was introduced. The tailored threshold was chosen from a range from 0.1% up to 2.5% with a step size of 0.1. The enhancements to the trading framework resulted in improvement in profit and risk. Reported comparison result to DBA showed that total profit was tripled while risk (maximum drawdown) was reduced by around 300%.

A DC trend forecasting classifier was proposed by Bakhach, Tsang and Ng (2015). The goal was to establish the estimative power of the DC approach. To this end, they proposed 3 new DC indicators derived from a technical indicator. These indicators were used in forecasting price value at OS extreme point. The accuracy reported was around 70%. Another DC trend reversal forecasting classifier was proposed by Bakhach, Tsang and Jalalian (2016). The goal was to forecast DCCs of a DC event series sampled using a larger threshold. To this end information from event series sampled using smaller threshold was used. The idea was based on the hypothesis that extreme points of an event series sampled with smaller threshold can also be found in the event series sampled with larger threshold and that the DC trends in the smaller threshold series are confirmed before those in the larger threshold series. The goal was therefore transformed into a classification problem with two attributes, (1) A Boolean dependent variable assigned a true value if an extreme points in a

Table 2.2: A comprehensive list of existing directional changes works on trend forecasting algorithms and trading strategies works

Author and year	**Summary**
Kablan and Ng (2011)	**Dataset:** EUR/USD, AUD/USD, GBP/USD, USD/CHF, and USD/JPY. **Aim:** Trading period forecasting with sensitivity to intra-day volatility. **Result:** Outperforms Buy-and-hold and linear forecasting .
Aloud, Tsang and Olsen (2014)	**Dataset:** EUR/USD **Aim:** Develop a trading strategy (DCT0) based on agent models that mimic human traders **Result:** Agent resembled human trader to a certain degree.
Aloud (2016a)	**Dataset:** EUR/USD. **Aim:** Develop a forecasting algorithm and strategy (DCT1) that outperforms DCT0s. **Result:** Return on investment is significantly larger.
Bakhach, Tsang and Ng (2015)	**Dataset:** EUR/USD and Gold price. **Aim:** Forecast DC trend reversal points using J48Graft and M5P. **Result:** Average recall, precision and accuracy of both algorithms was 0.658909091, 0.643954546,0.687636364 respectively over 11 quartiles.
Gypteau, Otero and Kampouridis (2015)	**Dataset:** Stock price of Barclays, Marks & Spencer, NASDAQ and NYSE. **Aim:** Develop a multi DC-threshold strategy with Genetic Programming. **Result:** Outperformed single threshold strategy.
Bakhach, Tsang and Jalalian (2016)	**Dataset:** EUR/CHF, GBP/CHF, and USD/JPY. **Aim:** Predict DC magnitude. **Result:** Mixed results. Accuracy of up to 80% in some cases and unable to outperform dummy strategy in others.
Aloud (2016b)	**Dataset:** Saudi Arabia stocks - SAMBA, SABB, RAJHI, STC and ZAIN. **Aim:** Develop an adaptive trading strategy (DCT2) with capability to switch thresholds over trading period and compare with DCT0 and DCT1. **Result:** Average return on investment was 65% better than trend follow DCT0.
Bakhach et al. (2016)	**Dataset:** EUR/CHF, GBP/CHF and EUR/USD. **Aim:** Develop a trading strategy with dynamic selection of thresholds. **Result:** Profitability comparison outperformed Buy-and-hold. Risk-adjusted-return comparison outperformed EURO STOXX 50.
Ye et al. (2017)	**Dataset:** GBP/USD and EUR/USD. **Aim:** Develop a trading strategy that combines technical and DC scaling laws. **Result:** Comparison to strategies based on DC only scaling law was not significant.
Alkhamees and Fasli (2017b)	**Dataset:** FTSE 100 index. **Aim:** Identify and forecast trends in data streams using dynamic threshold. **Result:** Same day match of detected trends to published news headlines.
Alkhamees and Fasli (2017a)	**Dataset:** FTSE 100 index. **Aim:** Develop dynamic threshold trading strategy. **Result:** Outperformed a single threshold trading strategy.
Kampouridis and Otero (2017)	**Dataset:** EUR/GBP, EUR/USD, EUR/JPY, GBP/CHF, and GBP/USD. **Aim:** Evolve multi-threshold trading strategies. **Result:** Statistically outperformed a single threshold trading strategy, other multi-threshold trading strategies, a technical analysis based strategy and buy-and-hold at 10% significance level.
Bakhach et al. (2018)	**Dataset:** AUD/CAD, AUD/USD, GBP/CAD, GBP/NZD, NZD/USD, and EUR/NZD. **Aim:** Develop dynamic threshold trading strategy incorporated with additional order size and risk management functionality. **Result:** Rate of return and maximum-drawdown result statistically outperformed a predecessor trading strategy.
Bakhach et al. (2018)	**Dataset:** FTSE 100, Hang Seng, NASDAQ 100, Nikkei 225 and S&P 500. **Aim:** Develop trading strategies based on mean OS length and Median OS length.. **Result:** Average rate of return ranged from -14.93% to 62.60%.
Palsma and Adegboye (2019)	**Dataset:** EUR/GBP, GBP/CHF, GBP/USD and EUR/USD. **Aim:** Develop multi-threshold strategies with Particle Swarm Optimization and Shuffled Frog Leaping Algorithms. **Result:** Average return of around 0.01% by Particle Swarm Optimization.

larger threshold series coincides with an extreme point in the smaller threshold series, and (2) A real value independent variable, the overshoot event value in the larger threshold series. J48 algorithm is then used in creating a model that classifies the DCC point of larger thresholds once the DCC of the smaller threshold is confirmed. The classification model recorded an accuracy of around 81%.

Bakhach, Tsang and Raju Chinthalapati (2018) proposed the idea of embedding the forecasting model proposed by Bakhach, Tsang and Jalalian (2016) in a contrarian trading strategy called TSFDC. Bakhach, Tsang and Raju Chinthalapati (2018) proposed generating separate forecasting models for uptrends and downtrends. Depending on the market and DC event type experimented, average rate of return was between 4% and 81% and maximum drawdown was not worse than – 6%. Bakhach, Tsang and Raju Chinthalapati (2018) also compared the results to those reported in Kampouridis and Otero (2017) (another DC-based strategies compared) and concluded that TSFDC performed considerably better.

Ye et al. (2017) proposed 4 types of strategies with similar trade opening approach (at the DCC point) and different trade exit timing. The first two strategies used traders experience to decide when to close opened trades. Strategy one used a limit order to close opened trade while strategy two used a trailing stop order. It is however unclear how the limit order and trailing order levels were set. Opened trades using the third strategy were closed according to a DC trend reversal estimation algorithm. The fourth strategy, an extension of strategy three combined DC with Directional Movement Index, a technical analysis indicator to measure the strength of the trend. Trades opened using the fourth strategy were closed using the same approach as in strategy three if the strength of the trend is above a certain level specified in the technical indicator. Ye et al. (2017) experimented the strategies in two markets (EUR/USD and GBP/USD) and with 10 thresholds, the first five thresholds were from 0.01 to 0.09, with a step size of 0.02 and the second five thresholds were from 0.1 to 0.9, with a step size of 0.2. Experimental result showed that strategy two was most profitable across the markets and thresholds while strategy four, a combined DC trend reversal estimation algorithm and technical indicator strategy was the least risky. Maximum drawdown (MDD) was used to measure risk. Strategy three recorded the second-best profitable strategy and the second least risky strategy. Their result shows that trading using DC trend reversal forecasting techniques can yield profitable result at comparatively low risk.

Kablan and Ng (2011) proposed a neuro-fuzzy-logic based trading strategy that can capture the volatility in DC trends. A future price estimation algorithm was embedded in their trading framework to predict the future price of assets based on the current price and the immediate past 3 consecutive observations in the market. The trading strategy's returns outclassed the physical-time scale trading strategies they compared with.

Alkhamees and Fasli (2017a) highlighted a problem in summarizing price movements based on single fixed threshold over a long physical-time period. They argue that assuming a threshold of 0.01% is used in summarising events, if overtime significant events level drops to 0.009%, the events will not be captured. Alkhamees and Fasli (2017a) recommended summarising events over a shorter physical time frame and recalibrate threshold size for new event summaries. They proposed to generate event series daily with dynamically adjusted thresholds size according to current and previous day price movement. Comparison results showed that trading on event series generated in shorter time frame with dynamic threshold was more profitable than trading on event series generated using fixed threshold over longer periods. Similar conclusion was reached by Alkhamees and Fasli (2017b) having explored the same idea of generating event series using dynamically adjusted thresholds in data-stream.

Evolutionary computation based trading frameworks under DC approach have also been proposed in the literature. Gypteau, Otero and Kampouridis (2015) proposed a genetic programming (GP) based strategy that combined trend recommendations from multiple thresholds. They argue that sampling DC event series from physical time series with different thresholds generate different event series. Therefore, for the same physical time data point one threshold can detect a downward trend and another detect upward trend. The strategy was used in combining trend recommendation from multiple thresholds so that strongest trend is selected through GP evolution. The leaf nodes of the GP tree were Boolean values representing the type of DC trend detected by randomly selected thresholds. Upward trend resolved to TRUE while downward trend resolved to FALSE. The inner nodes were logical operators { AND, OR, NOX, XOR and NOT}. The GP strategy combined the leaf values using the logical operators to evolve a multi-threshold trend recommendation. If the recommended trend was an upward a sell action is triggered otherwise a buy action was triggered. The Strategy was tested on four datasets, two stocks and two international indices and compared with results from trading using individual thresholds. Results showed that combining thresholds was more profitable than trading on single threshold.

Kampouridis and Otero (2017) observed that the DC-OS event length ratio does not always follow the average 1:2 ratio originally proposed by Glattfelder, Dupuis and Olsen (2008) and, instead ranged between 1.8 and 2.0 of DC event length. Kampouridis and Otero (2017) therefore proposed tailoring the DC-OS event length ratio to datasets. Like the approach proposed by Aloud (2016b), DC event series was split into upward event series and downward event series and separate ratios calculated for each subtype. To anticipate trend reversal point, DC event lengths known at the

DCC points were summed with estimated OS event length deduced using the appropriate subtype ratio. The trend reversal forecasting algorithm was then embedded in a genetic algorithm based multi-threshold trading strategy that took trading decisions by optimising trend reversal point recommendations from multiple thresholds. They used two different types of datasets to evaluate the strategy; 1) tick data and 2) intra-day data at 10-minute intervals and compared its trading result to those from other DC based strategies and physical-time benchmarks. On tick data, the mean return of the optimised strategy outperformed all other strategies. On intra-day data at 10-minute intervals, the mean return of the optimised strategy outperformed all physical-time benchmarks and all other DC based strategies but one. Howbeit, the mean return result from that other DC strategy was not statistically significant.

To the best of our knowledge, the only comparative study of optimised multi-threshold strategy in DC was carried out by Palsma and Adegboye (2019). They compared strategies where trading recommendations were optimised using genetic algorithm, particle swarm optimization and shuffled frog leaping algorithms respectively. In the work, the average return by particle swarm optimization was highest. However, the performance was neither across all datasets nor statistically significant. We are therefore of the opinion that further investigation is required in this kind of comparative study.

We are able to conclude from the works reviewed that (1) successful trend reversal forecasting algorithms and profitable trading strategies can be developed using the DC approach, (2) systematic selection of threshold size captures significant event better than arbitrary selection and this was noticeable in the profit reported, (3) trading strategy based on multiple recommendation is more profitable than single threshold based strategies, (4) evolutionary algorithm techniques have shown to be a promising approach for optimising recommendations from individual thresholds.

We observed that majority of existing works in DC used parametric models in expressing the relationship between DC and OS event length. As we could see, the relationship discovered in Ye et al. (2017) was a finite parameter exponential function and others discovered finite parameter linear functions. We also observed in previous works on the assumption that a DC trend is always composed of DC and OS event. Inspection of event series sampled with some thresholds indicated that this is not always the case even when the stylised fact that on average OS event length is twice DC event length holds. Nonetheless, estimating OS event where none exists could lead to incorrect prediction and undesired trading outcomes. In addition, majority of the works achieved positive returns at minimum risk (i.e., works that reported on risk metrics). However, it is unknown whether the result can be generalised because experimentation was carried out using limited datasets.

We are thus, motivated to explore further the question of forecasting trend reversal according to the stylised fact on the relationship between DC event and OS event length. We explore for non-parametric models that can express richer relationship between DC event length and OS event length.

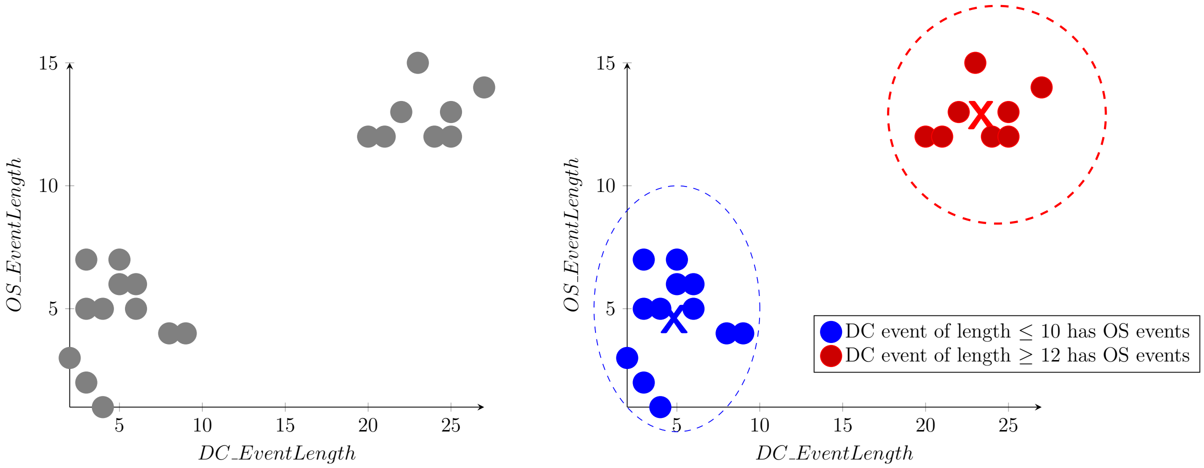
Chapter 3

Machine Learning

Machine learning (ML) research focuses on designing algorithms for building problem solving models that can learn, adapt and in some cases improve over time according to new signals received from their external environment (Holmes, Donkin and Witten 1994). ML-based models are well-employed approaches for financial forecasting and for developing trading strategies (Dymova, Sevastjanov and Kaczmarek 2016; Huang, Chai and Cho 2020; Sezer, Gudelek and Ozbayoglu 2020; Dixon, Halperin and Bilokon 2020). ML algorithms are commonly grouped into two categories namely unsupervised and supervised learning. The difference between the two groups are the learning goals and types of input variables.

Unsupervised Learning

Unsupervised learning is a group of machine learning techniques for finding patterns or correlating anomalies in data without specifying a target attribute (Hastie, Tibshirani and Friedman 2009) or reward to guide the learning (Sathya and Abraham



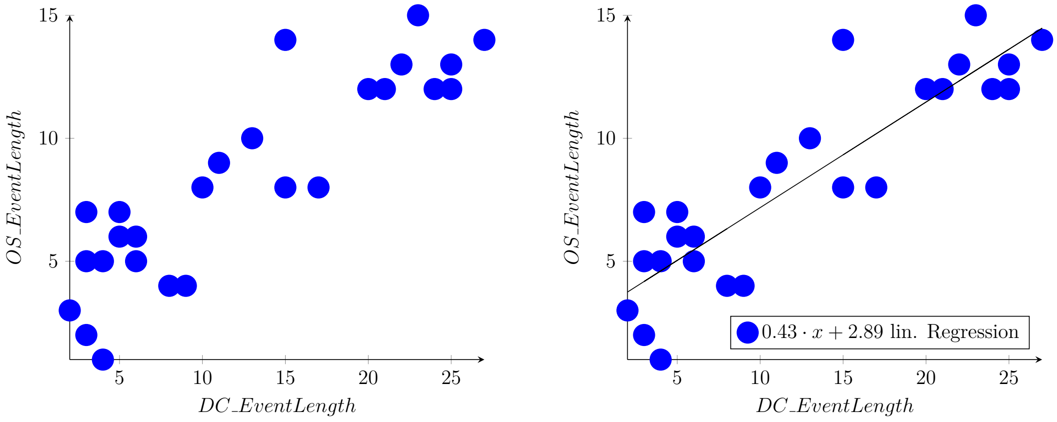
(a) (b)

Figure 3.1: In (a) an example of DC and OS event length data from DC trend snapshot in an event series; (b) potential output of an unsupervised learning algorithm, where the data is grouped into 2 clusters to depicted DC events that have corresponding OS event.

2013). It aims to describe relationships amongst a set of attributes. Some examples of common unsupervised learning algorithms are K-mean, Local Outlier Factor, Isolation forest and self-organising map. Unsupervised learning can be used in answering a question like “what types of DC events have a corresponding OS event (i.e., the OS event length is zero) “? Figure 3.1a illustrates the scattered graph of a DC event and OS event length attributes of a given DC event series. If we apply an unsupervised learning algorithm, it could produce an outcome such as in Figure 3.1b which can be interpreted as ”DC events having lengths between 0 and 10 or greater than 18 have OS event”.

Supervised Learning

Supervised learning is a group of machine learning techniques for discovering a model that maps the relationships between predictor attribute(s) and target attribute(s).



(a) (b)

Figure 3.2: In (a) an example of DC event and OS event lengths data from DC trend snapshot in an event series; (b) potential output of a supervised learning algorithm, that shows a linear equation of best fit that represents the relationship between DC event length and OS event length. Attribute X in the linear equation represents DC event length known at DCC point.

The aim is to use the discovered model in predicting similar relationships in unseen data. Examples of supervised learning tasks are classification and regression. Linear regression, support vector machine, ID3 and evolutionary algorithms are examples of techniques for solving supervised learning tasks (Kotsiantis et al. 2007). Figure 3.2a shows a graphical representation of a dataset from which we are interested in learning the relationship between DC event length and OS event length. Figure 3.2b shows a fitted linear equation function representing the discovered relationship which can be used in predicting future OS event lengths considering DC event length is the predictor attribute.

In our study, the nature of our problem falls within supervised learning remit. We are interested in: (1) predicting whether a DC event is followed by an OS event; this constitutes a classification problem, and we thus provide more information on classification problems in Section 3.1, (2) predicting the length of an OS event; this constitutes a symbolic regression problem, and we provide a discussion on this in Section 3.2, and (3) optimising trend reversal point recommendation from multiple thresholds which constitutes a numerical optimisation problem, and we provide more information about this concept in Section 3.3. To tackle above problems, (1) we used an automated machine learning tool presented in Section 3.4 to build a classifier that categorised into DC event is followed by an OS event and others followed by another DC event, (2) we used Genetic Programming (GP) to evolve a symbolic regression model for estimating OS event length after DC event is confirms, and (3) and we used Genetic Algorithms (GA) to solve out optimisation problem. GP and GA Evolutionary Computation (EC) algorithms. We thus provide a brief description of EC in Section 3.5, and then detail the specific characteristics of GP and GA in Sections 3.5.2 and 3.5.3, respectively. Section 3.6 concludes the chapter.

3.1 Classification

Classification is a type of machine learning problem solved by developing prediction models from patterns discovered in historical observations. The aim is to categorise similar patterns into predefined groups, called classes, in unseen data. Each recorded observation is represented by properties called predictor attributes and target attributes. There are four common groups of predictor attributes namely, categorical, continuous, discrete, and ordinal. Categorical attribute has unordered finite set of values, for example, set of colours. Continuous attribute has infinite number of numeric real values, for example, DC event length, OS event length and so on. Discrete attribute has enumerable number of values between a lower and an upper bound, for example the number of DC event in a DC series. Ordinal attribute has ordered finite set of values, for example, the test grade of an exam A, B, C, D, E and F, where

A is considered the highest and descending to the lowest F. Target attributes are properties we are interested in understanding from historic observations and seeking to categorise in future observations.

Table 3.1: Dataset for Loan application classification.

Age group	Gender	Housing	Annual Salary	Marital Status	Dependency	Loan Amount	Approve Loan (Class)
20s	Male	Owned	50K	Married	3	20K	Yes
30s	Male	Owned	35K	Married	2	5K	Yes
20s	Male	Renting	10K	Single	3	30K	No
50s	Male	Renting	75K	Single	1	20K	Yes
40s	Male	Renting	0	Married	2	25K	No
50s	Female	Renting	0	Single	0	10K	No
50s	Female	Renting	0	Married	0	20K	No
50s	Female	Renting	50K	Married	1	20K	Yes
60s	Female	Renting	60K	Married	1	100K	No
20s	Male	Renting	35K	Single	0	40K	No
60s	Female	Owned	100K	Married	1	20K	Yes

To further illustrate attribute categories, Table 3.1 shows a dataset of applicants applying for a loan at a financial institution. In the table, there are 7 predictor attributes: The first attribute "Age group" {20s, 30s, 40s, 50s, 60s} is an ordinal attribute; the fourth, sixth, seventh and eighth attributes are continuous attribute; the second and third and fifth attributes, "Gender" {Male, Female} and "Housing" {Owned, Rent}, "Marital Status" {Married, Single} are categorical attributes; the target attribute is the "Approve Loan" {Yes, No}. The common steps in creating classification models from such dataset are data pre-processing, attribute selection and classification algorithm selection and hyper-parameterisation (Beniwal and Arora 2012). In the pre-processing step, activities such as data integration, data cleaning and discretization are carried out first. Then, a subset of predictor attributes, relevant to the classification task is selected. This step is advantageous at reducing over-fitting, reducing model complexity and increasing model creation speed. Attribute selection techniques can be divided into filter and wrapper methods (Karegowda, Manjunath and Jayaram 2010). The filter method uses statistical methods like information gain and correlation between each predictor attribute and the target attribute as criteria for attribute selection. The wrapper method uses cross-validation in training classifiers on subsets of predictor attributes at a time. The subset of attributes that generate the best performing classifier is selected for the remainder of the modelling process. The wrapper methods tend to perform better because consideration is given relationships amongst predictor attributes (Wah et al. 2018). Nevertheless, it can be costly in a high-dimensional dataset.

Many classification algorithms have been proposed in the literature, including nearest-neighbour methods, decision tree induction, error back-propagation, reinforcement learning, lazy learning, rule-based learning, Bayesian learning and so on (Ali and Smith 2006). With the vast number of classification algorithms, the choice of algorithm for a given classification problem remains a challenge. This is because it is possible for Algorithm A to outperform algorithm B on a certain classification problem 1, at the same time, it is also possible for Algorithm A to underperform algorithm B on a different classification problem 2 (Wolpert, Macready et al. 1995). Algorithm performance is usually measure based on the percentage of correct classification and computational complexity. The finding is, there isn't a universal classification algorithm that is the most accurate across all datasets (Michie, Spiegelhalter and Taylor 1994). To evaluate and select the best performing classifier, a common approach is the cross-validation approach after which the algorithms' performances are ranked according to percentage of correction classification and weighted F-measure (Ali and Smith 2006; Reif et al. 2014). After selecting a classification algorithm, it is important to configure the algorithm's parameters, called hyper-parameters, to control the learning process so that an optimised model that balances between accuracy and generalization is obtained (Maher and Sakr 2019). Due to different possible combinations for these hyper-parameter values, optimisation techniques are employed to find a good set of values (Braga et al. 2013; Feurer, Springenberg and Hutter 2015; Bergstra et al. 2011). The final step in a classification problem is modelling, which can either be black-box modelling or white-box model. White-box models differ from black-box models in terms of interpretability. Besides recognising patterns in unseen data, the inner workings and mapping of predictor attributes to the target attribute in white-box models are understandable to practitioners.

3.2 Regression

Regression is another type of machine learning problem solved by developing prediction models from patterns discovered in historical observations. It differs from a classification problem in two ways, the nature of the target attribute and the objective. In regression problems, the target attribute is continuous, and the goal is to assign a real value to unseen data. For example, a company could use regression models to forecast its potential percentage growth in a financial year from models developed using past spending on adverts. Regression problems models can be grouped into 2 main categories, namely parametric and non-parametric regression (Mahmoud et al. 2021).

3.2.1 Parametric Regression

In parametric regression, the form that the equation describing the relationship between predictor and target attributes is known (Mahmoud et al. 2021). The task is to find the best coefficients for the attributes of an equation. The modelling techniques in this category are simple, fast and models can be created from small dataset. An example of a parametric modelling approach is Linear regression. Assuming we have a dataset $\{x_{i1}, x_{i2}, ..., x_{im}, y_i\}_{i=1}^n$ of n observations, each observation represented by m predictor attribute values and y_i the target attribute value, a linear regression model takes the form of Equation 3.1.

$$y_i = \beta_0 + \epsilon + \sum_{j=1}^{k} \beta_j x_{ij} \tag{3.1}$$

where β_0 is the y-intercept, that is, the mean of the target attribute when all predictor attributes values are set to zero; k is the number of predictor attributes used in representing n observations; β_j is the linear regression coefficient for the j-th predictor attribute. x_{ij} are the j-th predictor attribute values for the i-th instance, and ϵ known as residuals is a random error that cannot be explained by the model. The model searches the space of β_j values so that the mean squared error (MSE) of observed and predicted target attribute is minimized (Xiao 2015).

3.2.2 Non-Parametric Regression

In non-parametric modelling, the form that the equation that maps predictor attributes to the target attribute is unknown. The task is to find the structural form of the regression function using large amount of data which we show in Equation 3.2.

$$y_i = f(x_i) + \epsilon_i \tag{3.2}$$

Different from parametric regression, we do not assume the number of parameters the equation should take, neither do we assume that the data follows a normal distribution. One of the techniques for finding an acceptable form of the equation is symbolic regression. The idea is to explore a space of possible mathematical equations that might fit the data, in search of an equation that best describes the said relationship. It is impossible to know if we have found the optimal equation, so the procedure continuously searches the space of candidate equations until a predefined stop criteria is reached, or the algorithm converges at a local optima.

3.3 Numerical Optimisation

In numerical optimisation, the task is a search for the best fit values (discrete or continuous) from a large pool of configuration to assign to attributes under certain constraints to obtain the best possible outcome. The best possible outcome could either be to maximise or minimise an objective. Constraints are set of conditions that input variables are required to satisfy. Optimisation problems are commonly written in the form maximise $f(x)$, where f is the objective function and x is the input variable whose optimal value and we expect an optimiser to discover so that the objective function value is improved. Figure 3.3 is a simple example of a such function where the objective is to find an x value that maximises equation $-x^2-x-8$ which is -0.5, that is highlighted with a green dot and a constraint could be that $X > -5$.

In real life unfortunately, finding global maxima is not as simple, the objective function can consist of multiple attributes and the optimisation surface can have a range of local maxima which makes finding the global maxima challenging (Dorigo and Stützle 2004). A common method for finding the optimal values of input attributes is gradient descent, an algorithm that describes the slope of a function telling whether it increases or decreases in a certain direction towards the objective. In a maximisation problem the algorithm takes iterative steps in the direction of the gradient of the function until maxima point is reached. Some of the challenges of gradients are (1) choice of learning rate - rates that are too small rate can lead

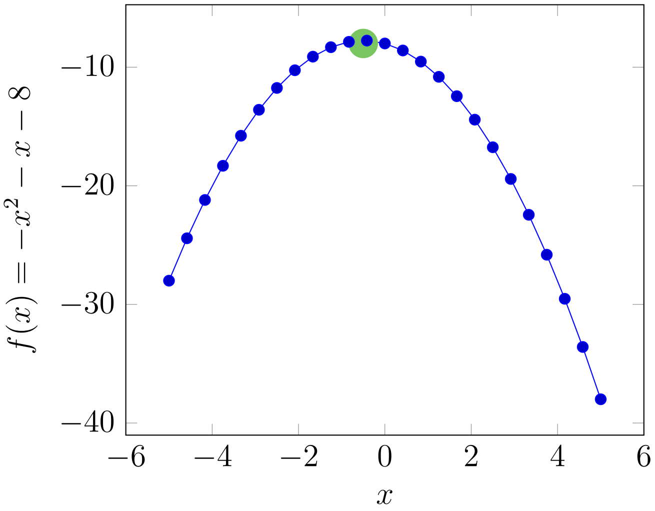


Figure 3.3: A graph of an equation where the objective is to find the global maxima which we illustrate with a green dot.

to slow convergence or hinder convergence and (2) entrapment in numerous local optima in a complex optimisation surface (Ruder 2016).

An alternative practice is to use meta-heuristic algorithms (Metaxiotis and Liagkouras 2012), a method for finding a near-optimal solution in a reasonable computational time. Meta-heuristic algorithms can either be based on single solution or population based. Algorithms that are based on single solution like local search and simulated annealing focus on exploitative search within a local neighbourhood that is defined according to certain criteria. Population based algorithms like ant colony optimization and genetic algorithm are explorative in nature, therefore rely on the diversity of the population for success. They are non-deterministic, optimal solution is not guaranteed and they have a poor theoretical foundation (Carr 2014). Nevertheless, they have proven to be successful in practise. Population based numerical optimisation algorithms share some similarities with non-parametric regression algorithms, they do not have an upper bound on search duration or provide an indication of when a discovered solution is close to the optimal solution, therefore a stop criteria and a fitness function is required.

3.4 AutoML

The efficient selection of classification algorithm for a given problem and the tuning of the hyper-parameters requires poses a challenge as it requires knowledge and experience (Khanmohammadi and Rezaeiahari 2014). Advances have been made in creating tools that automate the learning process so that non-expert can automatically create well-performing prediction models. These tools, part of a growing research area called AutoML, automatically perform the steps of non-deterministically selecting optimal classification algorithm and tuning the hyper-parameters. It is usually the case that the only parameter a user is required to specify is the time limit for the tool to evaluate the best classifier and configuration (Kotthoff, Thornton and Hutter 2017). Examples of such tools are meta-collaborative filtering framework (Smith et al. 2014), Hyperopt-sklearn (Komer, Bergstra and Eliasmith 2014), Auto-sklearn (Feurer et al. 2015), TUPAQ (Sparks et al. 2015), Auto-WEKA (Thornton et al. 2013), DAUB (Sabharwal, Samulowitz and Tesauro 2016), MLPlan (Mohr, Wever and Hüllermeier 2018) and more recently Auto-Keras (Jin, Song and Hu 2019). In this work, we use Auto-WEKA for selecting and tuning our classifier because it provides an application programming interface (API) that can easily be incorporated into our DC trend reversal estimation framework facilitating the creation of custom model for each dataset we experimented and it is based on Weka (Hall et al. 2009), a well-known open source package for performing classification tasks.

3.4.1 Auto-WEKA

Weka (Hall et al. 2009) supports 39 classification algorithms that can potentially be used for a classification task. However, the performance of the algorithms can vary from one dataset to another. Therefore, Auto-WEKA framework was introduced. It uses advances in high-dimensional stochastic optimisation to fully automate the process of choosing the best classification algorithm and optimising its hyper-parameter values. Auto-WEKA can be executed in two modes, namely single-threaded and multi-threaded. We have chosen to perform our experiments using single-threaded mode due to limitation of available hardware resources. The default execution time for Auto-WEKA is 1 minute but to get the best result from Auto-WEKA, tuning is required to determine the appropriate execution time. Because the search space (i.e., 39 algorithms) exploration is non-deterministic and result depends on the initialisation seed, it is recommended to execute Auto-WEKA multiple times with different seeds, resulting in as many recommended classifiers (Tighe, Lewis-Morris and Freitas 2019; Basgalupp et al. 2020). From the pool of recommended classifier, the best classifier can then be selected according to a fitness measure.

3.5 Evolutionary Algorithms (EA)

Evolutionary Algorithms are techniques that are based on Darwin inspired biological evolutionary theory of natural selection to preserve the fittest individuals for the procreation of future generations. The fittest individuals' traits are passed on to successive generations, resulting in improvement in these traits over the generations. EA makes use of biological evolutionary concepts such as reproduction, mutation, recombination, and selection. EAs search for an optimal or near-optimal acceptable

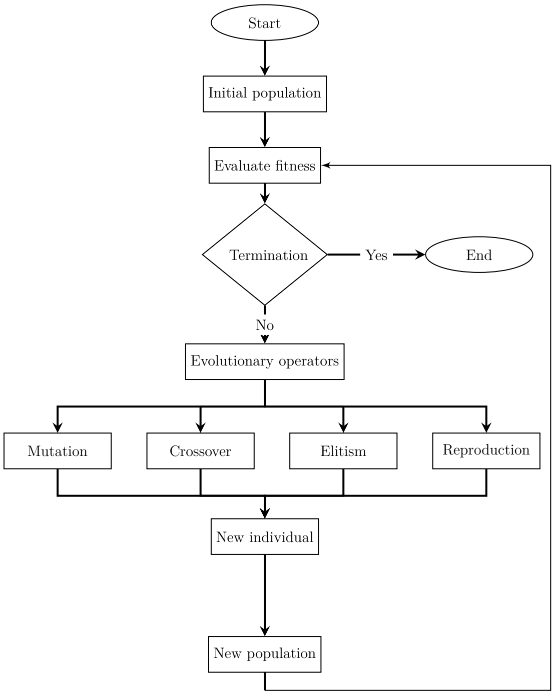


Figure 3.4: An illustration of EA framework.

solution to optimisation problem by choosing the best-performing solution from a group of solutions that were stochastically created and evolved over several generations. Some examples of EA algorithms are evolutionary programming (Fogel, Owens and Walsh 1966), differential evolution (Storn and Price 1997), genetic programming (GP) (Koza 1992), genetic algorithm (GA) (Holland 1992) and evolution strategy (Rechenberg 1973). EAs have a common framework in evolving new individuals namely initialisation, fitness evaluation, selection strategy, evolutionary operations and stopping criteria.

3.5.1 EA Framework

Figure 3.4 presents the building blocks of an EA framework. The process initialises by stochastically creating an initial population of solutions called individuals. The aim is to have a diverse population that covers a wide range of potential solutions to a given problem. As a next step, the fitness of the individuals are evaluated with a fitness function to determine how well the individuals perform a set out objective. If an individual performs sufficiently well, the process terminates, and the individual is presented as an acceptable solution. Otherwise, an evolutionary process is initiated and genetic operators applied to selected parent individuals that evolve new individuals, called offsprings. Offsprings constitute a new population that are expected to have better traits for meeting the set out objective. If the new population is not able to solve the problem sufficiently, the evolutionary process continues until a termination criteria is reached. The genetic operators that are applied are mutation, crossover, and reproduction. During evolution, weak individuals in the population are eliminated and the best individuals known as elites are preserved across generations. Preservation of best performing individuals across generations follow the principle of survival of the fittest. Tournament selection is a common method used in identifying parent individuals with desired traits for evolution.

Fitness function

Fitness function is a mechanism that describes the objective that individuals are expected to fulfil (Poli et al. 2008). It is used to guide the EA algorithm towards regions in the search space with potentially better solutions. An example of a fitness function is root mean square error that measures the difference between a value predicted by an individual and the value observed.

Selection strategy

Similarly to natural evolution, individuals are selected for breeding based on their fitness level. It is based on the idea that selecting parents with high fitness level will breed offspring with improved fitness level (Liu et al. 2017). Several selection strategies can be employed, and tournament selection is commonly used. It involves choosing random individuals from the population as candidates for breeding. A contest is then held the fittest individual amongst these candidates is selected for breeding an offspring. Another selection strategy is the roulette wheel. A probability is assigned to an individual based on its fitness level in relation to the sum of the fitness level of all individuals in the population. The probability of the individuals is then normalized whereby individuals with higher probability have a greater chance of getting selected in a random selection. Truncation Selection is another selection strategy. In this case, individuals are ordered according to their fitness level. Candidate individuals for evolution are randomly selected from individuals with fitness level higher than a cut-off point (Poli et al. 2008).

Genetic operators

The main evolutionary operations can be categorised as follows:

- **Crossover** is a binary operation that recombines two parent individuals by taking parts of their chromosomes to produce a child individual. The aim is to exploit regions in the search space where promising solutions have been found already.

- **Mutation** is a change in the genetic structure of a single individual. The aim is to stir the algorithm towards exploring different regions in the search space.

Mutation can occur at single or multiple sites within the gene structure of an individual.

- **Reproduction** is the random copy of a program from one generation to the next without any modification. The aim is to maintain diversity in the population.

Elitism

Elitism is a selection operator for preserving the highest fitness level individual(s) in future generation. This is done to improve performance by reducing the time spent rediscovering previously found partial solution. Though it speeds up convergence, it can reduce population diversity if allowed to dominate the population.

Stopping criteria

Stopping criteria indicates when the evolutionary process should end. Different qualities of the population can be used to determine stopping criteria. Three examples of stopping criteria are when: an upper bound threshold on the number of generations is attained, a lower bound in step size improvement from one generation to another and an upper bound threshold on the number of fitness function evaluation is reached (Safe et al. 2004). Once the process ends, the individual with highest fitness level in the latest generation is returned as the best solution.

As a reminder, in this work we aim to (1) estimate as accurately as possible OS event length once a DC event length is known and use the two lengths to forecast trend reversal, and (2) optimise forecasting models of multiple DC thresholds to improve trading. To this end, we utilise genetic programming (GP) and genetic

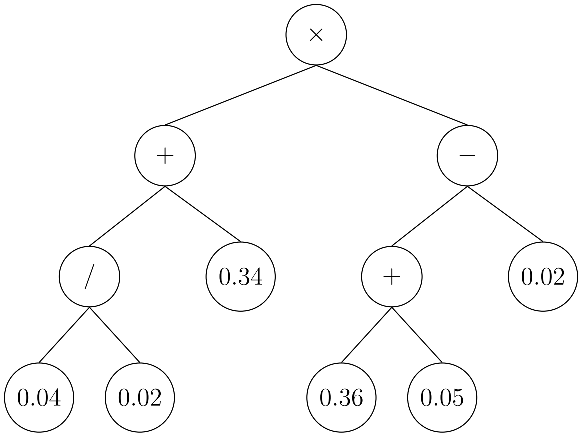


Figure 3.5: An illustration of a GP Tree. The terminal nodes of the tree are 5 DC thresholds 0.04, 0.02, 0.34, 0.36 and 0.05. The non-terminal nodes are mathematical operators $\times, -, +, /$

algorithm (GA). These two techniques are state of the art algorithms for solving symbolic regression problems and solving optimisation problems respectively.

3.5.2 Genetic Programming

GP is an evolutionary algorithm for evolving computer programs[1] to solve a problem without specifying explicit information on how the problem should be solved or what the structure of the program should look like in advance but follows the process illustrated in Figure 3.4 (Koza 1992).

Representation

A typical GP program is composed of variables, constants and functions and can vary in shape and size. A common way of representing a GP individual is a tree representation (Koza 1992). Figure 3.5 presents a sample GP tree. The GP tree

[1] A computer program in this context is a structure that represents a structure e.g., boolean expression, mathematical equations.

is composed of interdependent components known as the terminals and functions. Although tree representation is the most common, other forms of representing GP exist such as linear and cartesian representation (Nordin 1994; Miller 1999). Linear GP is a flattened tree that expresses a sequence of programming instructions to mimic how computer architecture represents programs. Cartesian GP is a directed graph representation of programs. It is based on the idea of creating and evolving genetic representations of electronic circuits (Miller 2019).

Terminals set are symbols representing the end of a branch. They consist of independent variables and constants. Constants are numeric values, either integer or real (e.g., 4, 3.14). Independent variables represent input values to the program. For example, in an equation such as $x^2 + 4x + 5 = 0$, x is an independent variable. Another type of terminals are functions with no argument. For example, Rand() that returns a different value on each call and Ephemeral random constants (ERC), a set of randomly generated terminals that retain their values across the population at initialisation and during evolution.

Function set (internal nodes of the tree) are symbols representing operations. The function set consists of different expressions that define permissible relationships between internal nodes, and between internal nodes and terminal set. The number of child nodes connected to an internal node is based on the number of operands the function in the internal node can manipulate. For example, the binary functions "+, −, ×, ÷" in Figure 3.5 can have exactly two child nodes. Other types of functions include mathematical functions ($\sin, \cos, \tan, LOG, EXP$), logical operators ($AND, OR, NOT$), conditional operator (If-Then-Else), comparison operators ($\leq$, $\geq$, <, >, =, !=) and other domain specific functions.

GP property

For GP to successfully find the solution to a given problem, it must satisfy certain properties, namely closure and sufficiency (Koza 1992):

- **Closure** is the syntactic correctness of a GP tree and the correct handling of exception that may occur in rare cases. Closure can be divided into type consistency and evaluation safety. Type consistency is the property of a function set where each function is required to generate outputs that are valid inputs for other functions in the set. This ensures that the evolved GP trees are syntactically correct at compile time. If a function does not fulfil this requirement, it can be constrain limiting the number of functions it can receive input from or send its output to as input. Evaluation safety refers to the syntactic correctness at runtime. A GP program must be able be to executed from start to finish successfully without crashing. If this is unavoidable, exception handling can be introduced in such cases (Poli et al. 2008). In this work, we enforced closure through function design so that the ones that can potentially output NaN (not a number) and Inf (infinity) were protected. For example, division operation was protected by returning a specific value when the denominator was equal to zero.

- **Sufficiency** is the completeness of a primitive set so that it can express a solution to a problem. This is domain-specific, and the user must ensure that the terminal and functional sets supplied to a GP algorithm is powerful enough to evolve a valid approximation of the optimal solution (Koza 1992).

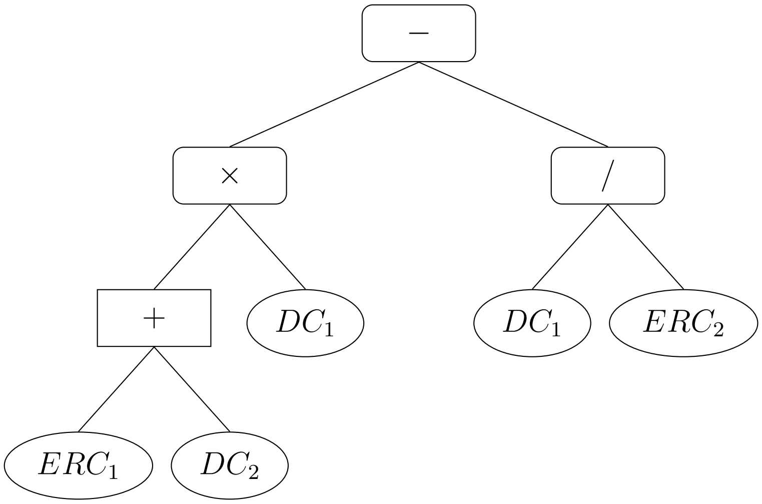


Figure 3.6: An example tree showing the Grow initialisation. The tree terminals are directional changes thresholds and ERCs while the non-terminal nodes are mathematical operators.

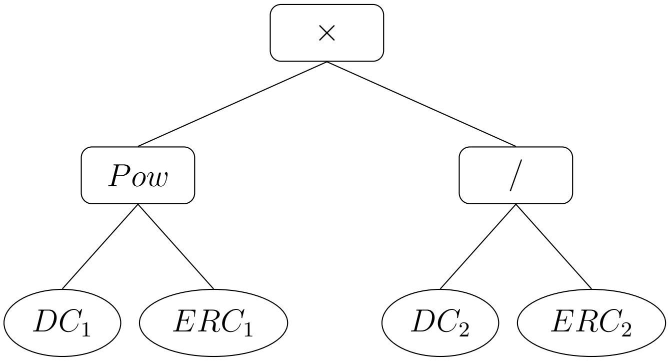


Figure 3.7: An example tree showing the Full initialisation. The tree terminals are directional changes thresholds and ERCs while the non-terminal nodes are mathematical operators.

GP Initialisation

GP population is initialised by randomly combining terminals and functions to form individuals. The aim is to have a diversified set of individuals that sufficiently represent the solution space (Poli et al. 2008). Two of the common approaches in generating an initial population are grow and full methods (Koza 1992). There is also ramped half-and-half initialisation, which is a combination of the grow and full methods (Poli et al. 2008). In all three methods, the generated individuals must not exceed a maximum depth specified upfront. Maximum depth is the number of edges traversed between the root node and every leaf node (Poli et al. 2008).

In the grow method, unbalanced and asymmetric trees of varied sizes and shapes are created. At first, a root node is randomly chosen from the primitive set (i.e., function or terminal), if the root node is a terminal, the tree is not grown further. On the other hand, if the root node is a function, the tree branches out according to the arity of the function at the root node. The branches are filled with randomly selected functions or terminals. Branches of terminals are closed out and branches of functions are recursively grown until terminals is selected or the maximum depth is reached. If maximum depth is reached and the furthest node is not a terminal, a terminal is mandatorily selected. An example of such tree is presented in Figure 3.6 which represents an equation that estimates OS event length as $OS_l = ((ERC + DC_l) \times DC_l) - \frac{DC_l}{ERC})$.

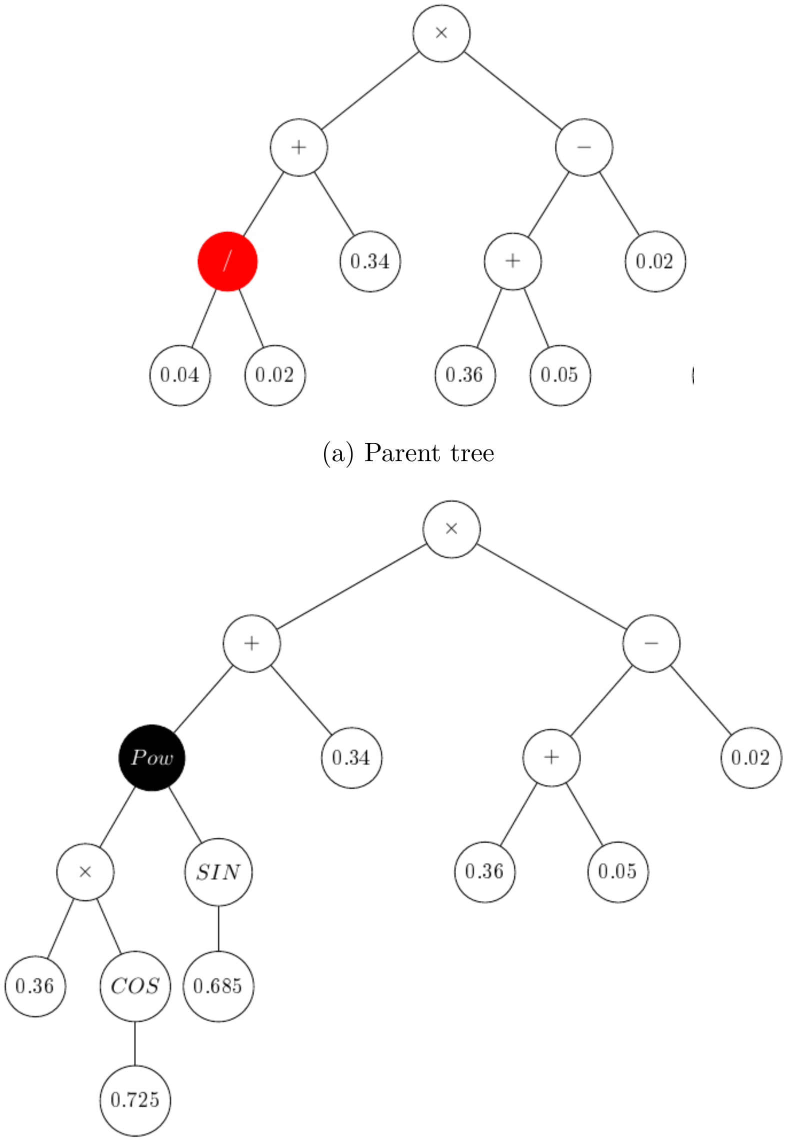
The full method creates balanced and symmetric trees of potentially different shapes and sizes (Poli et al. 2008). It is initiated by deciding on a depth then a root node is randomly selected from a function set. The tree branches out according to the arity of the function at the root node. The inner nodes are filled with randomly picked functions from the set recursively until the maximum depth is reached. At maximum depth, a terminal is randomly selected, and recursion ends. An example of such a tree is presented in Figure 3.7 which represents an equation that estimates OS event length as $OS_l = ((DC_l^{ERC}) \times \frac{DC_l}{ERC}) - (\frac{ERC}{DC_l}) + (ERC \times ERC))$.

The third type of initialisation is ramped-half-and-half, a combination of grow and full initialisation which ensures greater diversity of program population. Half of the tree population is created using grow and the other half is created using full. This ensures higher structural diversity of the trees in the initial population to improve search space coverage (Koza 1992).

Genetic operators

These are operators that lead GP process towards a solution to a problem during evolution by exchanging genetic material in a program with new/different ones. The main types of operators are crossover, mutation, and reproduction. An operator is randomly selected for evolving a tree. Excessive use of just one operator might not yield desired result. Reproduction alone leads to copies of the same program in the population. Crossover alone leads to early conversion and a locally-optimal might be selected as the best program. Mutation alone will cause GP to make big jumps without exploring its neighbourhood of solutions sufficiently. Reproduction is a random copy of a GP individual from one generation to the next without changes to the tree. More detail about crossover and mutation operator is provided below.

Mutation is a genetic operation that changes the genetic structure of a single tree. The aim is to maintain diversity in the population by exploring different regions in the search space. In Figure 3.8, we present sample mutation operation, this mutation is known as subtree mutation. A new offspring is created by removing a random node which we highlight in red and its substructure from a tree. The



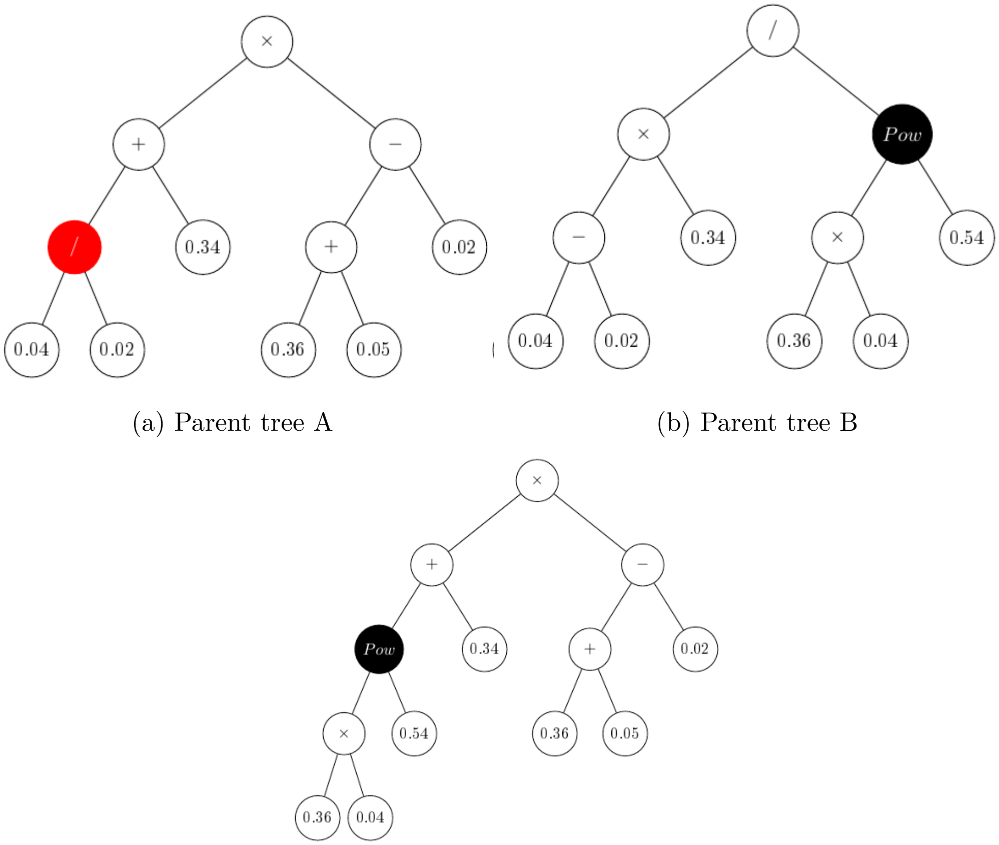
(a) Parent tree

(b) Offspring tree

Figure 3.8: An illustration of how an offspring tree is evolved with the mutation operator. Terminals 0.04, 0.02, 0.05 are DC thresholds and rest of the terminals are ERC. The function set are /, +, −, *Pow*, *Sin*, *Cos* and ×

removed node is replaced with a randomly generated tree. The new program in Figure 3.8b is then copied to the next generation. Another form of mutation is point mutation, a single node is replaced with a different member of the primitive set with the same arity and return type. It is not possible to list all the different types of mutation operator. Nonetheless, we acknowledge other types of mutation such as Size-fair subtree mutation (Langdon 1998), Hoist mutation (Kinnear 1994), Shrink mutation (Angeline 1996), Permutation mutation (Koza 1992; Maxwell and Koza 1996), Constant mutation (Schoenauer et al. 1996) to mention a few.

Crossover is a binary operation that creates a new tree (i.e., offspring) by combining copies of two selected parent trees. To select each parent, candidate parents are stochastically selected from the population, then a tournament is held amongst them and the winner of the tournament emerges as a parent tree. Crossover points are points of division where a subtree is selected from both parent trees. For example, in Figure 3.9, crossover points in parent trees A and B, which we highlight in red and black are selected. The subtree at the crossover point in parent tree A is removed and replaced with the subtree at crossover point in parent B. The new tree as can be seen in Figure 3.9c is a new offspring and it is copied to the next generation. The type of crossover described in this example is known as subtree crossover (Poli et al. 2008). There are several types of crossovers cited in the literature like, one-point crossover (Poli et al. 2008), two-point crossover (O'neill et al. 2003), uniform crossover (Poli et al. 2008), context-preserving crossover (D'haeseleer 1994), size-fair crossover (Langdon 2000), depth-based crossover (Harries and Smith 1997) and so on. There are also crossover operators that are based on GP representation such as ripple crossover (O'neill et al. 2003) for grammatical evolution and a real-valued inspired crossover (Clegg, Walker and Miller 2007) for Cartesian GP.



(a) Parent tree A

(b) Parent tree B

(c) Offspring tree

Figure 3.9: An illustration of how an offspring tree is evolved with the crossover operator. Terminals 0.04, 0.02, 0.05 are DC thresholds and rest of the terminals are ERC. The function set are /, +, −, *Pow* and ×

3.5.3 Genetic Algorithm (GA)

Genetic Algorithm (GA) is an Evolutionary algorithm to search in a solution-space for either the optimal or the near-optimal solution to an optimisation problem (Atilgan and Hu 2018). An example of such problem is finding the appropriate weight to assign to thresholds in a multi-threshold trading strategy.

Representation

A common form of representing GA is an array of binary bits. The gene array in this representation holds the value of either 0s or 1s as can be seen in Figure 3.10. Other forms of representation exist such as integer-value representation, real-value representation, permutation representation and so on. The determining factor in choosing a representation is the type and number of attributes that define the problem to be solved. Once determined, the gene type and chromosome size are fixed throughout the initialisation and evolution phases.

0	1	0	1	···	1

Figure 3.10: A sample representation of a binary bit genetic algorithm. The array of bits is known as a chromosome and each cell in the array is a gene.

GA initialisation

GA population is commonly initialised randomly, heuristically or a mixture of both. In random initialisation, random values are assigned to chromosomes of the gene. The idea is to have a diverse population that is representative of the search space. Heuristic initialisation, on the other hand, encodes the population based on prior knowledge of how to solve a problem. For example, specific values can be assigned to the chromosomes. This kind of initialisation requires care to avoid the population from being homogenous, which could inhibit the GA from solving the optimization problem. The third approach is the combination of random and heuristic initialisation where only a hand full of individuals are initialised heuristically and the rest are randomly initialised.

Genetic operators

Genetic operators lead GA towards an optimal solution to a problem through the exchange of genetic materials in a GA population. Similar to GP, the main types of operators are crossover, mutation, and reproduction.

Mutation is used to keep genetic diversity during evolution to prevent convergence to a locally-optimal solution. Mutation is done by modifying one or more gene values of a chromosome. Different ways of mutating the genome of a GA individual exists. In a binary problem represented with 0s and 1s, mutation can be done by flipping the genes of the individual as illustrated in Figure 3.11. Others are mutations are uniform mutation, Interchanging mutation, reverse mutation, boundary mutation and so on (Sivanandam and Deepa 2008).

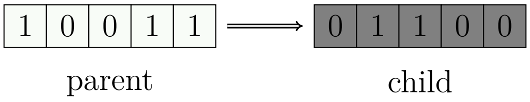


Figure 3.11: An illustration of GA evolution using a flip mutation. 0 bit genes are flipped to 1 and vice-versa.

Crossover is used to implements a search within a region in the solution space where individuals with high fitness levels have already been found. Crossover is done by selecting two parent individuals from a mating pool. Then mating sites from the parent individuals are selected at random along their lengths. Then values at the mating sites in both parents are swapped to generate the new offspring. Traditional GA uses a single point crossover illustrated in Figure 3.12.

In single point crossover, genes after a mating site in one parent are replace with genes after mating site in a second parent. Other types of crossovers are two point

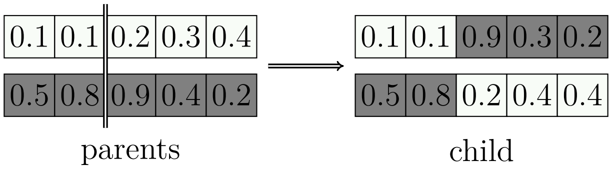


Figure 3.12: An illustration of GA evolution using a single point crossover. Genes after the mating site in one parent are replaced with those from the second parent. Either of the children can be selected for the next generation

crossover, uniform crossover, shuffle cross and so on (Sivanandam and Deepa 2008).

3.6 Summary

In this chapter, we briefly discussed two types of machine learning algorithms: unsupervised and supervised learning algorithms. We discussed supervised learning algorithms, where our interest lies. We detailed the two main supervised learning tasks: classification and regression. Both tasks find patterns in predictor attributes that express how they relate to a target attribute. The goal in classification task is to predict a categorical value, while in regression the goal is to predict a continuous value.

Furthermore, we discussed techniques that can be applied to tack the three problems studied in this work: our three supervised learning task: AutoML for classification task, i.e., classify DC trends into two categories; consisting of DC and OS events or only DC events; symbolic regression GP for non-parametric regression task, i.e., evolve a function that estimate OS event length given a DC event length; and GA for optimisation task i.e. evolve an optimised multi-threshold trading strategy.

Chapter 4

Symbolic Regression forecasting model

Directional Changes (DC) is an alternative approach for sampling price movement and described in Chapter 2.5, the occurrence of a DC event is known only in hindsight. Additionally, the length of the corresponding OS event remains unknown until the next DC event in the opposite direction is confirmed i.e., when the trend is reversed. The challenge in trend reversal forecasting from a DC perspective is the ability to determine the length of OS event before the next DC event is detected. There has been an empirical study that investigated regularities i.e., scale invariant properties in DC summarised data (Glattfelder, Dupuis and Olsen 2011). One of the regularities is the existence of a power law distribution between DC and OS event lengths. It was empirically observed that irrespective of the threshold used in sampling DC series, on average, if DC event takes t amount of physical time, its corresponding OS event will take twice the amount ($2t$). Such regularities provide traders with greater understanding of price movements, which can be leveraged for trend[1] reversal prediction and trade strategy modelling. This is because if investors can accurately estimate OS event length, they will be able to anticipate when a DC trend is expected to end. Thus, with this OS length regularity, investors can devise new strategies to generate profit on their investments. Other richer relationships exists between DC and OS event length albeit scale variant, which can be more benefial to investors for profit maximisation (Kampouridis and Otero 2017). This has motivated us to explore for equations that express richer relationships yet to be discovered by existing algorithms (Glattfelder, Dupuis and Olsen 2011; Kampouridis and Otero 2017).

To this end, we developed a tailored GP that creates equations to describe the DC-OS event length relationship. Our goal is twofold: (1) demonstrate that symbolic regression GP (SRGP) can be used to estimated OS event length, thus anticipating trend reversal in Forex data, and (2) demonstrate that the SRGP derived equations can improve trading profitability. We thus replace OS event length estimation equation of a DC-based trading strategy first proposed by Kampouridis and Otero (2017) with our SRGP created ones.

We acknowledge that risk is an important trading metric. However, as a first step we focus on profitability measure of the trading strategy. We analyse profitability and risk in our proposed trading strategy described in Chapter 5.

The rest of the chapter is organized as follows: Section 4.1 describes the uncovered linear relationships between DC and OS event lengths. Section 4.2 presents our methodology, where we explain in detail how we use a tailored SRGP in estimating OS event length, thus forecasting trend reversal. Section 4.3 presents the experimental setup, and Section **??** presents and discusses our findings. Section 4.5 concludes

[1] In the context of directional changes, trend can be described as sum of DC event length and OS event length. At the end of a trend, another one begins in the opposite direction. Just before the start of the new trend, is the trend reversal point of the trend that just ended.

the chapter.

4.1 Linear and non-linear DC-OS Relationships

There are two works that described the linear relationships between DC and OS event lengths, Glattfelder, Dupuis and Olsen (2011); Kampouridis and Otero (2017). Glattfelder, Dupuis and Olsen (2011) found that an OS event lasts on average twice the length of a DC event and proposed Equation 4.1 which we call Factor-2. Kampouridis and Otero (2017) confirmed the scaling law and, also discovered richer linear relationship, howbeit, not scaling invariant and proposed Equation 4.2 a tailored real value representing the observed relationship per dataset which we call Factor-M.

$$OS_l \approx 2 \times DC_l \tag{4.1}$$

$$OS_l = C \times DC_l \tag{4.2}$$

where :

OS_l = the length of an OS event
DC_l = the length of a DC event.
C = a time-varying[2] constant, which can take any positive real value, C > 0.

We propose re-writing Equations 4.2 in a general form as Equation 4.3. Using the new equation, we explore for relationships between DC and OS event lengths that could be either linear or non-linear. To discover the form and parameter values of the underlying equation, we developed a tailored symbolic regression genetic programming (GP). GP is a state-of-the-art algorithm for evolving unknown solution for

[2] value changes according to the DC trend and the DC:OS ratio in the dataset

a task, in our case a mathematical expression describing the relationship (Žegklitz and Pošík 2020; Wang, Wagner and Rondinelli 2019).

$$OS_l = f(DC_l) \tag{4.3}$$

where :

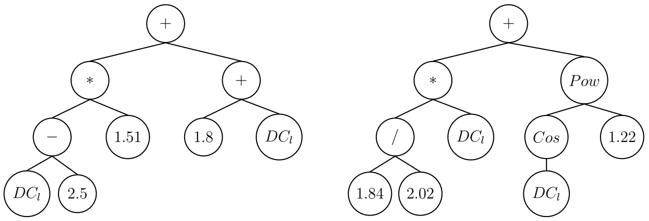
OS_l = the length of an OS event

DC_l = the length of a DC event

4.2 GP - based OS length Estimation

In this section we present the composition and evaluation of our SRGP-based OS length estimation algorithm. The algorithm is a tree based SRGP and uses as input the DC length.

First, a threshold is used in transforming physical time-based dataset into DC-based dataset. Then, the DC dataset was split into separate upward and downward DC datasets. Finally, we apply a SRGP that searches for an equation $f(DC)$ which maps DC event length to OS event length. The sum of the DC event length observed in the dataset and the estimated OS event length by our SRGP algorithm becomes our forecasted trend reversal point i.e., the start point of the next DC event in the opposite direction. In our approach there are separate equations tailored for upward DC trend and downward DC trend respectively. The rational is based on our preliminary experiment that showed varying ratio between the two types of DC trends and was also reported by Aloud (2016b). For the equation created by our SRGP to be valid, the expression must have at least one DC event length as a parameter to express the equality relationship between DC and OS event lengths.



(a) Sample tree 1 (b) Sample tree 2

Figure 4.1: Sample SRGP individual trees where internal nodes are represented by arithmetic functions. The leaf nodes are represented by numeric constants and the DC length, denoted as DC_l. Given a DC event length the tree estimates the corresponding OS length.

4.2.1 Model representation

To make the SRGP satisfy the closure property, we catch these errors using run-time exceptions and reduce the fitness of such programs. This is done to reduce the chance such programs from getting reproduced in future generations (Poli et al. 2008).

We represent our evolved SRGP individuals using tree structures. The inner nodes of the trees are composed of linear and non-linear functions. We utilised 2-arity functions {addition, subtraction, division, multiplication, power} and 1-arity functions {sine, cosine, power, logarithm, exponential}. In certain cases, it was possible for SRGP composed of division, logarithm, power and exponential functions to output undef[3] (undefined), NaN[4] (not a number) or Inf[5] (infinity). We therefore protected, these functions to ensure that valid numerical values are always returned as output by first checking for potential problems before execution and returning a default value in such cases. The terminal nodes consisted of an ERC[6] and an

[3] If Divisor is 0.

[4] If the argument of the Logarithm function less than 0, if the first argument of power function is a negative real number and the second argument is a positive real number.

[5] If the argument to the Logarithm function is 0 or the first argument to the power function is greater than 0 and the second argument is less than 0 or the argument to the exponential function is a real number greater than a certain value.

[6] Ephemeral random constants, explained in Chapter 3.5

external input to represent DC event length. We selected an ERC with a probability Pr; the DC event length with probability $1 - Pr$. All our functions and terminals are presented in Table 4.1. To initialise the population, we used ramped half-and-half technique, a technique already described in Chapter 3.5.2. Figures 4.1a and 4.1b show sample trees our GP can create. Figure 4.1a and 4.1b are trees which represent equations that estimate OS length as $((DC_l - 2.5) \times 1.51) + (1.8 + DC_l)$ and $((\frac{1.84}{2.02}) \times DC_l) + (\cos(DC_l)^{1.22})$ respectively and the DC_l in both equations represent the length of DC event.

Table 4.1: Configuration of the proposed SRGP algorithm

Configuration	Value
Function set	addition, subtraction, protected division, multiplication, sine, cosine, power, log, exponential
Terminal set	input variable (i.e., DC events length) and ephemeral random constant.
Genetic operation	subtree mutation and subtree crossover

4.2.2 Model evaluation

To evaluate our trees, we measure error between actual OS length (OS_l) and OS length that the SRGP estimates ($\hat{OS_l}$). The prediction error ε was calculated using root mean square error shown in Equation 4.4.

$$\varepsilon \ = \sqrt{\frac{\sum_{i=1}^{N}(OS_l - \hat{OS_l})^2}{N}} \tag{4.4}$$

where :

N = The sample size

ε = The root mean squared error

During evolution, we penalise by reducing fitness of tree constructs that (1) have only constants as terminal nodes since they do not use the DC event length in the estimation, and (2) estimate a negative value since OS event occurs after DC event in time and the dimension of time in financial time series is irreversible (Flanagan and Lacasa 2016). Tournament selection is used to select parents based on fitness level. If there are more than one candidate trees with the same (highest) fitness, we evaluate tree size to break the tie and the tree with the smaller depth is selected.

4.2.3 Genetic operators

We use standard genetic operators to evolve the trees. The operators used are subtree mutation and subtree crossover (see Table 4.1). The evolution was controlled by a crossover ratio P_e. At P_e we selected subtree crossover operator and at $1 - P_e$ we selected subtree mutation operator. We used an elitism ratio to select the best performing trees and carried them forward from a generation to the next without modification. To control growth, we used hard limits on the depth of offspring trees.

4.3 Experimental setup

This section presents the experimental setup to accomplish our two goals. As a reminder, our first goal is to demonstrate that the SRGP evolved equations can be used to anticipate trend reversal; the second goal is to demonstrate that accurate anticipation of trend reversal can improve profitability. This is because forecasting closely to the true trend reversal point allows traders to make better decisions that

Algorithm 4.1 Pseudocode for evolving Equation 4.3 i.e., equation to estimate OS event length given a DC event length.

```
Require: Initialise variables (PopulationSize= 500; GenerationSize= 35; TournamenteSize = 3; CrossoverRate = 0.98; MutationRate = 0.02; MaximumDepth = 3; ElitismRatio = 0.1; Prune = True )

Initialise population: $P \leftarrow$ Generate PopulationSize individuals (Candidate programs) using ramped half-and-half
Evaluate: for each $p$ in $P$, calculate Fitness with Equation 4.4
while termination condition not satisfied do
    $P_g \leftarrow$ Initialise new population for generation $g$
    Get elite individuals in $P$: $ER[1,\ldots,($ ElitismRatio $\times$ PopulationSize$)]$
    Add elite individuals to $P_g$
    for $i = ER + 1$ to $P_g$ do
        if RandomNumber < CrossoverRate then
            Select parent1: probabilistically select TournamenteSize individuals from P
            Select parent2: probabilistically select TournamenteSize individuals from P
            $P_{gi} : \leftarrow$ Perform crossover between parent1 and parent2
        end if
        if RandomNumber < MutationRate then
            $P_{gi} : \leftarrow$ Perform mutation on $P_{gi}$
        end if
    end for
    Update: $P \leftarrow P_g$
    Evaluate: for each $p$ in $P$, calculate Fitness with Equation 4.4
end while
Return the individual (i.e., equation) with the highest fitness from P
```

increases profit Chen et al. (2021). To achieve these goals, our experiments are divided into two parts. In the first part, we use our SRGP algorithm presented in Algorithm 4.1 to uncover hidden DC-OS relationships i.e., evolve OS event length estimation equations. We then compared SRGP's regression error with previous works in this field (Glattfelder, Dupuis and Olsen 2011; Kampouridis and Otero 2017), i.e., Equation 4.1 and Equation 4.2. In the second part of our experiments, we embed the equations returned by our SRGP in an existing DC trading strategy, which was first presented by Kampouridis and Otero (2017), to demonstrate that richer DC-OS relationships, improves trend reversals estimation leading to improved trading results. We compare the trading results' returns to other DC trading strategies and popular financial benchmarks, such as technical analysis, and buy-and-hold.

4.3.1 Data

We used 10-minute interval high frequency data of the following currency pairs: EUR/GBP (Euro and British Pound), EUR/USD (Euro and US dollar), EUR/JPY (Euro and Japanese Yen), GBP/CHF (British Pound and Swiss Franc), and GBP/USD (British Pound and US dollar) from June 2013 to May 2014. We considered each month in the period as a separate physical-time dataset. Datasets from the months of June and July 2013 were used for tuning OS length prediction and trading strategy algorithms. The remaining dataset from the month of August 2013 to May 2014 were used for testing the tuned algorithm. Each tuning and non-tuning DC dataset was split in 70:30 ratio for training and testing.

4.3.2 GP parameter tuning

In total, we used 50 DC datasets for tuning. To generate the datasets, we used thresholds 0.010%, 0.013%, 0.015%, 0.018% and 0.020% in summarising the 50 DC event series from 10 minute high frequency Forex data of June and July 2013, of 5 currency pairs: EUR/GBP, EUR/USD, EUR/JPY, GBP/CHF and GBP/USD (i.e $5 \times 2 \times 5$). The selected thresholds were the same as those in previous work by Kampouridis and Otero (2017) that we build on. The thresholds were reported to have been selected after a threshold tuning step but the range of thresholds were not reported. As we have not carried out the threshold tuning ourselves and the range unreported, it is possible that these values are not optimal. In addition, using the same thresholds across datasets might not take into account the peculiarity of each dataset.

Despite the undocumented threshold selection range, our first goal is to demonstrate that symbolic regression GP is able to discover scale variant richer relationship between DC and OS event length and having similar experimental setup as Kampouridis and Otero (2017) is critical for such evaluation. With the same set of threshold, we are able to have an unbiased comparison of our trend reversal forecasting algorithm to the published work. In Chapter 5 we provide our own approach for threshold selection.

Table 4.2: IRACE setup for tuning the parameters of our SRGP.

Parameter	Type	Values
Population Size	Categorical	100, 200, 300, 400, 500, 600, 1000
Generation	Categorical	25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75
Tournament size	Integer	2 - 7
Tree Depth	Categorical	3, 4, 5, 6, 7, 8, 9
Pruning	Categorical	True, False
Crossover probability	Real	0.7 - 1
Mutation probability	Real	0.1 - 0.2
Elitism	Real	0 - 0.20

Table 4.2 presents the setup for tuning the parameters of our GP using the I/F-Race package (López-Ibánez et al. 2011). I/F-Race package is an extension of an iterated racing procedures. It implements racing methods for selecting the best configuration for an optimisation algorithm. The idea is to iteratively evaluate a set of candidate configurations that are associated with a sampling distribution and empirically choose an appropriate settings that maximises an objective (Birattari et al. 2010). After each iteration, the performance of the configurations are measured

against an objective function which in our case is the root mean squared error after estimating OS event length. Candidate configurations that do not perform well enough are eliminated and those that performs well are preserved for future race iterations. Before the next iteration, the sampling distribution of the configurations is updated so that it is biased towards the best configurations and the process is repeated until a stopping criterion is met (López-Ibánez et al. 2011). For our work, we ran irace procedure with a budget of 2000 iterations. At the end of the irace run, 4 candidate SRGP configurations are presented. From the 4 candidate configurations, we choose a configuration with the highest population size because in SRGP, population diversity is important in preventing early convergence or convergence in local optima (Rosca 1995; Langdon 1996; Banzhaf et al. 1998). Table 4.3 presents our SRGP configuration after the tuning step.

Table 4.3: Regression GP experimental parameters for detecting DC-OS relationship, determined using I/F-Race.

Parameter	Configuration
Population	500
Generation	35
Tournament size	3
Crossover probability	0.98
Mutation probability	0.02
Maximum depth	3
Elitism	0.10

4.3.3 Results

To evaluate our algorithm, we used 250 DC datasets. These datasets were generated using 5 DC thresholds (i.e., 0.010%, 0.013%, 0.015%, 0.018%, 0.020%) and the 10 months dataset from 5 currency pairs described in Section 4.3.1. We observed that on the average, the ratio between DC and OS lengths varied between 1.5 and 2.5, which corroborates the scaling law findings in (Glattfelder, Dupuis and Olsen 2011). Table 4.4 presents the average of root mean squared error for these datasets. From the table we see that the Reg-GP consistently outperformed both Factor-2 and Factor-M at estimating OS length in all currency pairs. To confirm the above results, we applied Friedman's non-parametric statistical test and compared the result of the best OS event length estimation equation (control method) to others to measure statistical significance. Our null hypothesis was that the algorithms come from the same continuous distribution. The result of the statistical test presented in Table 4.5 shows Equation Reg-GP ranking the highest. The adjusted p-value at the $\alpha = 0.05$ level, according to the Hommel post-hoc[7], shows the differences in ranks of our results to be statistically significant.

Table 4.4: Estimation results by OS length estimator algorithms on 10-minute interval out-of-sample data. Results show RMSE value. They are averaged over 5 different thresholds (0.010%, 0.013%, 0.015%, 0.018%, 0.020%), 5 different currency pairs and 10 different datasets (August 2013 to May 2014).

Algorithm	Reg-GP	Factor-2	Factor-M
EUR/GBP	**6.77462**	6.82228	7.15624
EUR/JPY	**4.08026**	4.70026	4.42172
EUR/USD	**5.77218**	6.41959	6.27207
GBP/CHF	**5.86789**	6.33501	6.36392
GBP/USD	**6.09010**	6.42833	6.66513
Mean	**5.71024**	6.13421	6.16841

The above results demonstrate that Reg-GP is able to create tailored[8] equations that better describe the relationship DC and OS lengths in given datasets, and this is shown by the significantly smaller error recorded in comparison to Factor-2 and

[7] An adjustment applied to the probability of obtaining a significant difference between algorithms after testing them under similar conditions with the same out-of-sample data.

[8] The equation is different for each DC event dataset

Factor-M. Our interest now shifts to using these new equations as part of a DC-based trading strategy to investigate whether these new and improved equations can lead to increased profit margins.

Table 4.5: Statistical test results of OS length prediction according to the non-parametric Friedman test adjusted with the Hommel post-hoc (using the best method (GP) as the control (c) method. Significant differences at the $\alpha = 0.05$ level. Statistically significant ranking in **bold** text.

Algorithm	Average Rank	$Adjust_{pHomm}$
Reg-GP (c)	1.272	-
Factor-2	**2.332**	2.12E-32
Factor-M	**2.396**	6.44E-36

4.4 Trading setup and result

Having shown that Reg-GP is able to evolve equations that better describe the relationship between DC and OS event lengths, we now aim to show that accurate trend reversal forecasting impacts profitability positively. We embed our SRGP into an existing DC-based trading strategy previously presented in Kampouridis and Otero (2017). The trading strategy which we detailed next used a real-value encoding genetic algorithm to optimise trend reversal estimation recommendations of multiple DC thresholds. To test the effectiveness of our SRGP created equations, we replaced Equation 4.2 originally in Factor-M+GA with the best performing equation returned by our SRGP and called the modified trading algorithm Reg-GP+GA. We also tested the strategy with Equation 4.1 as the OS event length estimator and called this version of the strategy Factor-2+GA. Lastly, we run experiments with a technical analysis algorithm and buy-and-hold.

4.4.1 Factor-M+GA

This is the original multi-threshold based trading strategy (i.e., Factor-M+GA) by Kampouridis and Otero (2017), which used Equation 4.2 (i.e., Factor-M) to estimate OS event length. To better understand how the strategy works in practice, we describe the strategy in a single threshold scenario. The trend following strategy is designed to take trading action towards the end region of a trend, just before trend reversal in order to maximise profit. To achieve this goal, buy actions are timed as closely as possible to trend reversal point in a downtrend and sell actions to trend reversal point in uptrend. At the DCC point of trends in the DC dataset, Equation 4.2 was used to calculate the OS event length. The value of the estimated OS event length is then combined with DC event length to estimate trend reversal region. However, this was an average over a dataset and in reality, it was possible for individual DC trends in the dataset to have longer or shorter OS event length than computed by the equation. To improve OS event estimation accuracy, they calculated separate DC to OS event length ratios for upward trends and downward trends and named them r_u and r_d respectively. In addition, they created two user-specified parameters namely b_1 and b_2. These parameters were used to define a range of time within an OS event period where trading was allowed. This was done to ensure that trading actions were taken towards the end of the OS event period. Equation 4.5 presents their formulas for computing the starting and ending periods in upward and downward OS event period where trading actions were allowed. t_0^u and t_1^u were calculation for start and end trading times in upward OS periods, and t_0^d and t_1^d were calculation for start and end trading times in downward OS periods. t_1^{dc} and t_0^{dc} were start and end times for individual trend's DC event and the difference of the two values gave the DC event length. r_u and r_d, and b_1 and b_2 as afore explained were the average ratio of OS events lengths of upward and downward trends in the DC dataset, and allowed user-specified trading action range respectively.

In high frequency data, it was possible to have multiple profitable trading opportunities within an action range (i.e. from b_1 to b_2) especially in high frequency data and it will be more profitable to take action closest to peak and trough prices. A user-specified parameter b_3 was therefore created to target trend peak and trough A sell action is taken when price is equal to $P_{peak} \times b_3$ and a buy action was taken when price is equal to $P_{trough} \times (1 - b_3)$.

$$\begin{aligned} t_0^u &= (t_1^{dc} - t_0^{dc}) \times r_u \times b_1 \\ t_1^u &= (t_1^{dc} - t_0^{dc}) \times r_u \times b_2 \\ t_0^d &= (t_1^{dc} - t_0^{dc}) \times r_d \times b_1 \\ t_1^d &= (t_1^{dc} - t_0^{dc}) \times r_d \times b_2 \end{aligned} \tag{4.5}$$

To further improve trading profitablity an optimised multi-threshold GA-based trading strategy (Factor-M+GA) was proposed (Kampouridis and Otero 2017). The GA was used to optimise peak and trough recommendations from multiple thresholds and quantity traded per transaction . Algorithm 4.2 presents their GA for evolving real values for the user specified parameters (i.e., b_1, b_2 and b_3), quantity Q to trade, and weights $W_1, W_2, ..., W_{N_\theta}$, associated to the thresholds, whose recommendation are being optimised. The evolved real values were embedded in a trading strategy presented in Algorithm 4.3 and at the end of trading, the fitness of the GA population was determined using the objective function presented in Equation 4.6

$$\begin{aligned} fitnessfunction &= Return - \alpha \times MDD \\ MDD &= \frac{P_{trough} - P_{peak}}{P_{trough}} \end{aligned} \tag{4.6}$$

Algorithm 4.2 Pseudocode for optimising threshold weights, quantity and user-specified parameters.

```
Require: Initialise variables (PopulationSize= 1000; GenerationSize= 35; TournamenteSize = 7; CrossoverRate = 0.90; MutationRate = 0.10; numberOfThresholds = 5; ElitismCount = 1 , Budget= 500000)

Initialise population: $P \leftarrow$ Generate PopulationSize individuals $W1$ = $W2$= $W3$= $W4$ =$W5$ = $\frac{1}{5}$, $b_1$= rand(0.0,1.0), $b_2$= rand(0.0,1.0), $b_3$= rand(0.0,1.0), Q = rand(0, budget)
Trade: for each $p$ in $P$, trade with strategy described in Algorithm 4.3
Evaluate: for each $p$ in $P$, calculate Fitness with Equation 4.6
while termination condition is not satisfied do
    $P_g \leftarrow$ Initialise new population for generation $g$
        (a) Elitism: copy elite individuals $r$ from $P$ to $P_g$
        (b) Select: probabilistically select $(p - r)$ individuals of $P$ to add to $P_g$ and
          – Perform crossover between a pair of selected individuals according to CrossoverRate
          – Perform mutation between a pair of selected individuals according to MutationRate
    Update: $P \leftarrow P_g$
    Trade: for each $p$ in $P$, trade with strategy described in Algorithm 4.3
    Evaluate:for each $p$ in $P$, calculate Fitness with Equation 4.6
end while
Return the individual (i.e., equation) with the highest fitness from P
```

4.4.2 Factor-2+GA

This is a modified version of Factor-M+GA. In this trading strategy, Factor-M is replaced with Factor-2 proposed by Glattfelder, Dupuis and Olsen (2011) and presented in Section 4.1.

4.4.3 Reg-GP+GA (proposed algorithm)

This is also a modified version of Factor-M+GA. In this trading strategy, Factor-M is replaced with Reg-GP, an equation evolved by our symbolic regression and presented in Section 4.1.

Algorithm 4.3 Pseudocode for calculating the return of a trading strategy. Source:(Kampouridis and Otero 2017)

```
Require: Initialise variables (Cash= budget, $Q_{trade}$= 0, current = 0, $N_\theta$ = 5)
Require: Initialise PFL i.e. The amount/quantity of currency pair held
Require: $W1$, $W2$ , $W3$, $W4$ , $W5$ , $b_1$, $b_2$ , $b_3$ , $Q$ // user-specified parameters
for i =0; i < $dataset_{length}$; i++ do
    Initialise weights for buy and sell: $W_B = W_S = 0$, number of upturn and downturn
  : $N_\uparrow = N_\downarrow = 0$
    current ← current + 1
    if $P_C > P_{peak}$ then // $P_C$ is the current price and $P_{peak}$ is the highest so-far price
        $P_{peak} \leftarrow P_C$
    else if $P_C < P_{trough}$ then
        $P_{trough} \leftarrow P_C$
    end if
    for $j$ = 0; j < $N_\theta$ ; j++ do
        Calculate $t_0^u$, $t_1^u$, $t_0^d$, $t_1^d$, as explained in Equation 4.5
        if event is Downturn Event then
            $W_B \leftarrow W_B + W_j$
            if current with range of [ $t_0^d$, $t_1^d$ ] then
                $N_\downarrow \leftarrow N_\downarrow$ +1
            else
                $N_\downarrow \leftarrow N_\downarrow$ -1
            end if
        else$W_S \leftarrow W_S + W_j$
            if current with range of [ $t_0^u$, $t_1^u$ ] then
                $N_\uparrow \leftarrow N_\uparrow$ +1
            else
                $N_\uparrow \leftarrow N_\uparrow$ -1
            end if
        end if
    end for
    if $W_S > W_B$ then
        if $N_\uparrow > 0$ and $P_C \geq (b_3 \times P_{peak})$ then
            $Q_{trade} \leftarrow (1 + \frac{N_\uparrow}{N_\theta}) \times Q$
            if $Q_{trade} > 0$ then
                Cash ← Cash + ( $Q_{trade} \times P_C$)
                PFL ← PFL - $Q_{trade}$
            else
                Hold
            end if
        else
            Hold
        end if
    else if $W_S < W_B$ then
        if $N_\downarrow > 0$ and $P_C \leq (P_{trough} + (1 - b_3 \times P_{trough}))$ then
            $Q_{trade} \leftarrow (1 + \frac{N_\downarrow}{N_\theta}) \times Q$
            if Cash > ($Q_{trade} \times P_C$) then
                Cash ← Cash - ( $Q_{trade} \times P_C$)
                PFL ← PFL - $Q_{trade}$
            else
                $Q'_{trade}$ a new traded quantity, up to the amount that is affordable
                Cash ← Cash - ( $Q'_{trade} \times P_C$)
                PFL ← PFL - $Q'_{trade}$
            end if
        elseHold
        end if
    end if
end for
Wealth ← cash + $Q_{trade} \times P_C$
Return ← 100 * $\frac{wealth}{budget}$
```

Evolutionary algorithms tuned parameters

Except for buy-and-hold trading strategy, the rest of the trading strategies we tested are evolutionary based and required parameter tuning. These tuned parameters are population size, number of generations, tournament size, crossover probability, mutation probability, and number of thresholds (for the multi-threshold DC strategies). Table 4.6 presents the value for each parameter. We used the same parameter configuration for all DC based trading strategies to avoid bias. Finally, for all evolutionary algorithms, the experiments are run 50 times on each dataset and the results presented correspond to the average value over the 50 executions. The buy-and-hold strategy is run just once per dataset since it represents a deterministic strategy.

Table 4.6: Trading strategy experimental parameters determined using I/F-Race.

Parameter	EDDIE	Factor-M+GA	Factor-2+GA	Reg-GP+GA
Population	500	1000	1000	1000
Generation	30	35	35	35
Tournament size	2	7	7	7
Crossover probability	0.9	0.9	0.9	0.9
Mutation probability	0.1	0.1	0.1	0.1

4.4.4 Technical analysis

Several technical indicators exist for algorithmic trading. To combine different indicators to formulate trading strategies, we use a GP algorithm called EDDIE. The technical indicators EDDIE combines in our experiments are moving average, trade break out, filter, volatility, momentum and momentum moving average (Tsang et al. 2000).

4.4.5 Buy-and-hold

Buy-and-hold is a common benchmark for trading algorithms. A trading strategy that buys a financial product at the start of a period with the expectation that price will appreciate over time, then sells at the end of the period.

4.4.6 Comparison among trading algorithms

Table 4.7: Average return (%) results for all trading strategies. DC strategies using 5 thresholds. 10-minute interval out-of-sample data. 5 different currency pairs and 10 different datasets (August 2013 to May 2014)..

Algorithm	Reg-GP+GA	Factor-M+GA	Factor-2+GA	EDDIE
EUR/GBP	0.00703	0.00058	0.00341	0.00001
EUR/JPY	0.11600	0.06327	-0.07723	-0.00378
EUR/USD	0.00733	0.00055	0.02455	-0.00002
GBP/CHF	-0.0198	-0.00357	0.00903	0.00004
GBP/USD	-0.00629	-0.00045	-0.00580	0.00001
Mean	**0.01896**	**0.01125**	**-0.0093**	**-0.00076**

Table 4.7 presents the average monthly returns after trading five currency pairs from August 2013 to May 2014. Reg-GP+GA was the best in 2 currency pairs, second best in 1 currency pair and performed worst in two currency pairs. However, Reg-GP+GA had higher mean return than Factor-M+GA and Factor-2+GA i.e., 0.01896%, 0.01125% and -0.0093% respectively. Forex market is open 24 hours a day in different parts of the globe, therefore, the annualised return[9] of Reg-GP+GA is 0.22776%. The annualised return is relatively small in comparison to alternative investments opportunities around the same period of our sample datasets. For instance, individual savings account (ISA), a type of investment in the UK that is exempted from tax, yielded between 0.5% and 1.0% around the same period and to

[9] formula for calculating annualised return is $[(1 + return)^{12} - 1] \times 100$

address the low profit recorded, we propose a new strategy as part of our second contribution (mortgage strategy 2019).

We also compared profitability between Reg-GP+GA and physical time-based trading strategies. Results showed that returns obtained by Reg-GP+GA significantly exceeded that of EDDIE and buy and hold. In fact, EDDIE experienced negative returns in 4 out of the 5 currency pairs (EUR/JPY, EUR/USD, GBP/CHF, GBP/USD), which also resulted in a negative mean return of -0.00076%. On average, over all currency pairs analysed, buy-and-hold strategy made a mean return of 0.01274% which was 32.81% less than the mean return by Reg-GP+GA.

Table 4.8: Total number of positive months per currency pair in 10 months in % values

Algorithm	Reg-GP+GA	Factor-M+GA	Factor-2+GA	EDDIE
EUR/GBP	60%	50%	50%	60%
EUR/JPY	60%	40%	50%	20%
EUR/USD	50%	30%	80%	40%
GBP/CHF	30%	20%	70%	70%
GBP/USD	50%	40%	40%	60%
Total	**50%**	**36%**	**58%**	**50%**

Furthermore, we performed Friedman's non-parametric statistical test to evaluate the statistical significance of the returns by the evolutionary algorithm based trading strategies. The null hypothesis is that the trading strategies come from the same continuous distribution. From the result presented in Table 4.9, we observed that returns from Reg-GP+GA is ranked the highest and they statistically outperformed Factor-M+GA and $EDDIE$ at $\alpha = 0.05$ level. There was no statistical significance between Reg-GP+GA and Factor-2+GA; nevertheless, Reg-GP+GA was ranked higher than Factor-2+GA. From the statistical and mean return result of the DC based strategies, we can infer that it is more profitable to use equations that express richer relationships as estimators of OS event length in trend reversal forecasting algorithms.

Table 4.9: Statistical test of trading returns according to the non-parametric Friedman test with Homel post-hoc test (using the best strategy (Reg-GP+GA) as the control (c) . 10-minute interval out-of-sample data. Significant differences at the $\alpha = 0.05$ level

Algorithm	Average ranking	$Adjust_{pHomm}$
Reg-GP+GA (c)	1.64	-
Factor-2+GA	1.76	0.64
Factor-M+GA	**2.78**	2.01E-17
EDDIE	**3.82**	9.27E-17

Table 4.8 details the number of positive trading days over which the excess return in Table 4.7 was made. This information further shows the consistency of the trading strategies over the trading period. Reg-GP+GA is ranked second, achieving 50% positive trading days across all currencies. Factor-2+GA had the highest number of positive trading days (58%) across all currency pairs analysed, notwithstanding, result indicates that of the 5 currency pairs, Reg-GP+GA had more profitable days than Factor-2+GA in 3 (EUR/GBP, EUR/JPY, GBP/USD). Factor-2+GA incurred higher loss during the non-profitable days, and this was evident in the negative return of -0.0093%. Similarly, EDDIE had the same number of profitable days as Reg-GP+GA but loss incurred during the non-profitable days was high which led to a negative return. Finally, Factor-M+GA had the least number of profitable days (36%), the excess return was large enough to remain profitable after trading across all currency pairs, nevertheless it was not enough to surpass Reg-GP+GA due to the difference in the number of profitable days.

4.5 Summary

Based on our experimentations and results, we achieved our main goal to improve estimation of trend reversal. Greater insight into trend reversal prediction in DC can be achieved if we can correctly estimate OS event length. Our out-of-sample experimentation results show that our algorithm was able to estimate OS length better than the other two algorithms currently in use. To evaluate the performance of our algorithm at trading, we embedded it in an existing trading strategy. Overall, test results showed that our version of the trading strategy called Reg-GP+GA was the most profitable trading strategy statistically outperform on of the DC based strategies and the physical time-based trading strategies.

In summary, anticipating trend reversal correctly is crucial for record profitable trading results. Once a directional change is confirmed, we can estimate OS length, thus forecast trend reversal in DC. We used SRGP to create equations that improved prediction of OS event length. We showed that our SRGP can estimate OS length with improved accuracy in comparison to other known OS length estimation techniques experimented. We showed that profitability of an existing trading strategy was improved after replacing the OS event length estimation equation with the SRGP created one, thereby, outperforming two intrinsic-time and two physical-time trading strategies (technical analysis, buy-and-hold). The limitations of this contribution includes; (1) assumption that all DC trends are composed of both DC event and OS event which makes the our trend reversal forecasting algorithms prone to incorrect predictions when DC trend is composed of only DC event,(2) use of generalised threshold size to summarise DC events which may not be efficient in capturing the most significant events in physical time dataset, (3) limited number of currency pairs compared which limits the ability to generalise the findings, (4) limited analysis without consideration for the risk level of the trading strategies to achieve profitable returns and (5) example equations evolved by SRGP were not presented. These are all going to be addressed in the next chapter.

Chapter 5

Combining Classification and Symbolic Regression for Trend Reversal Prediction

In the previous chapter, we presented a symbolic regression GP-based algorithm that explores the types of relationship discoverable between DC event and OS event lengths. To the best of our knowledge, an assuption in the existing approaches for discovering a relationship between DC event and OS event lengths is that all DC events have a corresponding OS event. After testing with thresholds from 0.005% to 1.0% with step size of 0.0025, we observe that, in certain cases, it was possible to have as little as 14.77% of DC event having a corresponding OS event. For example, generating DC event series with a threshold of 0.0125% from a 10-minutes, EUR/CSK, 01/06/2016 - 30/06/2016 times series, resulted in DC event series where 48% of DC events had a corresponding overshoot and the maximum observed in our preliminary experimentation was 52.46%. Thus, one should not assume that all DC events are followed by an OS event or in other words an OS event of length greater than zero, since it can lead to incorrect trend reversal prediction, consequentially causing traders to misjudge the trend reversal point, consequently making incorrect trading decisions.

Another observation, which we made in the previous chapter is the technique used in selecting threshold to sample DC event series. Although it was reported by Kampouridis and Otero (2017) that thresholds are chosen by first performing tuning to select best performing threshold, tuning was done across their five currency pair datasets. According to Tsang et al. (2017), different thresholds observe distinct DC events and trends from the same physical time-series. We can thus hypothesise that the same threshold can observe different DC events and trends from different physical time-series. These observations have motivated us to propose two improvements to our current DC trend reversal approach.

We propose, (1) a dynamic threshold selection step that finds a threshold that best captures significant events in physical time dataset, and (2) a classification step to separate DC trends into two categories. In the first category are DC trends composed of DC event and OS event which we called αDC. In the second category are those DC trends composed of only DC events (also can be considered as DC trends with OS event of zero length) and we called them βDC. In βDCs, DC events are followed by another DC event of the opposite direction (e.g., a downwards DC trend is directly followed by an upwards DC trend). This knowledge will enable us use the OS event length estimation equation more efficiently, thus improving trend reversal point forecast accuracy.

Therefore, our goal in this chapter is twofold: (1) to improve our trend reversal estimation model by tailoring threshold selection, distinguishing DC trends composed of DC event and OS event from others without OS event and estimate OS event length only when a DC trend is classified to compose of OS event and DC event, (2) to embed our trend reversal estimation approach in a novel trading strategy, to show that improved trend reversal estimation leads to increase in excess return at reduced risk. The improvement proposed in our trend reversal estimation model is carried out using only training data to avoid bias when we apply our algorithm to the unseen data. The rest of the chapter is organized as follows: Section 5.1 presents our dynamic DC threshold selection and OS event length estimation approaches. Section 5.2 presents our novel DC trend classification approach. Section 5.3 presents our DC trend estimating framework that combines DC trend classification and OS event length estimation. Section 5.4 presents our novel trading algorithm. 5.5 presents our experimental setup. Section 5.6 presents our benchmark trading strategies, and Section 5.7 presents and discusses our findings. Finally, Section 5.8 concludes the chapter.

5.1 DC threshold and Symbolic Regression GP Selection

Selection of an appropriate threshold to snapshot significant events from a physical time series is vital. This is because each threshold provides a distinct perspective of the data. Figure 5.1 presents a physical time-series that is transformed into a DC event series using thresholds 0.01% and 0.018%. As we can see, each threshold produced a different event series. Each event series has a distinct number of trends which conditions the maximum possible profit of the event series it captures (Tsang et al. 2017). Therefore, the choice of an appropriate threshold size is crucial. In our first contribution in Chapter 4, we used five fixed size thresholds to create DC event

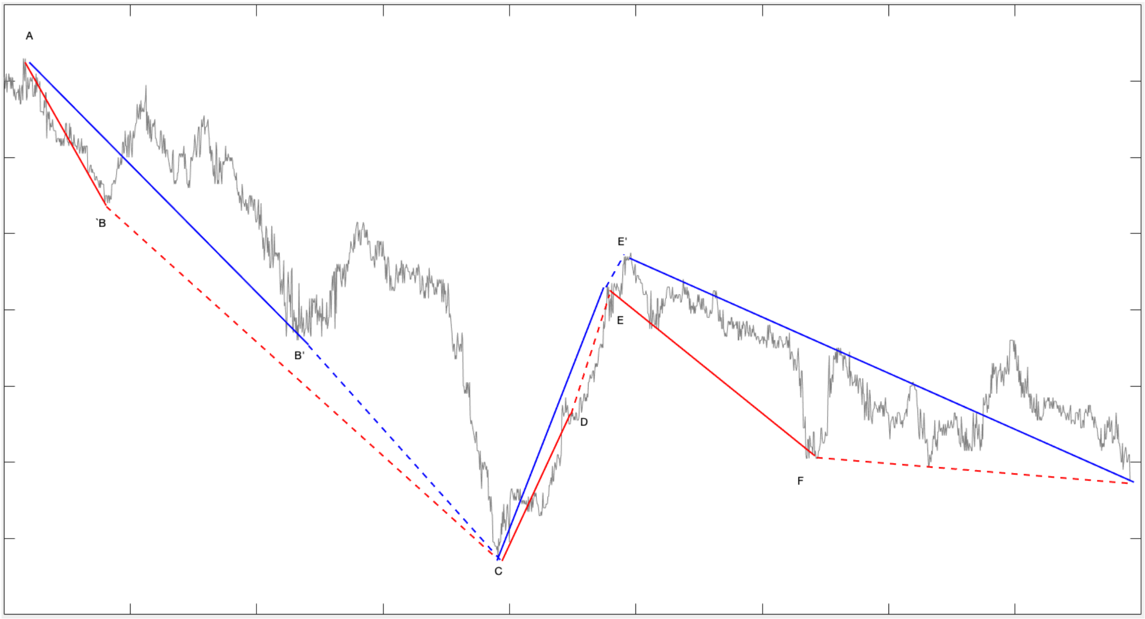


Figure 5.1: Directional changes for GBP_JPY currency pair. The red lines denote a set of events defined by a threshold $\theta = 0.01\%$, while the blue lines refer to events defined by a threshold $\theta = 0.018\%$. The solid lines indicate the DC events, and the dashed lines indicate the OS events. Under $\theta = 0.01\%$, the data is summarised as follows: Point $A \mapsto B$ (downturn directional change event), Point $B \mapsto C$ (downward overshoot event), Point $C \mapsto D$ (Upturn directional change event), Point $D \mapsto E$ (Upward overshoot event), Point $E \mapsto F$ (Downturn directional change event). Under $\theta = 0.018\%$, the data is summarised as follows: Point $A \mapsto B^{'}$ (Downturn directional change event), Point $B^{'} \mapsto C$ (Downward overshoot event), Point $C \mapsto E$ (Upturn directional change event), Point $E \mapsto E^{'}$ (Upward overshoot event). Points A, C, E, and $E^{'}$ are DCE points (DC Extreme). Points B, $B^{'}$, D, E, and F are called DCC points (DC Confirmation).

series of the five currency pairs that we experimented, an approach which may not be optimal. In this contribution, we selected the threshold that was associated with the highest ranked symbolic regression model.

Figure 5.2 presents how we simultaneously chose a threshold, DC event series and SRGP model (i.e., symbolic regression GP). First, we created a pool of candidate DC thresholds from 0.005% to 0.1% and a step size of 0.0025. We then used each threshold to generate an event series from the same physical time series. From each

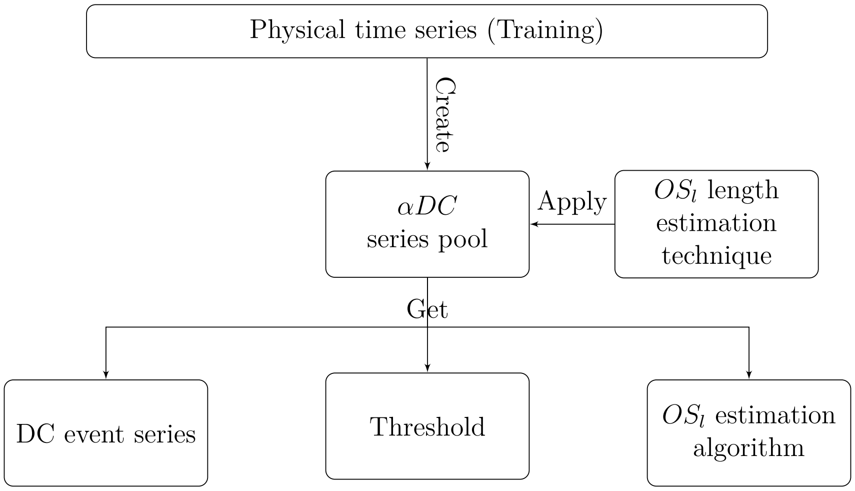


Figure 5.2: Our proposed framework for evolving symbolic regression model and selecting threshold and DC event of best fitting regression model.

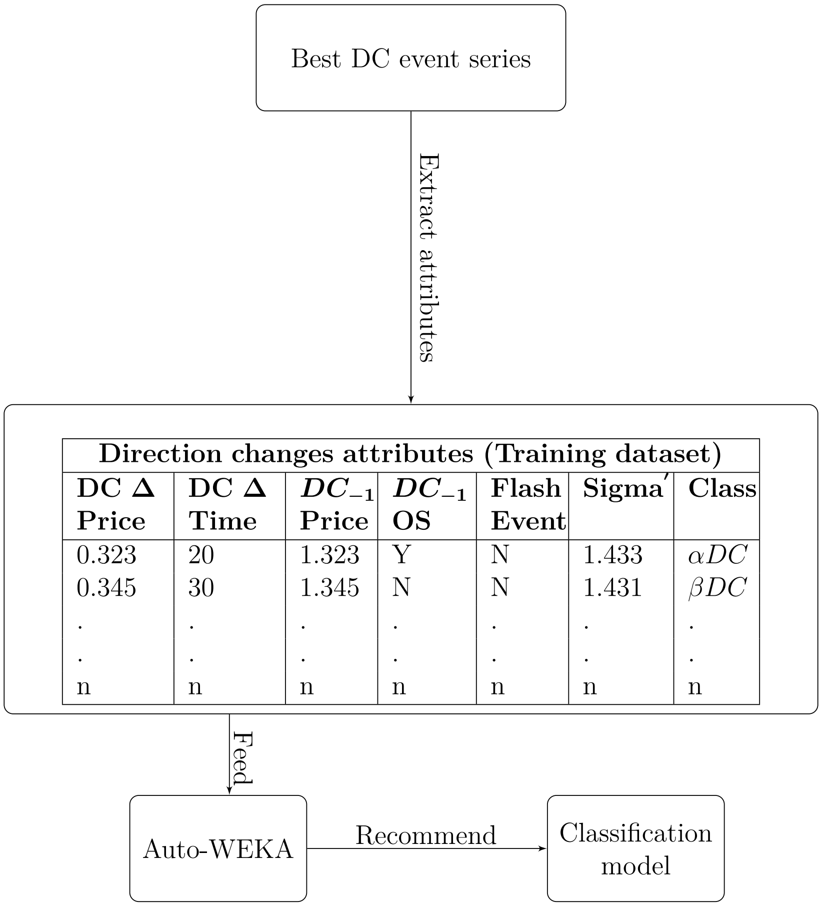
event series (i.e. the DC dataset), we selected trends that have OS event length greater than 0 to create a new event series i.e., αDCs,. We then evolved symbolic regression GP model for each new event series that estimated the average OS event length. We ranked the models according to their estimation accuracy and selected one that had the least root mean squared error (RMSE) together with the threshold and original DC event series associated with it for the rest of experimentation.

$$OS_l = f(DC_l) \tag{5.1}$$

where :

OS_l = the length of an OS event

DC_l = the length of a DC event



Direction changes attributes (Training dataset)						
DC Δ Price	**DC Δ Time**	**DC_{-1} Price**	**DC_{-1} OS**	**Flash Event**	**Sigma′**	**Class**
0.323	20	1.323	Y	N	1.433	αDC
0.345	30	1.345	N	N	1.431	βDC
.	.	.	.	.	.	.
.	.	.	.	.	.	.
n	n	n	n	n	n	n

Figure 5.3: Our proposed framework for creating a classification model. The classification model classifies DC trends into αDC and βDC

5.2 Classification Step

Figure 5.3 presents how we used Auto-WEKA, an AutoML framework we already described in Chapter 3.4.1 to find a suitable classifier for classifying DC trends as either αDC or βDC. As can be seen in the figure, we extracted relevant DC attributes from ‘Best DC event series’, that is, the DC events series that was selected in the previous step we described in Section 5.1. We then fed the attributes dataset into Auto-WEKA which outputs the recommended classification model.

Table 5.1: Classification dataset attributes - A brief description of independent variables used for classifying whether a DC trend has OS event or not.

Attributes	**Name**	**Formula**	**Description**
X1	DC˙eventPrice	$\left\lvert \frac{P_{ext} - P_{dcc}}{P_{ext}} \div \theta \right\rvert$	This is the price difference between the upturn/downturn extreme point and the confirmation point (Tsang et al. 2017).
X2	DC˙eventTime	$\left\lvert \frac{T_{ext} - T_{dcc}}{T_{ext}} \div \theta \right\rvert$	This is the time difference between the upturn/downturn point and the directional change confirmation point (Tsang et al. 2017).
X3	$Sigma'$	$\left\lvert \frac{DC_{event}Price \times \theta}{DC_{event}Time} \right\rvert$	This is the speed at which price change from upturn/downturn extreme point to the directional changes confirmation point.
X4	$DCevent_{-1}$ price	P_{dcc-1}	This is the market price at previous confirmation points.
X5	$DCevent_{-1}$ OS		This is a Boolean variable (Yes/No). Indicates whether the immediate previous DC trend has an OS event.
X6	Flash event		This is a Boolean variable (Yes/No). Indicates whether DC event start time and end time are equal

The attributes used for creating our classification model are all DC-related and presented in Table 5.1. We use six different attributes, which are related to DC and OS events' price and time, as well as the speed of price changing. Attributes X1 and X2, were first derived and presented in (Glattfelder, Dupuis and Olsen 2011). Attributes X3, X4, X5 and X6 are new attributes derived from experiments with a set of different attributes and identifying the ones with the best classification performance on the validation dataset. To illustrate how the attributes are derived from a DC event series of EUR_USD, we use the second DC trend presented in Figure 5.4. The price and time at the extreme point are 1.37 USD and around 4.5 × 10^4 respectively and the price and time at DCC point are 1.33 USD and 7.2 × 10^4. The value of attribute **X1**, **X2** and **X3** are 1.33, 20 and 0.002 respectively. The value of attribute **X4**, **X5** and **X6** are extracted directly from the dataset and are 1.34, Yes and No respectively.

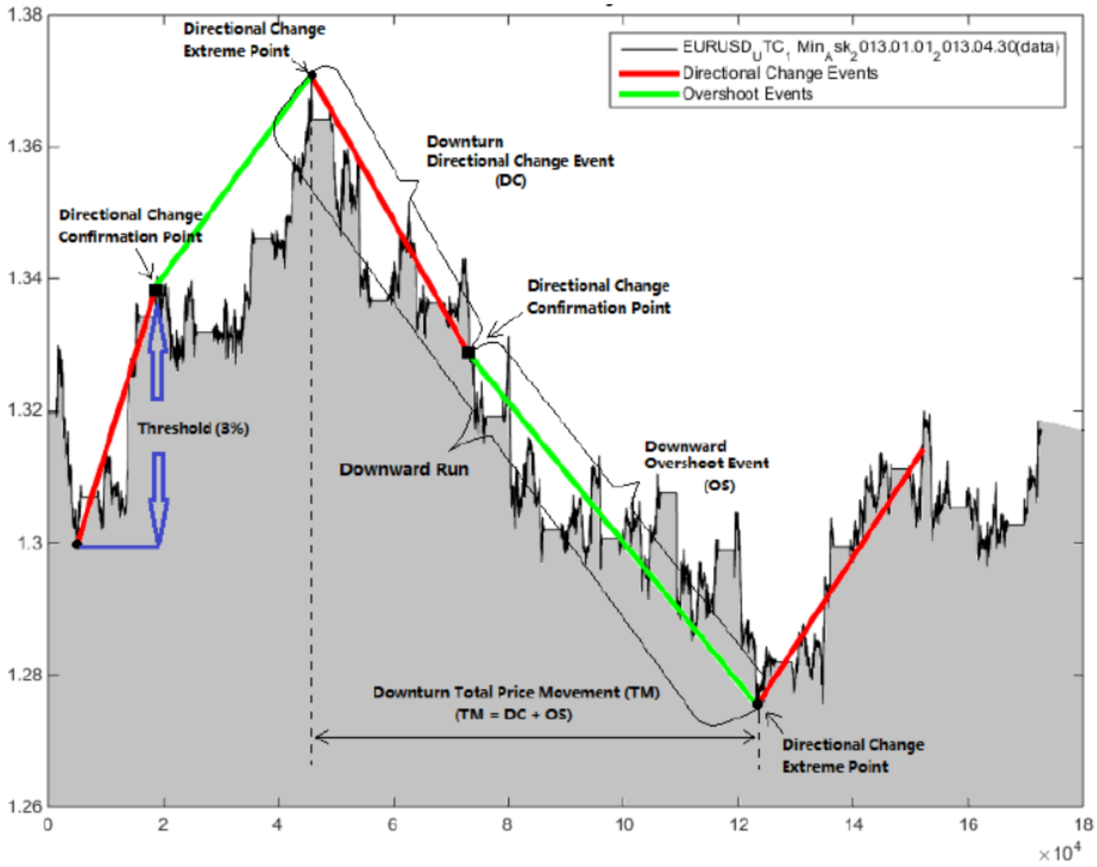


Figure 5.4: A set of directional changes trends (three DC events and two OS events) for EUR_USD currency pair captured from a minute physical time series using a 3% threshold. The red lines denote DC events and the green lines marks OS events. Source Tsang et al. (2017)

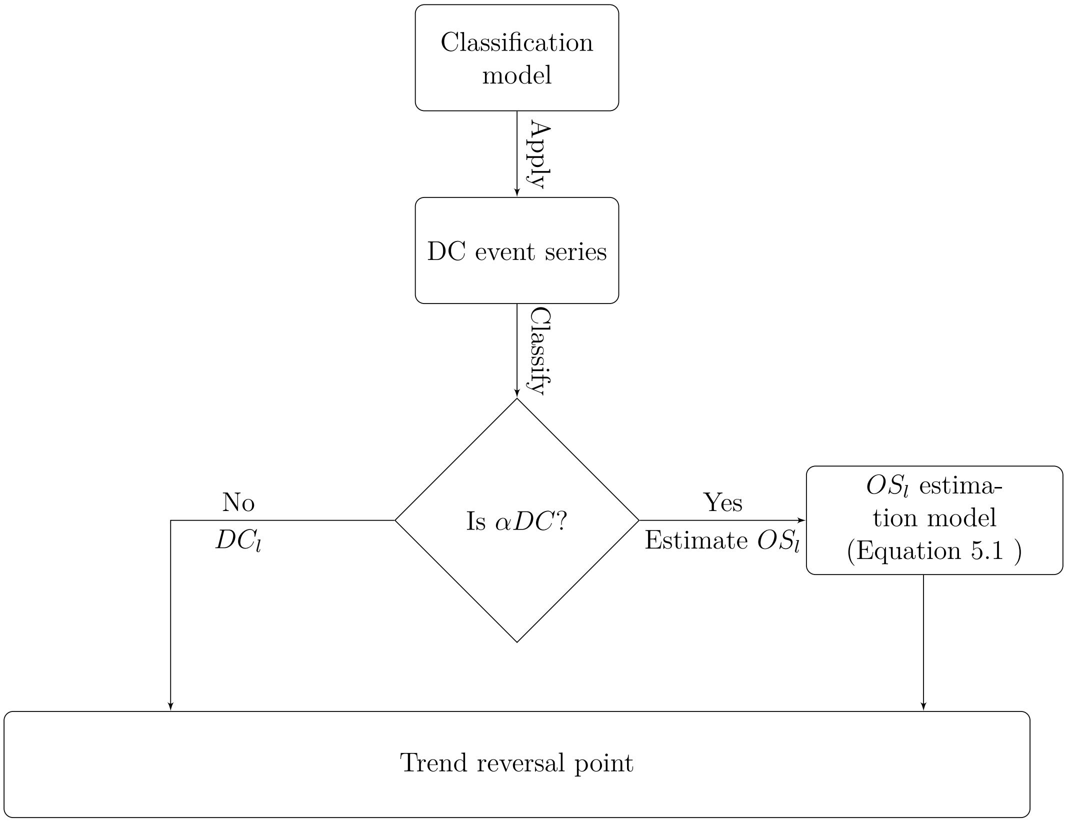


Figure 5.5: Our proposed framework to build a trend reversal point forecasting model. A DC trend classified to compose of only DC event is expected to reverse at DCC point, while DC trend classified to compose of DC and OS events is expected to reverse at estimated DCE point, which is the sum of DC event length at DCC point, and the OS event length predicted using SRGP derived Equation 5.1.

5.3 Trend Reversal Estimation Model

Figure 5.6 presents our framework for forecasting trend reversal point. Having chosen the best threshold, evolved a symbolic regression GP model and selected a classification model, we are now able to combine them to form our trend reversal point forecasting model. The model uses the classifier to determine whether a DC trend is an αDC or βDC. If the DC trend is classified as αDC, we conclude that, at the end of the DC event an OS event in the same direction follows. We use our symbolic regression GP model formulated according to Equation (5.1) to estimate the OS event length. The estimated trend reversal point is then calculated as the sum of the DC event length and the estimated OS event length. On the other hand, if the DC trend is classified as βDC, we conclude that the trend reverses at the directional change confirmation point.

5.4 Trading Strategy

To evaluate the effectiveness of our trend reversal forecasting model, we embed it in a novel trading strategy and Figure 5.4 presents the framework of the strategy. In the rest of the section, we present the trading strategy and how it is evaluated.

5.4.1 Rules Overview

A Forex market is a quotation of two currency pair, with the value of one currency being quoted against another. The first currency in the pair is called the base currency, and the second currency is called the quote currency. The market price indicates how much of the quoted currency is needed to purchase the base currency. For example, assuming the most recent quotation of EUR/USD market 1.2500. This means 1.2500 Dollars is required to purchase 1 Euro. Algorithms 5.1 and 5.2 present the rules used by our trading algorithm for selling and buying the base currency. We sell the base currency at the estimated trend reversal point in upward trends, provided there is not an existing open position and return is positive after deducting transaction costs. We buy the base currency at the estimated trend reversal point in downward trends if there is an existing open position and the return is positive after

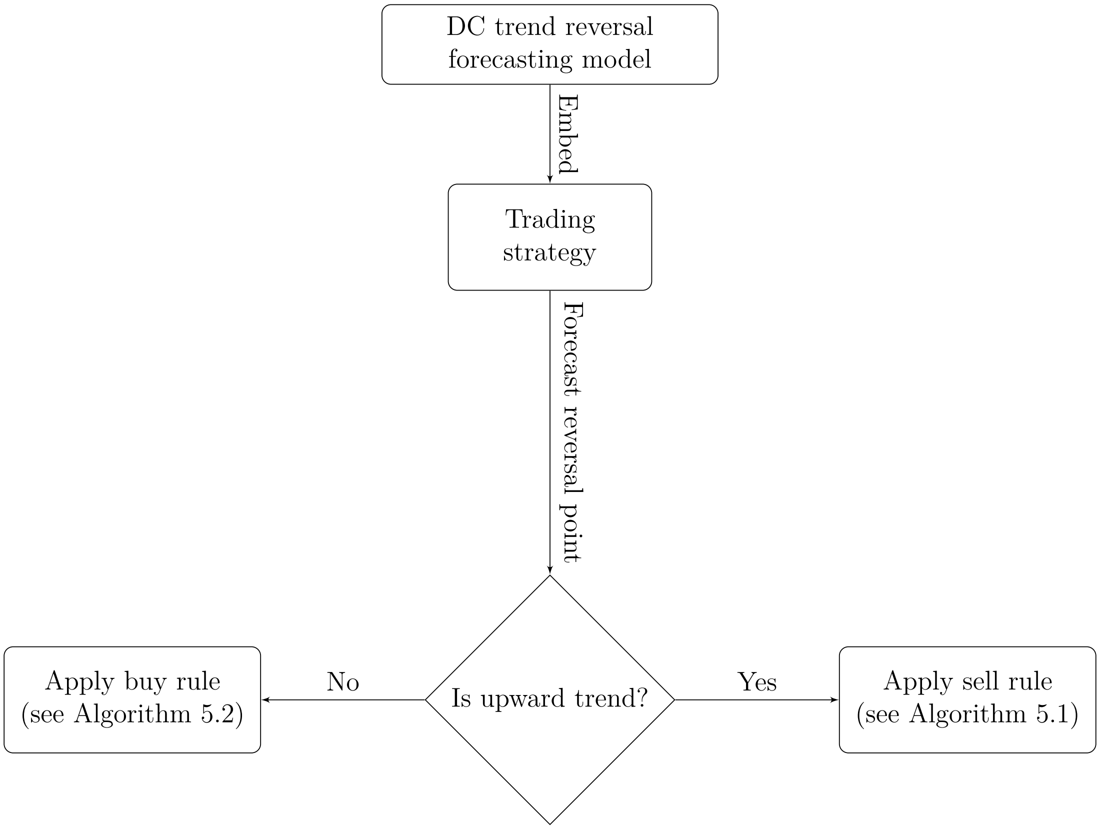

DC trend reversal forecasting model
Embed
Trading strategy
Forecast reversal point
Is upward trend?
No
Yes
Apply buy rule (see Algorithm 5.2)
Apply sell rule (see Algorithm 5.1)

Figure 5.6: Our proposed framework to build an single threshold-based trading strategy. It combines a DC trend reversal point forecasting model and trading rules. The forecasting model and trading rules are applied at every directional change confirmation point in the DC event series.

deducting transaction costs. It is worth noting that the trend reversal point can be either of a αDC or βDC trend. In all other cases, we adopt a hold trading strategy. For example, if a new DC trend is confirmed before the estimation trend reversal point is reached. All transactions are done using our entire capital. The transaction cost is 0.025% per transaction.

Algorithm 5.1 Trading rules used for selling the base currency

```
Require: Sell rule
  if DC trend is upward then
      if There is no open position then
          if Is $\beta DC$ && Return is not negative then Open a position at DCC point
          else if Is $\alpha DC$ && DC trend does not reverse before estimated DCE point && Return is not negative then Open a position at forecasted DCE point
          else Hold
          end if
      end if
  end if
```

Algorithm 5.2 Trading rules used for buying the base currency

```
Require: Buy rule
  if DC trend is downward then
      if There is an open position then
          if Is $\beta DC$ && Return is not negative then Close position at DCC point
          else if Is $\alpha DC$ && DC trend does not reverse before estimated DCE point && Return is not negative then Close position at forecasted DCE point
          else Hold
          end if
      end if
  end if
```

5.4.2 Trading strategy evaluation

To evaluate our trading strategy, we measure profitability and risk. We report average return, Sharpe ratio and Maximum Drawdown (MDD). Return, shown in Equation 5.2, measures the amount earned or the loss incurred from a Forex buy or sell transaction after considering transaction cost. Transaction cost, shown in Equation 5.3, is the expense incurred for completing a trade transaction and it is calculated per transaction as 0.025% of the Forex amount traded. We also measure return relative to the degree of risk involved in achieving it using Sharpe ratio, shown in Equation 5.5. Sharpe ratio is used by investors to determine whether the expected return outweighs the risk involved with the transaction. It is calculated by deducting risk-free rate from the mean return and dividing the result by the standard deviation.

Risk-free return is a theoretical return ascribed to an investment that provides a guaranteed return with zero risk which is factored into the pricing of an asset and excluded when evaluating the risk-adjusted return. To the best of our knowledge, there aren't such risk free investments in Forex trading and for that reason risk-free rate is assign a zero value. To evaluate the potential loss in value of the currency that we buy due to changes in market conditions, we measure MDD, shown in Equation 5.4. It is measured by calculating the maximum observed loss from a peak price to a trough before a new peak is reached.

$$R = (Q - TC) * FXrate \tag{5.2}$$

$$TC = Q * \frac{0.025}{100} \tag{5.3}$$

$$MDD = \frac{P_{trough} - P_{peak}}{P_{peak}} \tag{5.4}$$

$$SharpeRatio = \frac{R - RFR}{\sigma_R} \tag{5.5}$$

In Equation 5.2, R is the return, Q is the quantity, TC the transaction cost, $FXrate$ the FX rate of the relevant currency pair, MDD is the mean Maximum Drawdown, P_{trough} the trough of the price, P_{peak} the peak of the price, RFR the risk free rate, and σ_R is the standard deviation of the return.

5.5 Experimental setup

We use 10-minute interval high frequency data from March 2016 to February 2017 of the following currency pairs: AUD_JPY (Australian Dollar and Japanese Yen), AUD_NZD (Australian Dollar and New Zealand Dollar), AUD_USD (Australian Dollar and US Dollar), CAD_JPY (CAD Dollar and Japanese Yen), EUR_AUD (Euro and Australian Dollar), EUR_CAD (Euro and Canadian Dollar), EUR_CSK (Euro and Czechoslovak koruna), EUR_NOK (Euro and NOK), GBP_AUD (British Pound and Australian Dollar), NZD_USD (New Zealand Dollar and US Dollar), USD_CAD (US Dollar and Canadian Dollar), USD_NOK (US Dollar and Norwegian Krona), USD_JPY (US Dollar and Japanese Yen), USD_SGD (US Dollar and Singaporean Dollar), USD_ZAR (US Dollar and South African Rand), EUR_GBP (Euro and British Pound).

We also use 10-minute interval data from June 2013 to May 2014 of the following currency pairs: EUR_USD (Euro and US dollar), EUR_JPY (Euro and Japanese Yen), GBP_CHF (British Pound and Swiss Franc), and GBP_USD (British Pound and US Dollar). We consider each month in the period as a separate physical-time dataset. In our tuning phase, we use 200 DC datasets for tuning (i.e., five DC thresholds × 20 currency pairs × first two months of our physical-time data). For the rest of the experiment, we use 1000 DC datasets (i.e. five DC thresholds × 20 currency pairs × remaining 10 months of our physical time datasets). Each tuning and non-tuning DC dataset is split in 70:30 ratio as training and testing sets.

As different DC thresholds produce different DC event series, we chose to evaluate five different thresholds for all tuning and non-tuning DC datasets. These thresholds are the best five thresholds that are dynamically selected based on the RMSE of their associated OS event length estimation algorithm that we already presented in Section 5.1. In the results section, we report the average performance of each algorithm over these five DC thresholds.

5.5.1 Parameter tuning

With regards to the classification task, the only parameter of Auto-WEKA requiring tuning was the execution time. This is because Auto-WEKA requires time to search its algorithms and hyperparameter space for a classification model that best predict our two class labels (αDC, βDC). To avoid any bias and ensure that all data points are considered, Auto-WEKA trains the best fit classifier is determined using only training dataset and 10-fold cross-validation. We experimented with different runtime configurations, namely 15, 30, 45, 60 and 75 minutes. We chose a runtime of 60 minutes based on the mean F-measure, which we observed to diminish at a runtime of 75 minutes. Depending on the number of CPU cores available, it is possible to execute Auto-WEKA in serial or parallel mode. For our experiments we executed Auto-WEKA in serial mode, using a single CPU core..

With regards to the OS length estimation task, the only necessary tuning was for the GP algorithm (Equation 5.1). We tuned the GP population size, number of generations, tournament size, crossover probability and maximum depth parameters using the I/F-Race package (López-Ibánez et al. 2011). It implements a racing method for the selection of the best configuration for an optimisation algorithm by empirically selecting the most appropriate settings for the parameters of an optimisation problem (Birattari et al. 2010). Table 5.2 presents our GP parameter configuration determined by using I/F-Race run with a budget of 2000 iterations, a survivor rate of 4 and selecting the configuration with the largest population size.

Table 5.2: Regression GP experimental parameters for detecting DC-OS relationship, determined using I/F-Race.

Parameters	**Configuration**
Population	500
Generation	37
Tournament size	3
Crossover probability	0.98
Mutation probability	0.02
Maximum depth	3
Elitism	0.10

5.6 Trading

To evaluate the efficiency of our proposed trading framework, we compare it with several other benchmarks. These benchmarks are grouped into two categories: DC-related algorithms, and non-DC-related algorithms. In the rest of the section, we present in detail the different algorithms that we use as benchmarks.

5.6.1 DC-related algorithm

(C+) Factor-2 : This is an algorithm that is based on the DC approach originally presented in Glattfelder, Dupuis and Olsen (2011) where Factor-2, already described in Chapter 4, is used to estimate OS event length at directional change confirmation point. In this algorithm, we dynamically select the best threshold for sampling DC event series and replace the classification, and regression steps of our trend reversal estimation approach with Equation 5.6. Thus, Factor-2 is the approach without classification, and we estimate trend reversal for all DC trends in the dataset. DC trend reversal is calculated as the sum of DC and OS event length. OS event length is calculated as DC event length multiplied by 2 and calculated . *C+Factor-2* is the same approach with an additional classification step. In *C+Factor-2*, we estimate OS event length with Equation 5.6 only if a classifier predicts a trend to be αDC. If the DC trend is classified as βDC, the DCC point is treated as the trend reversal point.

$$OS_l \approx 2 \times DC_l \tag{5.6}$$

(C+) Factor-M : This is an algorithm that is based on the DC approach originally presented in (Kampouridis and Otero 2017) where Factor-M, already described in Chapter 4, is used to estimate OS event length at directional change confirmation point. In this algorithm, we dynamically select the best threshold for sampling DC event series and replace the classification, and regression steps of our trend reversal estimation approach with Equation 5.7. Thus, Factor-M is the approach without classification, and we estimate trend reversal for all DC trends in the dataset. DC trend reversal is calculated as the sum of DC and OS event length. OS event length is calculated as DC event length × M. *C+Factor-M* is the same approach with an additional classification step. In *C+Factor-M*, we estimate OS event length with Equation 5.7 only if the classifier predicts a trend to be αDC.

$$OS_l = M \times DC_l \tag{5.7}$$

where :

OS_l = the length of an OS event

DC_l = the length of a DC event.

C = a time-varying constant, which can take any positive real value, M > 0 .

(C+) Reg-GP : This is an algorithm that is based on the DC approach presented in our first contribution presented in Chapter 4 where Reg-GP is used to estimate OS event length at directional change confirmation point. In this approach, we dynamically select the best threshold for sampling DC event series and use Equation 5.1 for estimating OS event length. DC trend reversal point is calculated as the sum of DC and estimated OS event length. Reg-GP is the approach without classification, and we estimate OS event length at every DCC point encounter in the event series. *C+Reg-GP* is the same approach with an additional classification step which we already describe in Section 5.3. we estimate OS event length with Equation 5.1 only if the classifier predicts a trend to be αDC.

***p*-trading** : These are variations of the three afore mentioned approaches at estimating trend reversal (i.e., Factor-2, Factor-M and Reg-GP). A probability ratio p of DC event being followed by an OS event is calculated from the training DC event series associated with Factor-2, Factor-M, Reg-GP respectively and used to predict whether DC event will be followed by OS event in testing. For example, if there are

100 DC trends in an event series and 90 of them are composed of DC and OS events, p is calculated to be 0.9. At the directional change confirmation (DCC) point , we stochastically generating a real value between 0.0 and 1.0. If the value is less than or equal 0.9, we estimate OS event length of the DC trend. If the value is greater than 0.9, the DC trend is considered to reverse at the DCC point because there isn't an OS event (i.e. OS event length is 0). The motivation behind this scenario is to highlight the importance of the classification step before estimating OS event length. As there are three different event series, there are as a result three variations of the tailored trading benchmark namely, *p+Factor-2*, *p+Factor-M* and *p+Reg-GP*.

Trade at DCC point : Again, there are three different event series created and associated with Factor-2, Factor-M and Reg-GP. There are as a result three variation of this scenario namely *DCC+Factor-2*, *DCC+Factor-M* and *DCC+Reg-GP*. In this scenario, trade actions are taken as soon as a directional change is confirmed, i.e., at the DC confirmation point (DCC). The motivation behind this scenario is to investigate the trading profitability if the OS events length is ignored and instead, focus trading only on the DC events. Provided that our proposed classification algorithms outperform this scenario, it would again demonstrate that the introduction of the classification step is advantageous and better than not having classification and the knowledge of the OS length.

Non-DC benchmarks

Technical analysis trading strategy : Technical analysis based trading strategy is a well-known approach used in trading. It uses technical indicators for insight when to make trading decisions. We experimented with seven trading strategies that utilised the following technical indicators; Exponential Movement Average

(EMA), Bollinger Bands (BOLLIN), Simple Moving Average (SMA), AROON Oscillator (AROON), Rate of Change (ROC), Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD).

Buy-and-hold : Buy and hold is a well-known benchmark for trading algorithms. Under this trading strategy we buy the quoted currency in the first month of our non-tuning data and then sell it in exchange for the base currency after the 10 month period.

5.7 Results

Our experiment aims: (1) to demonstrate that the introduction of the classification step led to improvement in trend reversal estimation, and (2) to demonstrate that the dynamic selection of thresholds and the introduction of the classification step significantly improve the profitability of DC-based strategies in general, and specifically, our proposed trading strategy that embed our trend reversal estimation algorithm yielded more profit, outperforming other DC and non-DC based trading strategies such as technical analysis and buy-and-hold.

5.7.1 Regression result

Table 5.3 presents the average RMSE result of the OS length estimation step from the top five DC thresholds over 10 months for GP-Reg, Factor-M, and Factor-2. For each of these three algorithms, we present the RMSE for two variations: (1) with the classification step and (2) without the classification step. For the variations that included the classification step (denoted with prefix $C+$), trends in the event series are first classified as either αDC or βDC. Then, OS event length is estimated in the cases that trends are classified as αDC. Whereas in βDC, the trend is considered to reverse at DCC point and OS event length is not estimated.

Table 5.3 shows that C+Reg-GP has the lowest average RMSE (18.8175) across all six algorithms. C+Reg-GP has 11 cases (out of the 20 currency pairs) that returned the lowest RMSE per currency pair, Reg-GP has four such cases, C+Factor-M three cases, and C+Factor-2 two cases. More importantly, we observe that the average RMSE for each algorithm with the classification step has returned a lower average RMSE when compared to its respective variation without classification: C+Reg-GP (18.6550) vs Reg-GP (20.3216), C+Factor-M (20.5592) vs Factor-M (34.4474), and C+Factor-2 (21.3247) vs Factor-2 (25.7951). It is worth noting that the classification accuracy (presented in brackets) is high, ranging between 70% and 85%. As we have hypothesised, the introduction of classification step played an important role in reducing the average RMSE for all three algorithms (Reg-GP, Factor-M, and Factor-2).

An interesting observation is that while the EUR/CSK pair has relatively low classification accuracy across all variants (55-58%), howbeit, based on the average RMSE result recorded, the variants with the classification step is closer to the line of best fit. We investigated further and found that the EUR/CSK currency pair has the lowest average number of DC events (55 in the training set, 18 in the test set), while the average number of DC events for all other currency pairs is 194 in training and 60 in test. In addition, the length of DC events (i.e., number of physical time data points making up a single event) is the longest for EUR/CSK (46 in training, 32 in test), compared to an average of 12 in both training and test for all other currency pairs. This meant that there were fewer and long significant DC trends captured from the EUR/CSK physical time datasets. When the algorithms are run without the classification step, every DC event is assumed to be followed by an OS event and because the DC event lengths are long, the estimated OS event lengths are also long. Therefore, every time the trend reversal forecasting model that does not include the classification step (Reg-GP, Factor-M, Factor-2) estimated OS event where in reality the OS event length is zero, the RMSE recorded became larger. The classification step, despite the low accuracy (55-58%), reduces significantly the RMSE by reducing the number of times that the OS length estimation algorithm is used which in turn avoid larger trend reversal estimation errors.

Table 5.3: Mean RMSE values for each OS length estimator algorithm. 1000 datasets consisting of five different dynamically generated thresholds tailored to each DC dataset, 20 currency pairs, and 10 months of 10-minute interval data for each currency pair. In brackets is reported the classification accuracy, for reference (for C+Reg-GP, C+Factor-M, C+Factor-2). The best mean RMSE per currency pair is in bold text

Algorithms	C+Reg-GP	Reg-GP	C+Factor-M	Factor-M	C+Factor-2	Factor-2
AUD/JPY	**15.5670** (0.851)	15.6270	17.1570 (0.778)	25.5270	18.4720 (0.782)	22.2690
AUD/NZD	27.368 (0.805)	**24.3320**	27.4110 (0.806)	51.2420	31.9260 (0.761)	41.5920
AUD/USD	**11.5800** (0.829)	12.8140	12.7200 (0.768)	16.0950	11.8270 (0.745)	14.0600
CAD/JPY	**11.8430** (0.820)	18.7850	14.6860 (0.779)	39.9700	16.8800 (0.764)	27.2510
EUR/AUD	21.1710 (0.821)	20.5690	20.2010 (0.799)	25.7280	**14.7520** (0.751)	19.7490
EUR/CAD	**16.2050** (0.839)	17.7190	21.0950 (0.784)	23.1830	22.6420 (0.750)	24.8670
EUR/CSK	**41.9900** (0.557)	52.9490	46.0270 (0.581)	188.6080	63.0420 (0.565)	83.8450
EUR/GBP	24.1730 (0.825)	22.6350	25.5870 (0.766)	31.4300	**17.2120** (0.752)	18.7900
EUR/JPY	**19.9650** (0.821)	21.1170	23.4540 (0.758)	28.1620	23.1640 (0.748)	25.2040
EUR/NOK	**13.7170** (0.818)	13.7620	20.4120 (0.727)	27.2010	19.5710 (0.728)	22.4990
EUR/USD	28.2600 (0.806)	31.0610	**26.8990** (0.786)	38.5320	27.6690 (0.762)	30.0380
GBP/AUD	15.1380 (0.837)	**14.7190**	19.2820 (0.832)	21.6700	14.8810 (0.780)	17.9100
GBP/CHF	**15.9610** (0.831)	17.2040	17.5260 (0.784)	19.3580	21.4210 (0.769)	23.6690
GBP/USD	19.2040 (0.851)	24.8890	**17.8250** (0.790)	21.2230	25.3210 (0.746)	27.7780
NZD/USD	**10.2300** (0.848)	10.5880	11.0920 (0.772)	14.7310	13.1350 (0.773)	15.8960
USD/CAD	26.9340 (0.797)	**26.8180**	27.1330 (0.766)	34.6540	27.5190 (0.739)	29.3150
USD/JPY	**13.7040** (0.850)	14.5430	15.9860 (0.774)	17.9980	16.0310 (0.777)	18.3260
USD/NOK	07.7180 (0.887)	**7.3570**	9.96900 (0.813)	14.1280	8.1830 (0.792)	10.7640
USD/SGD	**26.9320** (0.780)	34.1480	31.9440 (0.799)	41.7120	27.4980 (0.720)	34.3600
USD/ZAR	5.4400 (0.877)	4.7960	**4.7770** (0.807)	7.7960	5.3470 (0.813)	7.7200
Average	**18.8175**(**0.818**)	20.5687	20.7382(0.773)	34.9169	21.4748 (0.749)	25.9807

Table 5.4: Statistical test results of OS length estimation according to the non-parametric Friedman test with the Hommel post-hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

Algorithm	Average Rank	$Adjust_{pHomm}$
C+Reg-GP (c)	1.9000	
Reg-GP	**2.5**	0.3105
C+Factor-M	**2.9499**	0.1518
C+Factor-2	**3.1500**	0.1038
Factor-2	**4.8999**	1.5835E-6
Factor-M	**5.5999**	1.9985E-9

Table 5.4 presents the result of Friedman's non-parametric statistical test. The null hypothesis is that all algorithms come from the same continuous distribution. The table shows the average rank according to the Friedman test (first column), and the adjusted p-value of the statistical test when that algorithm's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to the Hommel post-hoc test (second column). As we can observe, C+Reg-GP ranks first and statistically outperforms at the $\alpha = 0.05$ level all other algorithms. More importantly, C+Reg-GP outranks Reg-GP, C+Factor-2 outranks Factor-2, and C+Factor-M outranks Factor-M.

Summarising our findings so far, the addition of the classification step (C+Reg-GP, C+Factor-M, C+Factor-2) to existing DC-based algorithms (Reg-GP, Factor-M, Factor-2) that use Equations 4.1, 5.7 and 5.1 has reduced the average predictive error. Additionally, the DC algorithms that use the classification step outrank their variation that estimate OS event length at every DC event.

5.7.2 Trading result

Our interest now shifts to the trading step in order to investigate whether the introduction of classification step also leads to an increase in trading profit margins (in addition to reduced OS length estimation error, as we have just seen in the previous section).

We compare the average returns, maximum drawdown and Sharpe ratio results of our approach to other strategies. We group the results according to data sampling technique namely, directional changes, technical analysis and buy-and-hold. We further breakdown the presentation of directional changes based result according to their trend reversal prediction algorithm. We would like to draw the attention to cases where 0.00 is reported as return. This indicates that a hold action was taken by the trading strategy in the 10 months period we experimented. Best value for each row (currency pair) is shown in boldface. In all tables, the best value among the different variants (Reg-GP, Factor-M, Factor-2, technical indicator) is underlined. We also present their Friedman non-parametric statistical test result. The null hypothesis in all cases is that the algorithms come from the same continuous distribution. The first column on the table for the Friedman test result presents the average rank of each algorithm and the second column presents the adjusted p-value of other algorithm's average rank compared to that of the control algorithm (i.e., algorithm with the best rank). The adjusted p-value is calculated by the Hommel post-hoc test.

Comparison to DC based trading strategies

Table 5.5 presents comparison monthly mean return result of C+Reg-GP and other strategies that estimated OS event length using symbolic regression GP, namely

Reg-GP, p+Reg-GP and DCC+Reg-GP. The results shows that C+Reg-GP has the highest return and outperformed them in 14 of the 20 currency pairs compared.

Table 5.5: Average monthly return (%) result. Comparison between C+Reg-GP and other GP based trading strategies. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. five DC dataset were generated using five dynamically generated thresholds tailored to each DC dataset. Best value for each row (currency pair) is shown in boldface. Result shown in % values.

Dataset	C+Reg-GP	Reg-GP	p+Reg-GP	DCC+Reg-GP
AUD_JPY	0.0000	0.0000	0.0000	0.0000
AUD_NZD	**0.2600**	-0.0890	0.0709	0.0110
AUD_USD	**0.2727**	-0.4636	-0.2061	-0.2223
CAD_JPY	0.0000	0.0000	0.0000	0.0000
EUR_AUD	**0.1861**	-0.0391	-0.0868	-0.1626
EUR_CAD	**0.1922**	-0.2428	-0.2218	-0.1332
EUR_CSK	0.0336	0.0102	0.0191	**0.0455**
EUR_GBP	**0.1040**	-0.0865	-0.0350	0.0218
EUR_JPY	**0.0202**	-0.0623	-0.0036	-0.0486
EUR_NOK	**0.3509**	-0.0428	-0.1281	0.0048
EUR_USD	-0.0006	**0.0202**	-0.0688	-0.2548
GBP_AUD	**0.3542**	0.2956	-0.1312	-0.0526
GBP_CHF	**0.2022**	-0.1160	-0.0536	-0.1384
GBP_USD	-0.0590	**-0.0478**	-0.1415	-0.4172
NZD_USD	**0.2803**	-0.4779	-0.0115	0.0738
USD_CAD	**0.0443**	0.0109	-0.3405	-0.3064
USD_JPY	0.0000	0.0000	0.0000	0.0000
USD_NOK	**0.4612**	-0.0208	-0.2210	-0.0662
USD_SGD	**0.0303**	0.0272	-0.0516	-0.1478
USD_ZAR	**1.7625**	0.8403	-0.0913	0.6432
Average	**0.2247**	-0.0242	-0.0851	-0.0575

Table 5.6 presents comparison result of return between C+Reg-GP and C+Factor-M, Factor-M, p+Factor-M and DCC+Factor-M. Result shows that C+Reg-GP outperformed all Factor-M variants in 14 currency pairs. C+Factor-M outperformed other Factor-M variants in 13 currency pairs.

Table 5.6: Average monthly return (%) result. Comparison between C+Reg-GP and trading strategies based on Factor-M. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. five DC dataset were generated using five dynamically generated thresholds tailored to each DC dataset. Best value for each row (currency pair) is shown in boldface. Result shown in % values

Dataset	C+Reg-GP	C+Factor-M	Factor-M	p+Factor-M	DCC+Factor-M
AUD_JPY	0.0000	0.0000	0.0000	0.0000	0.0000
AUD_NZD	**0.2600**	0.0747	-0.0626	0.1084	0.0223
AUD_USD	**0.2727**	-0.0037	-0.3270	-0.1749	-0.2933
CAD_JPY	0.0000	0.0000	0.0000	0.0000	0.0000
EUR_AUD	**0.1861**	0.0139	-0.1244	-0.0974	-0.0197
EUR_CAD	**0.1922**	0.0784	-0.0194	0.0621	-0.2208
EUR_CSK	0.0336	0.0381	0.0355	0.0264	**0.0643**
EUR_GBP	**0.1040**	0.0682	-0.0609	0.0625	-0.1136
EUR_JPY	0.0202	0.0112	-0.0197	**0.0218**	0.0007
EUR_NOK	**0.3509**	0.1703	-0.1475	-0.0895	0.0955
EUR_USD	**-0.0006**	-0.0894	-0.1049	-0.1939	-0.1396
GBP_AUD	**0.3542**	0.1012	-0.2473	0.0719	0.0035
GBP_CHF	**0.2022**	-0.0209	-0.0866	-0.1372	-0.1590
GBP_USD	**-0.0590**	-0.1226	-0.2035	-0.2336	-0.3485
NZD_USD	**0.2803**	-0.1234	-0.1586	-0.0155	-0.0886
USD_CAD	**0.0443**	-0.0293	-0.2238	-0.2230	-0.2475
USD_JPY	0.0000	0.0000	0.0000	0.0000	0.0000
USD_NOK	**0.4612**	0.1419	-0.4332	-0.1482	0.0011
USD_SGD	0.0303	**0.1108**	-0.0233	0.0229	-0.0028
USD_ZAR	**1.7625**	0.9516	0.6954	0.3467	0.6417
Average	**0.2247**	0.0686	-0.0756	-0.0295	-0.0402

Table 5.7 presents comparison result between C+Reg-GP and C+Factor-2, Factor-2, p+Factor-2 and DCC+Factor-2. Return shows that C+Reg-GP outperformed all Factor-2 variants in 12 currency pairs. C+Factor-2 outperformed other Factor-2 variants in 10 currency pairs.

C+Reg-GP has the highest mean return (0.2247%) across all algorithms and an annualised return of 2.73% . Furthermore, all versions that have introduced the classification step recorded the highest average return in their respective group

Table 5.7: Average monthly return (%) result. Comparison between C+Reg-GP and trading strategies based on Factor-2 . 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. five DC dataset were generated using f dynamically generated thresholds tailored to each DC dataset. Best value for each row (currency pair) is shown in boldface. Result shown in % values.

Dataset	C+Reg-GP	C+Factor-2	Factor-2	p+Factor-2	DCC+Factor-2
AUD_JPY	0.0000	0.0000	0.0000	0.0000	0.0000
AUD_NZD	**0.2600**	0.0328	-0.0122	0.0616	0.0248
AUD_USD	**0.2727**	-0.0223	-0.1321	-0.2752	-0.4222
CAD_JPY	0.0000	0.0000	0.0000	0.0000	0.0000
EUR_AUD	**0.1861**	0.1434	0.0547	0.0926	-0.1400
EUR_CAD	**0.1922**	0.0810	-0.1512	-0.1225	-0.0871
EUR_CSK	0.0336	0.0139	0.0046	0.0146	**0.0548**
EUR_GBP	0.1040	**0.1105**	0.0317	0.0792	-0.0595
EUR_JPY	**0.0202**	-0.0287	-0.0156	0.0161	-0.0145
EUR_NOK	**0.3509**	-0.0300	-0.0691	0.1651	0.0693
EUR_USD	-0.0006	-0.0779	**0.0318**	-0.0416	-0.1367
GBP_AUD	**0.3542**	0.0990	0.4061	-0.2356	0.0387
GBP_CHF	**0.2022**	0.0514	-0.0283	-0.1698	-0.2131
GBP_USD	-0.0590	**-0.0050**	-0.0466	-0.1524	-0.3188
NZD_USD	**0.2803**	0.1294	-0.1738	-0.1421	-0.2045
USD_CAD	**0.0443**	-0.0465	0.0224	-0.3631	-0.1372
USD_JPY	0.0000	0.0000	0.0000	0.0000	0.0000
USD_NOK	**0.4612**	0.3929	-0.0188	-0.1846	0.0995
USD_SGD	0.0303	0.0712	**0.1299**	-0.0222	0.0348
USD_ZAR	**1.7625**	1.4571	0.4870	0.7265	0.8492
Average	**0.2247**	0.1186	0.0260	-0.0277	-0.0281

(C+Reg-GP, : 0.2247, C+Factor-M: 0.0684, C+Factor-2: 0.1186), whereas all other DC based strategies had negative mean returns, with the exception of Factor-2, which shows marginally positive average returns of 0.0260.

To support our findings, we applied Friedman's non-parametric statistical test. The result of the statistical test presented in Table 5.8 shows that all three DC versions with the classification step (i.e., C+Reg-GP, C+Factor-M, C+Factor-2) rank the highest, and outperform all other variants without the classification step. In addition, C+Reg-GP ranks first and statistically outperforms all algorithms, apart from C+Factor-M and C+Factor-2, at the 5% significance level.

Table 5.8: Statistical test results of average montly returns according to the non-parametric Friedman test with the Hommel post-hoc test of C+Reg-GP (c) vs DC based trading strategies. 10-minute interval out-of-sample date. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
C+Reg-GP (c)	2.6750	-
C+Factor-M	4.2250	0.1740
C+Factor-2	4.4250	0.1740
Factor-2	**5.9250**	0.0131
p+Factor-M	**6.7750**	0.0013
p+Factor-2	**6.9250**	8.0806E-4
DCC+Factor-2	**7.2250**	3.9541E-4
DCC+Factor-M	**7.5250**	1.4717E-4
Reg-GP	**7.6250**	9.9088E-5
p+Reg-GP	**7.8250**	5.6490E-5
DCC+Reg-GP	**8.2750**	9.0371E-6
Factor-M	**8.5750**	2.5118E-6

Even though C+Reg-GP recorded higher return than other DC based trading strategies, it is important to measure the risk taken to achieve it. For this reason, we also present results of our risk measures, namely MDD (maximum draw down) and Sharpe ratio. We did not record risk measures for currency pair AUD_JPY, CAD_JPY and USD_JPY, as no trading took place in these markets. Table 5.9 shows that C+Reg-GP had the lowest average MDD amongst Reg-GP based strategies recording a total of 0.1259 and outperform them in 13 currency pairs.

Table 5.10 presents comparison between the mean MDD result of C+Reg-GP and Factor-M based strategies. C+Reg-GP recorded the lowest average MDD and outperform these strategies in 9 currency pairs.

Table 5.9: Average maximum drawdown (%) result for Reg-GP based trading strategies. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. five DC dataset were generated using five dynamically generated thresholds tailored to each DC dataset. Best (lowest) value for each row (currency pair) is shown in boldface. Result shown in % values.

Dataset	C+Reg-GP	Reg-GP	p+Reg-GP	DCC+Reg-GP
AUD_NZD	**0.1230**	0.1506	0.1713	0.3185
AUD_USD	**0.1595**	0.3123	0.5710	0.6917
EUR_AUD	**0.1058**	0.1545	0.4086	0.6214
EUR_CAD	**0.1353**	0.2577	0.4772	0.4494
EUR_CSK	**0.0057**	0.0080	0.0218	0.0253
EUR_GBP	0.1005	**0.0778**	0.1460	0.2789
EUR_JPY	**0.0106**	0.0383	0.0112	0.0255
EUR_NOK	**0.1331**	0.1476	0.2844	0.3871
EUR_USD	0.1555	**0.0688**	0.2059	0.4034
GBP_AUD	**0.1912**	0.2391	0.6403	0.6260
GBP_CHF	**0.0956**	0.1064	0.1897	0.3278
GBP_USD	**0.1323**	0.1797	0.2095	0.3867
NZD_USD	**0.2892**	0.3242	0.4777	0.6830
USD_CAD	0.1678	**0.1615**	0.5727	0.5389
USD_NOK	**0.1406**	0.1747	0.7049	0.7367
USD_SGD	0.0770	**0.0741**	0.1891	0.3058
USD_ZAR	**0.1168**	0.1453	1.2417	1.2160
Average	**0.1259**	0.1542	0.3837	0.4719

Table 5.11 presents comparison between the mean MDD result C+Reg-GP and Factor-2 strategies and it shows that C+Reg-GP has the lowest MDD, outperforming them in 12 currency pairs. Surprisingly, our result shows Factor-M and Factor-2 are the best strategies in their categories, they outperform the version with the additional classification step in 15 currency pairs each.

Comparison between the MDD results of Factor-M and Factor-2 and their respective returns results that is presented in Tables 5.6 and 5.7, appear to indicate that there's a trade-off between higher return and risk. The MDD result of classification variants ranked second in both categories.

Table 5.10: Average maximum drawdown (%) result for Factor-M based trading strategies. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. five DC dataset were generated using five dynamically generated thresholds tailored to each DC dataset. Best (lowest) value for each row (currency pair) is shown in bold face. Result shown in % values

Dataset	C+Reg-GP	C+Factor-M	Factor-M	p+Factor-M	DCC+Factor-M
AUD_NZD	**0.1230**	0.2699	0.1588	0.2896	0.6500
AUD_USD	0.1595	0.5750	**0.1484**	0.6657	0.7631
EUR_AUD	0.1058	0.3006	**0.0768**	0.4332	0.9292
EUR_CAD	**0.1353**	0.2334	0.2033	0.3325	0.7941
EUR_CSK	**0.0057**	0.0133	0.0128	0.0189	0.1117
EUR_GBP	**0.1005**	0.1891	0.1375	0.2513	0.4845
EUR_JPY	0.0106	0.0091	**0.0084**	0.0157	0.0509
EUR_NOK	**0.1331**	0.2699	0.2560	0.4492	0.4974
EUR_USD	0.1555	0.2383	**0.1262**	0.3368	0.8276
GBP_AUD	**0.1912**	0.4910	0.2009	0.6226	0.8899
GBP_CHF	0.0956	0.3558	**0.0493**	0.4552	0.6282
GBP_USD	0.1323	0.3054	**0.1222**	0.3887	0.9857
NZD_USD	0.2892	0.5831	**0.2664**	0.6115	0.9989
USD_CAD	**0.1678**	0.2835	0.3403	0.6001	0.6407
USD_NOK	**0.1406**	0.4890	0.5694	0.6361	0.5926
USD_SGD	0.0770	0.1128	**0.0689**	0.1463	0.7689
USD_ZAR	**0.1168**	0.8950	0.2811	1.1893	0.8155
Average MDD	**0.1259**	0.3302	0.1780	0.4378	0.6723

To evaluate statistical significance of our MDD finding, we perform the Friedman statistical test, presented in presented in Table 5.12. As we can observe, the best ranking algorithm is C+reg-GP and it statistically outranked 8 of the strategies compared at the 5% significance level.

Figure 5.7 presents the total number of positive 5-month average Sharpe ratio. Excluding 6 periods where no trading took place, there are 34 risk-adjusted return summaries in total. Out of the 34, C+Reg-GP had positive Sharpe ratio in 28 which was the highest recorded amongst trading strategies compared. Of the 28 positive Sharpe ratio results, 6 where above 0.5, 18 were above 0.2 and less than 0.5. The rest were below 0.2.[1] Friedman test presented in Table 5.13, confirms our findings.

[1] A ratio of 0.2-0.3 is in line with the general market. A value of 0.5 is considered a market-beating performance if achieved over a long period, a ratio of 1 or better considered superb and

Table 5.11: Average maximum drawdown (%) result for Factor-2 based trading strategies compared. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. five DC dataset were generated using five dynamically generated thresholds tailored to each DC dataset. Best (lowest) value for each row (currency pair) is shown in bold face. Result shown in % values.

Dataset	C+Reg-GP	C+Factor-2	Factor-2	p+Factor-2	DCC+Factor-2
AUD_NZD	**0.1230**	0.3270	0.2059	0.3368	0.4763
AUD_USD	**0.1595**	0.4846	0.2754	0.7167	0.8327
EUR_AUD	**0.1058**	0.2599	0.2062	0.3246	0.5396
EUR_CAD	**0.1353**	0.2191	0.3079	0.3928	0.3773
EUR_CSK	0.0057	0.0171	**0.0024**	0.0254	0.0224
EUR_GBP	0.1005	0.2120	**0.0864**	0.2000	0.3884
EUR_JPY	**0.0106**	0.0372	0.0519	0.0100	0.0210
EUR_NOK	**0.1331**	0.4232	0.1516	0.2737	0.4209
EUR_USD	0.1555	0.2400	**0.0828**	0.2137	0.3925
GBP_AUD	0.1912	0.5337	**0.1773**	0.7777	0.9015
GBP_CHF	**0.0956**	0.2836	0.1908	0.3821	0.5696
GBP_USD	**0.1323**	0.2815	0.1617	0.3774	0.5990
NZD_USD	**0.2892**	0.4226	0.4036	0.7056	0.7510
USD_CAD	**0.1678**	0.3781	0.2393	0.5912	0.5521
USD_NOK	**0.1406**	0.3772	0.1716	0.7149	0.6544
USD_SGD	0.0770	0.1332	**0.0358**	0.2067	0.2644
USD_ZAR	**0.1168**	0.6761	0.3312	0.9046	1.0415
Average MDD	**0.1259**	0.3121	0.1813	0.4208	0.5179

C+Reg-GP ranks first and statistically outperforms all other trading strategies at the 5% level. In addition, C+Factor-M and C+Factor-2 rank second and third, respectively, which again demonstrates that the introduction of the classification step is beneficial to the DC algorithms.

Our final assessment measure of DC based strategies is Sharpe ratio presented in Figure 5.8. Results in cases where a hold strategy is used by the trading algorithms throughout the 10 months test periods are excluded. The x-axis presents the time period covered for the relevant currency pair, and the y-axis presents the average risk-adjusted return in percentages. C+Reg-GP consistently records positive adjusted returns, whereas the other strategies have a mix of both positive and negative returns

difficult to achieve over long periods and a negative Sharpe ratio indicates negative returns.

Table 5.12: Statistical test results of average maximum drawdown according to the non-parametric Friedman test with the Hommel post-hoc test of C+Reg-GP (c) vs DC based trading strategies. 10-minute interval out-of-sample date. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the α = 5% level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
C+Reg-GP (c)	2.1176	-
Factor-M	2.6471	0.6686
Reg-GP	3.1765	0.6686
Factor-2	3.5882	0.5879
C+Factor-M	**5.6471**	0.0173
C+Factor-2	**6.2941**	0.0037
p+Reg-GP	**7.1765**	2.5815E-4
p+Factor-2	**8.4706**	1.9537E-6
p+Factor-M	**8.4706**	1.9537E-6
DCC+Reg-GP	**9.3529**	4.4105E-8
DCC+Factor-2	**9.8834**	3.4164E-9
DCC+Factor-M	**11.1765**	2.6272E-12

over time.

Table 5.13: Statistical test results of average Sharpe ratio according to the non-parametric Friedman test with the Hommel post-hoc test of C+Reg-GP (c) vs DC based trading strategies. 10-minute interval out-of-sample date. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
C+Reg-GP (c)	2.6714	-
C+Factor-2	4.7571	0.0155
C+Factor-M	4.8714	0.0155
p+Factor-M	**5.8714**	6.1502E-4
Factor-2	**6.0714**	3.1945E-4
p+Reg-GP	**6.5286**	3.8169E-5
p+Factor-2	**6.6143**	2.3850E-5
Reg-GP	**7.5857**	8.3011E-8
DCC+Factor-M	**7.7000**	4.3202E-8
DCC+Factor-2	**7.9000**	1.1773E-8
DCC+Reg-GP	**8.4143**	2.6814E-10
Factor-M	**9.0143**	2.0347E-12

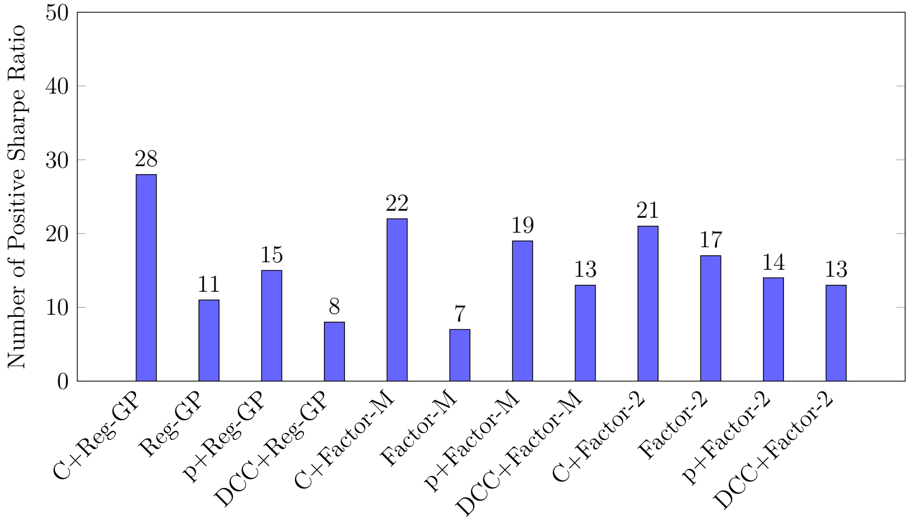
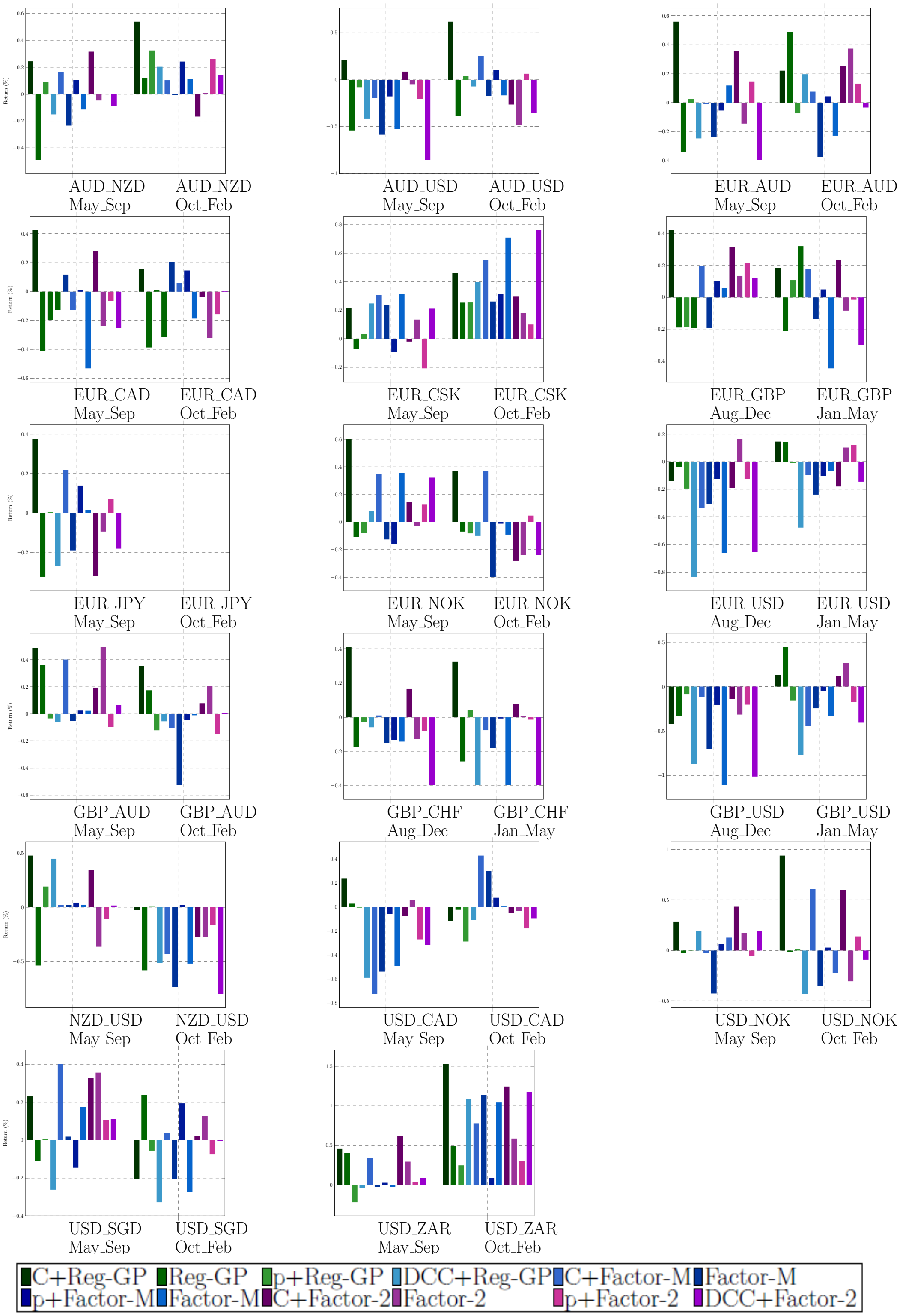


Figure 5.7: A comparison of total number of positive Sharpe Ratio between C+Reg-GP and other DC based trading approaches. 10-minute interval out-of-sample data. 20 currency pairs and 10 calendar months. Total of 40 Sharpe ratio results from five month average

Figure 5.8: Average Sharpe ratio for all currency pairs. C+Reg-GP versus other directional changes based trading strategies



Comparison to technical indicator based trading strategies

Our attention now shifts to comparison between C+Reg-GP and technical indicator based strategies namely, EMA, BOLLIN, SMA, AROON, ROC, RSI, MACD. Table 5.14 presents their average returns. We observe that C+reg-GP records higher return in 16 of the currency pairs and achieves the highest return amongst the strategies. All technical indicator based strategy recorded negative mean return. From this result, there isn't a clear second best strategy due to the fact that SMA record the second best average return overall and is second best in 4 currency pairs meanwhile AROON and RSI are second best in five currency pairs respectively.

Table 5.14: Average Technical Indicator return result for trading strategies compared. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. five DC dataset were generated using five dynamically generated thresholds tailored to each DC dataset. Best result per currency pair presented in boldface. BOLLIN is Bollinger bandwidth indicator

Dataset	**C+Reg-GP**	**EMA**	**BOLLIN**	**SMA**	**AROON**	**ROC**	**RSI**	**MACD**
AUD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
AUD/NZD	**0.260**	0.002	-0.007	-0.026	-0.002	-0.447	0.056	0.005
AUD/USD	**0.273**	-0.145	-0.393	-0.069	-0.025	-0.321	0.046	-0.147
CAD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
EUR/AUD	**0.186**	0.057	-0.365	-0.127	0.002	-0.166	-0.06	-0.092
EUR/CAD	**0.192**	-0.226	-0.759	-0.032	-0.068	-0.486	-0.013	-0.346
EUR/CSK	**0.034**	-0.233	-0.067	-0.164	0.000	-0.781	-0.138	-0.281
EUR/GBP	**0.104**	-0.135	-0.067	-0.048	-0.061	-0.367	-0.028	-0.240
EUR/JPY	**0.020**	0.015	0	-0.027	0	0	-0.022	0.013
EUR/NOK	**0.351**	-0.118	-0.232	0.149	0.003	-0.261	-0.043	-0.233
EUR/USD	**-0.001**	-0.492	-0.366	-0.25	-0.064	-0.262	-0.106	-0.409
GBP/AUD	**0.354**	-0.302	-0.201	-0.022	-0.061	-0.531	-0.159	-0.061
GBP/CHF	**0.202**	-0.268	-0.356	0.009	-0.087	-0.653	0.035	-0.331
GBP/USD	-0.059	-0.076	-0.61	-0.111	-0.045	-0.337	**0.008**	-0.361
NZD/USD	**0.280**	-0.234	-0.445	-0.151	-0.025	-0.333	0.124	-0.366
USD/CAD	**0.044**	-0.306	-0.64	-0.458	-0.016	-0.708	-0.299	-0.571
USD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
USD/NOK	**0.461**	-0.075	-0.592	0.069	-0.048	-0.7	-0.144	-0.154
USD/SGD	**0.03**	-0.044	-0.122	-0.178	-0.015	-0.513	-0.057	-0.295
USD/ZAR	**1.762**	0.344	-0.573	-0.356	0.004	-0.057	0.044	0.110
Average Return	**0.225**	-0.112	-0.29	-0.09	-0.025	-0.346	-0.038	-0.188

Friedman statistical test presented in Table 5.15 shows that the average return by C+Reg-GP is statistically significant in comparison to technical indicator based strategies and AROON, the best technical indicator based strategy in our experiment ranked second.

Table 5.15: Statistical test results of average returns according to the non-parametric Friedman test with the Hommel post-hoc test of C+Reg-GP (c) vs Technical Analysis based trading strategies. 10-minute interval out-of-sample date. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
C+Reg-GP (c)	1.6250	-
AROON	**3.4500**	0.0185
RSI	**3.4750**	0.0185
SMA	**4.4750**	7.0153E-4
EMA	**4.5250**	7.0153E-4
MACD	**5.9999**	2.8281E-6
BOLLIN	**6.2250**	1.7251E-8
ROC	**6.7250**	3.2042E-10

Average MDD result presented in Table 5.16 shows AROON the most risk aversive strategy followed by C+reg-GP. AROON recording the least MDD in 14 currency pairs. Table 5.17 presents Friedman statistical test of mean MDD results. Tt shows AROON statistically outperforms other strategy and C+reg-GP is ranked second.

To measure the risk-reward trade-off, we perform Sharp ratio comparison. Figure 5.9 presents comparison result between Sharpe ratio result of C+Reg-GP and the technical indicator based strategies.

Figure 5.10 presents the total number of positive Sharpe ratio recorded by each algorithm. The result shows that C+Reg-GP recorded 28 positive Sharpe ratios and had the best Sharpe ratio measure in 18 of the 34 risk-adjusted return summaries.

Figure 5.9: Average Sharpe ratio for all currency pairs. C+Reg-GP versus technical analysis based trading strategies

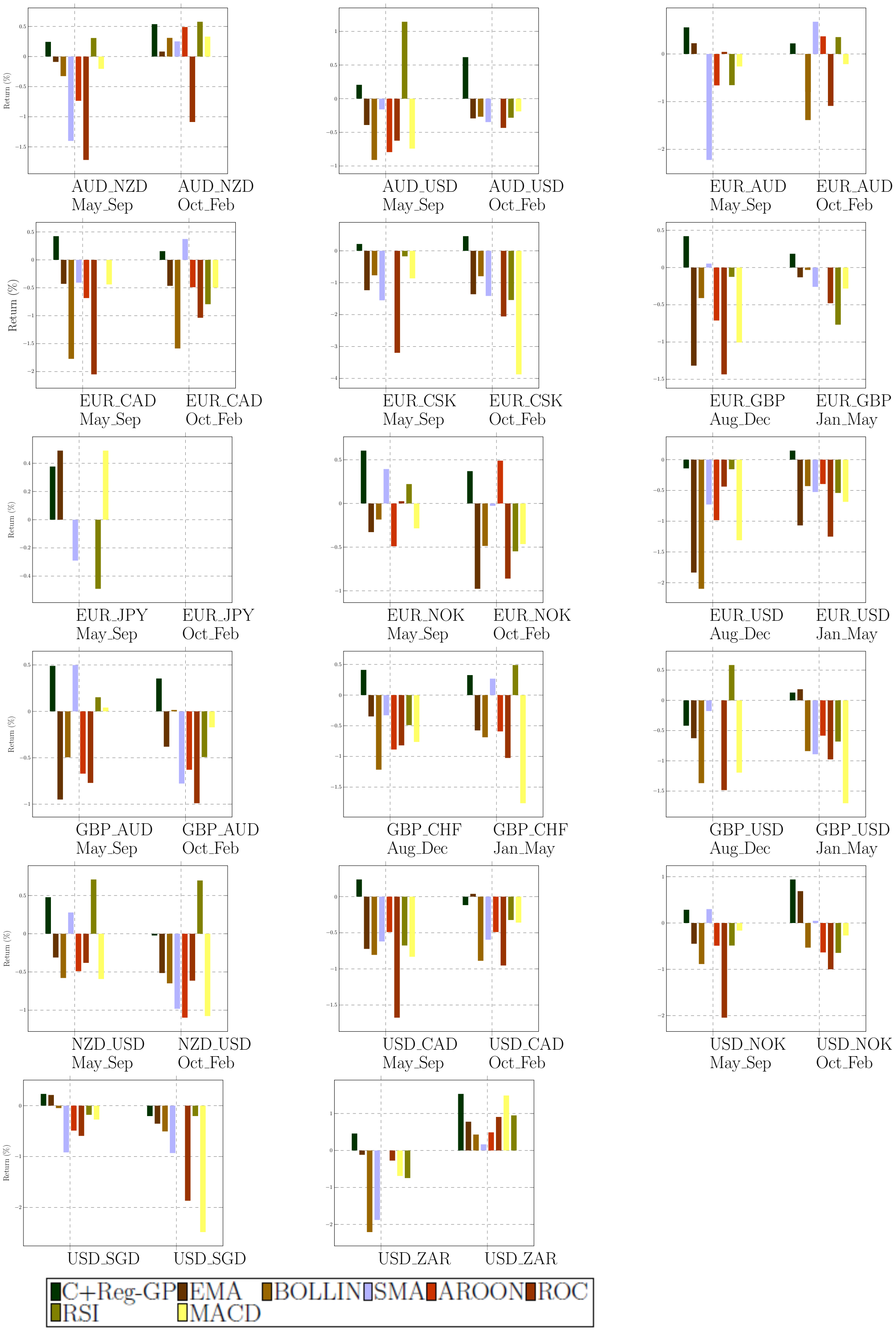


Table 5.16: Average maximum drawdown (%) results for Technical Indicator based strategies. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. five DC dataset were generated using five dynamically generated thresholds tailored to each DC dataset. Best result per currency pair shown in boldface.

Dataset	C+Reg-GP	EMA	BOLLIN	SMA	AROON	ROC	RSI	MACD
AUD_NZD	0.1230	0.1700	0.3680	0.2860	**0.0180**	0.6810	0.0710	0.1590
AUD_USD	0.1600	0.1180	0.6340	0.2280	**0.0250**	0.6220	0.1620	0.1700
EUR_AUD	0.1060	0.1200	0.4960	0.3800	**0.0410**	0.5080	0.1040	0.1730
EUR_CAD	0.1350	0.1230	0.8230	0.1720	0.0890	0.7480	**0.0800**	0.1480
EUR_CSK	0.0060	0.0360	0.0740	0.1660	**0.0030**	0.7830	0.1860	0.0390
EUR_GBP	0.1000	0.2420	0.3720	0.2040	**0.0640**	0.5320	0.0720	0.1960
EUR_JPY	0.0110	0.0040	**0.0000**	0.0400	**0.0000**	**0.0000**	0.0220	0.0010
EUR_NOK	0.1330	0.1140	0.4690	0.1860	**0.0040**	0.6880	0.0900	0.1010
EUR_USD	0.1550	0.2240	0.4950	0.2990	**0.0950**	0.4160	0.1910	0.2230
GBP_AUD	0.1910	0.2080	0.7310	0.2730	**0.0810**	1.0300	0.2830	0.2160
GBP_CHF	0.0960	0.1630	0.5250	0.2810	0.0870	0.7800	**0.0160**	0.2000
GBP_USD	0.1320	0.1220	0.8420	0.3400	**0.0770**	0.5080	0.1120	0.2700
NZD_USD	0.2890	0.2700	0.8610	0.3850	0.0260	0.6980	**0.0000**	0.2630
USD_CAD	0.1680	0.3230	0.8980	0.6090	**0.0240**	0.9060	0.3330	0.1670
USD_NOK	0.1410	0.2350	1.0510	0.2450	0.**0540**	0.9280	0.1730	0.0990
USD_SGD	0.0770	0.1270	0.3160	0.2670	**0.0150**	0.5900	0.1520	0.0790
USD_ZAR	0.1170	0.3570	1.4760	0.6280	**0.0000**	1.0690	0.1490	0.7090
Average MDD	0.126	0.1739	0.6136	0.2935	**0.0414**	0.6757	0.1292	0.1890

Table 5.17: Statistical test results of maximum drawdown of non-DC based trading strategies according to the non-parametric Friedman test with the Hommel post-hoc test. 10-minute interval out-of-sample data. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
AROON (c)	**1.2353**	-
C+Reg-GP	**3.2353**	0.0173
RSI	**3.3529**	0.0173
MACD	**4.0000**	0.0030
EMA	**4.0000**	0.0030
SMA	**5.9412**	1.0649E-7
BOLLIN	**7.0001**	4.0921E-11
ROC	**7.2353**	6.4656E-12

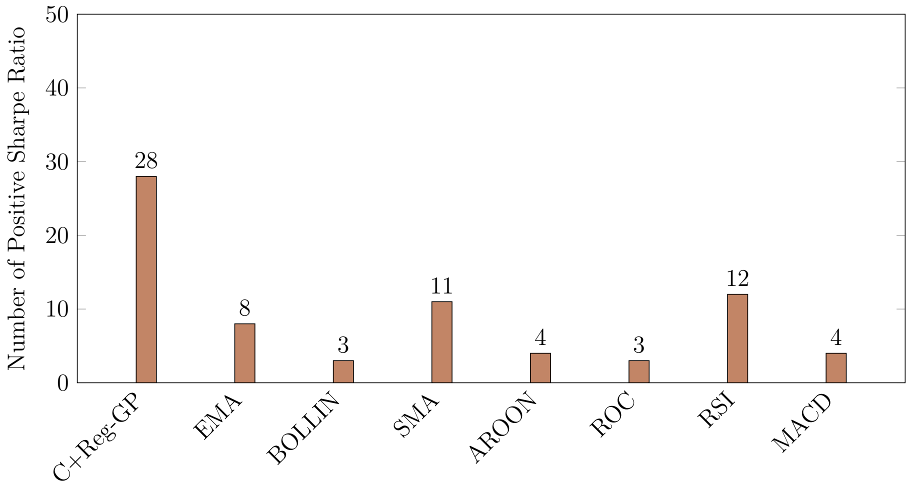


Figure 5.10: A comparison of total number of positive Sharpe Ratio between C+Reg-GP and technical indicator based trading approaches. 10-minute interval out-of-sample data. 20 currency pairs and 10 calendar months. Total of 40 Sharp ratio results from five month average

Finally, we perform Friedman statistical test of Sharpe ratio result presented in Table 5.18 which confirms our findings that C+Reg-GP outperformed all technical indicator based strategies and the performance is statistically significant at the 5% level.

Table 5.18: Statistical test results of Sharpe ratio of C+Reg-GP Vs non-DC based trading strategies according to the non-parametric Friedman test with the Hommel post-hoc test. 10-minute interval out-of-sample data. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
C+Reg-GP (c)	1.8286	-
RSI	**3.7429**	0.0011
EMA	**4.3571**	3.1439E-5
AROON	**4.4000**	2.3579E-5
SMA	**4.4857**	2.0959E-5
MACD	**5.1857**	4.9215E-8
BOLLIN	**5.5143**	1.8497E-9
ROC	**6.4857**	1.2683E-14

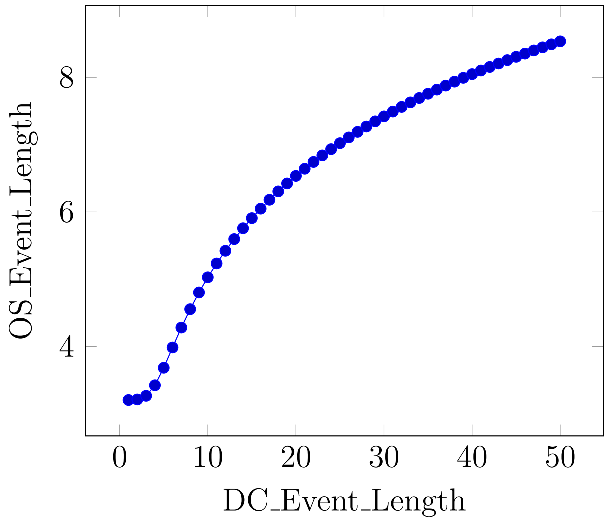
Comparison with Buy-and-hold

In terms of returns and risk, since C+Reg-GP was found to be the best algorithm in comparison with DC and technical analysis algorithms, we now turn our focus to comparing it against the well-known buy-and-hold (BandH) benchmark. We compare BandH separately because it is a fundamental analysis based strategy, a strategy that is not based on short-term price movements. We thus buy on the first day of the first month, and sell on the last day of the tenth month

Table 5.19 presents comparison trading result between C+Reg-GP and BandH. C+Reg-GP recorded positive mean returns in 15 of 20 currency pairs outperforming BandH in 12 currency pairs. C+Reg-GP's average return across all currency pairs was 0.225% and BandH was -0.128%; C+Reg-GP reported a variance of 0.153 and BandH's reported a variance of 0.515. Finally, we performed the Kolmogorov-Smirnoff statistical test to investigate whether there is a statistical significance in the results between C+Reg-GP and BandH. The p-value of the test was 7.2529e-04, which confirms that this difference is statistically significant. Therefore, the results show that C+Reg-GP can outperform BandH in more markets which makes it a more attractive investment strategy according to our data sample.

5.7.3 Sample of best GP models

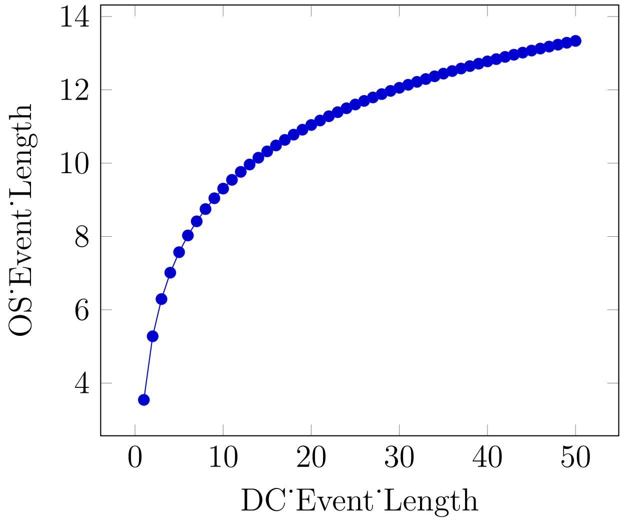
We present four samples equations that C+Reg-GP created and their plots. They are four of the best equations in terms of profitability across datasets experimented. In the equations, OS_l is the estimated OS event length and DC_l is DC event length.



$$OS_l = \log(a + DC_l^b) \tag{5.8}$$

where a= 1609.55 and b = 5.023.

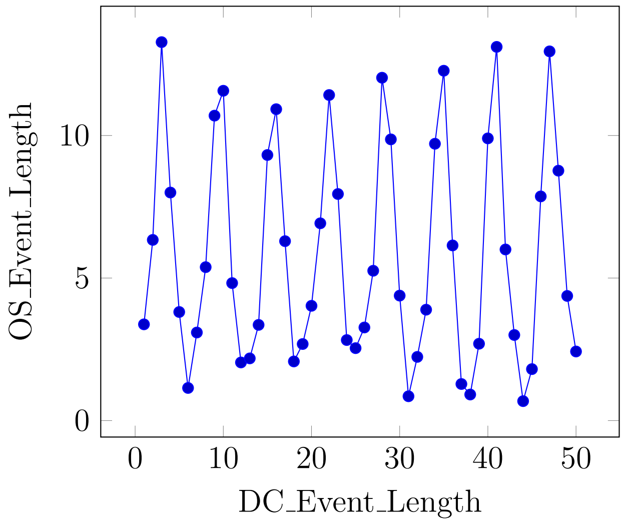
Figure 5.11: Equation 5.8 and the plot of OS event length and the corresponding DC event length over a range of 1 to 50.



$$OS_l = \log((DC_l \times a)^b) \tag{5.9}$$

where a= 4.117 and b = 5.764.

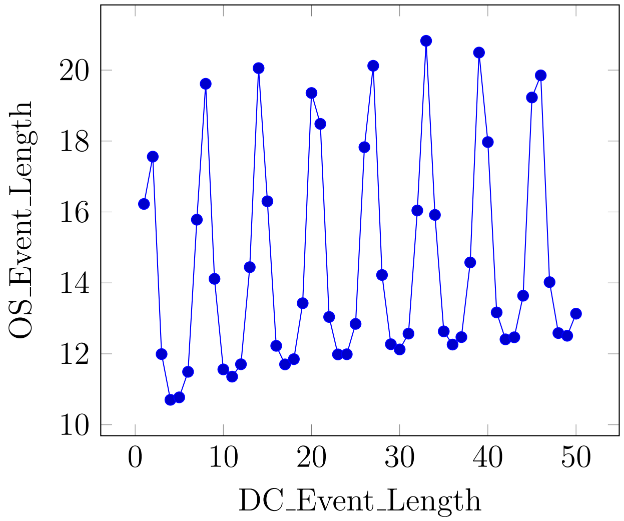
Figure 5.12: Equation 5.9 and the plot of OS event length and the corresponding DC event length over a range of 1 to 50.



$$OS_l = \cos(a \times \cos(DC_l)) + \frac{b}{exp(\cos(DC_l)} \tag{5.10}$$

where a = 292.160 and b= 4.569

Figure 5.13: Equation 5.10 and the plot of OS event length and the corresponding DC event length over a range of 1 to 50.



$$OS_l = \exp(\exp(\sin(\sin(DC_l)))) + (a \times (b + \log(DC_l))) \tag{5.11}$$

where a = 1.750 and b = 1.957

Figure 5.14: Equation 5.11 and the plot of OS event length and the corresponding DC event length over a range of 1 to 50.

As we can see, the equations have different structures; the first two are logarithmic equations, whereas the third has both cosine and exponential functions, and the fourth equation has exponential, sine and logarithmic functions. This is important because they confirm that the relationship between the DC and OS lengths can also be non-linear. Thus, our work of using C+Reg-GP to create equations for DC trends that are classified to have both DC and OS events has allowed uncover richer relationships and led to increase in profitability of the trading strategy.

5.7.4 Computational time

Table 6.11 presents the average computational times for all algorithms. We can observe that different algorithms can have significantly different computational times, which is not surprising. An algorithm such as C+Reg-GP includes the classification step, which consisted of Auto-WEKA running for 60 minutes to find the best classification model per dataset, and optimise its hyperparameters.[2] Additionally, it includes a GP, which requires some time to evolve an acceptable solution, since multiple individuals and generations are involved.

To elucidate, we present the computational times for each training task in our framework i.e., classification and (OS length) estimation. Not all algorithms used the classification step, but the ones that use it need approximately 65 minutes to complete model training task. The estimation task takes approximately 5-6 minutes for algorithms that use a GP to complete, and 20-30 seconds for the other algorithms. With regards to the trading step, all algorithms need around 3 seconds.

It is important to note here that, for trading, we would normally do the learning

[2] The time taken in the classification phase of C+Reg-GP, C+Factor-M, and C+Factor-2, went above the allotted time of 60 minutes due to CPU time slice as other processes were running on the hardware simultaneously. With the availability of a dedicated hardware with sufficient CPU cores, a large speed up might be obtained by switching the classification phase from serial mode to parallel mode.

processes on the training data off-line, and then simply apply the best model to the test data. Thus, the fact that classification and estimation last above 70 minutes is not a problem since they happen off-line. In contrast, applying the best model for trading takes only 3 seconds. Therefore, we believe that given the significant improvements we observed in returns and risk, this slower execution time is justified. Lastly, the overhead of including a classification step can be reduced by parallelising the Auto-WEKA process. It has been shown in the literature (e.g., (Ong and Schroder 2020)) that parallelisation can reduce computational times significantly.

5.8 Summary

Based on our experimental results, we can reach the following conclusions.

Introducing a classification step to a DC algorithm is an effective way of predicting the trend reversal in DC summaries. As we observed in Table 5.4, the positive classification results have led to significantly reduced RMSE, ranking each algorithm that uses a classifier higher than its respective variant without classification. In addition, C+GP ranked first and statistically outperformed all other DC-based trend reversal algorithms.

Introducing a classification step to a DC algorithm leads to higher returns during trading. As we observed in Tables 5.5, 5.6, 5.7 and 5.8, all algorithms that used a classifier (C+Reg-GP, C+Factor-M, C+Factor-2) outperformed other variants without a classifier. Furthermore, C+Reg-GP ranked first among 13 trading algorithms and statistically outperformed 12 algorithms, with the only exception the two other algorithms that were using a classifier.

Introducing a classification step to a DC algorithm leads to higher risk-adjusted return strategies. As we observed in the Sharpe ratio results, all the variants with the classifier ranked in the first 3 places (Table 5.13). This could potentially be attributed to the fact that the Sharpe ratio is a metric that includes both returns and risk. The MDD results presented a mixed picture, with C+Reg-GP ranking first across all algorithms, but Factor-M and Factor-2 ranking being higher than their variants with a classifier.

C+Reg-GP is an effective trading algorithm. It not only outperformed other DC-based algorithms, it also performed better than seven different technical indicator based strategies, as well as buy-and-hold in all metrics compared namely average returns, MDD and Sharpe ratio.

In our next contribution, we focus our attention on combining the best tailored thresholds under a single trading strategy. Our goal is to investigate whether using input from multiple DC thresholds to make trading decisions can lead to an increase in profitability while keeping risk at a minimum.

Table 5.19: Average trading (%) result of C+Reg-GP vs Buy-and-hold trading strategies per currency pair. 10-minute interval out-of-sample data. Results show RMSE value. They are averaged over five different dynamically generated thresholds tailored to each DC dataset and 20 currency pairs.

Trading strategies	**C+Reg-GP**	**Buy-and-hold**
AUD_JPY	**0.000**	-6.278
AUD_NZD	**0.260**	-0.516
AUD_USD	**0.273**	-5.728
CAD_JPY	**0.000**	-4.109
EUR_AUD	**0.186**	-2.672
EUR_CAD	0.192	**18.555**
EUR_CSK	0.034	**7.770**
EUR_GBP	**0.104**	-0.292
EUR_JPY	**0.020**	-6.211
EUR_NOK	0.351	**2.046**
EUR_USD	-0.001	**8.801**
GBP_AUD	0.354	**3.936**
GBP_CHF	**0.202**	-2.395
GBP_USD	-0.059	**8.464**
NZD_USD	**0.280**	-6.443
USD_CAD	0.044	**2.345**
USD_JPY	**0.000**	-9.430
USD_NOK	**0.461**	-6.102
USD_SGD	0.030	**0.207**
USD_ZAR	**1.762**	-4.505
Mean	**0.225**	-0.128

Table 5.20: Average computational times per run for C+Reg-GP, Reg-GP , p+Reg-GP, DCC+Reg-GP, C+Factor-M, Factor-M, p+Factor-M, DCC+Factor-M, C+Factor-2, Factor-2, p+Factor-2, DCC+Factor-2, RSI, EMA, MACD. BH takes less than 1 second to execute because we buy quoted currency at the start of trading period and sell quoted currency at the end of trading period.

Trading strategies	C+Reg-GP	Reg-GP	p+Reg-GP	DCC+Reg-GP
Classification (training)	$\sim$ 65 mins	–	–	–
regression (training)	$\sim$ 5.45 mins	$\sim$ 6.20 mins	$\sim$ 5.25 mins	–
Trading	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec
Trading strategies	C+Factor-M	Factor-M	p+Factor-M	DCC+Factor-M
Classification (training)	$\sim$ 65 mins	–	–	–
Regression (training)	$\sim$ 30 secs	$\sim$ 30 secs	$\sim$ 30 secs	–
Trading	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec
Trading strategies	C+Factor-2	Factor-2	p+Factor-2	DCC+Factor-2
Classification (training)	$\sim$ 65 mins	–	–	–
Regression (training)	$\sim$ 20 secs	$\sim$ 20 secs	$\sim$ 20 secs	–
Trading	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec
Trading strategies	EMA	BOLLIN	SMA	AROON
Classification (training)	–	–	–	–
Regression (training)	–	–	–	–
Trading	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec
Trading strategies	ROC	RSI	MACD	–
Classification (training)	–	–	–	–
Regression (training)	–	–	–	–
Trading	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec	$\sim$ 3 sec

Chapter 6

A Novel Multiple Threshold based Trading Strategy

In the previous chapter, we presented Figure 5.4 a trading strategy framework embedded with our proposed trend reversal forecasting model. The model consisted of a tailored classifier, which determines whether a DC trend is composed of either a DC and OS events (αDC) or just a DC event (βDC). If a DC trend is classified as αDC, we then used our tailored symbolic regression GP to estimate the expected length of the associated OS event. In this case, the forecasted trend reversal point occurs at the sum of the last known DC event length and the estimated OS event length measured from the start of the last known DC event. Conversely, if a trend is classified as βDC, we forecast trend reversal point to be the DCC point. The rationale behind forecasting trend reversal is to be able to anticipate reversal points whilst still in the trend, since directional changes are confirmed in hindsight and the ability to anticipate the points give traders opportunity to develop profitable strategies.

Despite the improvements we have made so far at both forecasting trend reversal points and trading profitably, the trading strategy framework in the previous chapter comes with a certain limitation. It views the market from a single trader's perspective thereby limiting insight into other market activities that influences price moment (Deng and Sakurai 2013) - i.e., once a threshold for sampling event summary is decided, we are locked-in at viewing the market activities from only the threshold's perspective. There are chances that potentially profitable price variations different from those captured by the threshold are ignored, leading to inaction or opportunity loss.

To address this issue and increase profit at reduced risk, we propose a novel multi-threshold trading strategy framework. This innovative approach allows a trader to view market activities from multiple perspectives before taking trading decisions. Thus, instead of selecting the best threshold from the threshold pool as was done for single threshold strategies in the previous chapter (see Figure 5.2), we instead select the best set of thresholds and optimise their trading recommendations. In this new approach (illustrated in Figure 6.1), trading actions and forecasted trend reversal points by individual thresholds become recommendations only. Same recommendations are combined, and trading action is decided through a majority voting system. However, there are some associated challenges that combining recommendations pose: (1) how to select the best set of threshold adequately; (2) how to determine whether a DC event is either αDC or βDC; (3) how to decide on a trading action when recommendations from multiple thresholds are conflicting; and (4) how to decide on the trend reversal point to trade when forecasted trend reversal points of multiple thresholds are different? For the first two challenges, we address them using similar approach as the one from Chapter 5. For the last two challenges, we address them with the new framework presented in this chapter. Our contribution in this chapter can thus be summarised as the creation of a DC trading strategy using recommendations from multiple DC thresholds that are optimised by a tailored GA algorithm.

We organize the rest of the chapter as follows: Section 6.1 describes the new trading strategy and the methodology employed. Section 6.2 presents the experimental and evaluation setup. Section 6.3 presents the test results. Section 6.4 concludes the chapter

6.1 Methodology

To address the two outstanding challenges, highlighted in the introduction to this chapter, we associate initial weights to the contributing trend reversal prediction models and apply GA to optimise their weight value. We now present our approach in detail.

6.1.1 Optimised multi-threshold strategies using a Genetic Algorithm

Our multi-threshold trading strategy framework is composed of 4 modules: (1) a pre-step for threshold selection; (2) symbolic regression model evolution; (3) classification model selection; and (4) trading strategy evolution. The first 3 modules of our framework follow the same approach described in Chapter 5. After creating forecasting model (i.e., classification and symbolic regression GP models combined) per threshold, we embedded them in our trading strategy and assigned weights to represent each threshold. Each chromosome consisted of N_θ genes, where N_θ is the

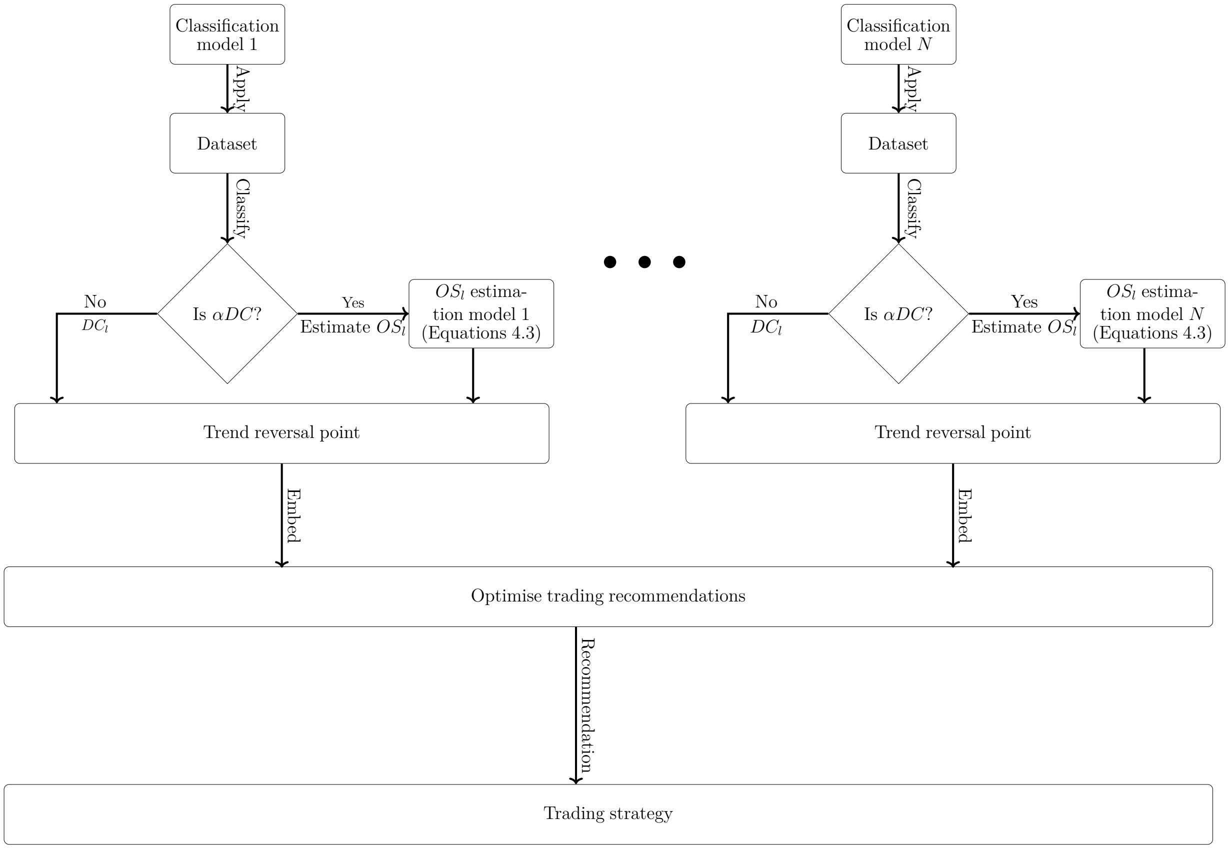


Figure 6.1: Our proposed framework for a multi-threshold based trading strategy. It embeds multiple thresholds and combines their recommendations using a majority vote system. It uses weighted average of contributing thresholds' forecast of trend reversal point in deciding when to trade.

number of thresholds used in the multi-threshold strategy. As a first step, we decided on the number of thresholds to use in the multi-threshold strategy. Then we assigned an initial weight to each gene. The weight is a measure of the importance of a forecasting model's recommendation in trading decisions. The weights are real values where the maximum weight value is one and the minimum value is zero.

To ensure that in the worst case scenario the GA performs as well as the highest ranked trend reversal forecasting model in the set we aimed to optimise, we seeded

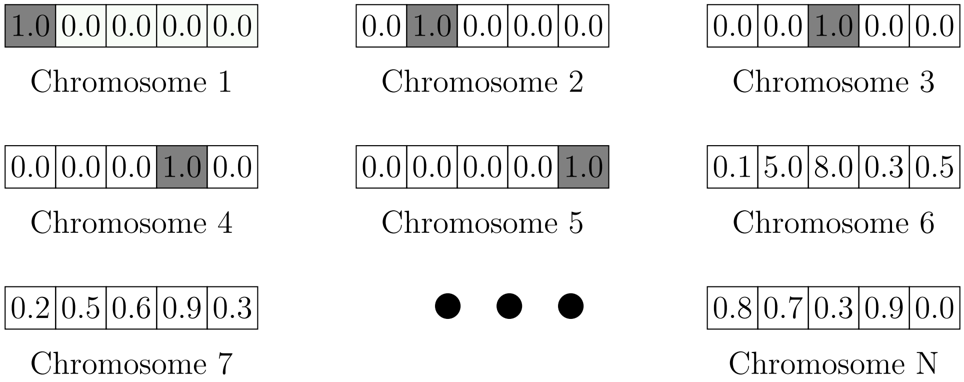


Figure 6.2: Illustration of GA population initialisation. Chromosomes 1-5 represents initialisation where only a single threshold is active

the first N chromosomes corresponding to the number of models being optimised. In each of these chromosomes, we activated one gene, setting the value to the maximum weight (i.e., 1) and associated it to a single threshold forecasting model while we set the value of the remaining genes to the minimum weight value (i.e., 0) to deactivate them. The remaining chromosomes in the population were randomly initialised by assigning real values between the minimum and maximum weights (inclusive). Figure 6.2 exemplifies the initialisation process. In the example, five trend reversal forecasting models were optimised, genes one, two, three, four, five were activated in chromosomes one, two, three, four and five respectively. The pseudocode presented in Algorithm 6.1 summarises the initialisation step after which the GA evolved real value weights for each threshold over a number of generations using training dataset. At the end of the evolution process our optimised model is created.

During trading, thresholds traverse their own event series. At each DCC point of the current trend in individual event series, trend reversal points are forecasted. First, each thresholds classifies DC trend as either αDC or βDC. Then, the trend reversal forecasting models use their symbolic regression GP to forecast trend reversal

Algorithm 6.1 Pseudocode for initialising chromosome weight in GA population

```
  for i = 0; i < numberOfThresholds; i++ do
      for j = 0; j < chromosomeInPopulation[$i$]; j++ do
          if index $i$ is threshold position in chromosome then
              $w_i \leftarrow 1.0$
          else
              $w_i \leftarrow 0.0$
      end if
  end for
end for
for i = numberOfThresholds; i < chromosomeInPopulation; i++ do
  for j = 0; j < chromosomeInPopulation[$i$]; j++ do
      $w_i \leftarrow RandomNumberFunction(0.0, 1.0)$
  end for
end for
```

points of trends classified as αDCs. Otherwise, trend reversal points are forecasted to occur at a DCC point. The recommended action and forecasted trend reversal point of all thresholds are passed to the multi-threshold trading strategy. The trading strategy uses the recommendations optimise decisions: (1) the action to follow; and (2) when in the future to act.

Recommended trading actions can be different because each threshold's event series is unique. Therefore, it is possible for one threshold to recommend a buy action while another simultaneously recommend a sell action. The strategy evaluates recommended actions according to the weights associated with each threshold. Weights of actions that are the same are summed up and the action with the highest sum of weight is followed, which we call "optimal trade action" (TA_{Opt}). Also, because of the distinct nature of individual event series, the multiple thresholds from which TA_{Opt} is based can forecast trend reversal points differently. The strategy uses Equation 6.1 to optimise individually forecasted trend reversal point of a subset of thresholds that recommended TA_{Opt}. It considers the weights of the thresholds, so the optimised forecasted trend reversal points tend towards the threshold with the largest weight. Algorithm 6.2 summarises the procedure of optimising trading actions and trend reversal points. Algorithms 6.3 and 6.4 summarise the trading rules applied at optimised trend reversal point at optimal trade action buy and sell respectively.

$$W = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i} \tag{6.1}$$

The types of action the strategy takes are buy, sell and hold. We consider the first two actions to be active actions and hold to be a passive action. Thus, if the winner action is a sell, we sell all available base currency in exchange for the quoted currency at the calculated trend reversal point. On the other hand, if the winner action is a buy, we buy all available base currency in exchange for the quoted currency at the calculated trend reversal point. Therefore, we do not have a situation where we have both base currency and quoted currency in our portfolio. Our action is passive (1) if the action is sell and there isn't enough base currency available to sell or (2) the action is buy and there isn't enough base currency to buy or (3) if the return is negative after deducting transaction cost.

We now clarify our proposed strategy with an example. Let's assume it is decided to use 5 thresholds for sampling significant events in the market. Due to aforementioned reasons, the recommendations of each threshold can be different. Thus, we use 5 gene GA algorithm to optimise the recommendations. Each gene in the GA is a weight to be associated with the recommendation of each threshold. Suppose at the end of the GA evolutions step, weights [0.3, 0.35, 0.1, 0.1, 0.15] are assign to each threshold respectively, coincidentally, all 5 thresholds consider a certain point in the dataset to be a DCC point. The first two thresholds [0.3 and 0.35] recommend a buy action with a weight sum of 0.65 and the last three thresholds [0.1, 0.1 and 0.15] recommend a sell action with a weight sum of 0.35. To resolve the divergence in the recommended action, we apply a majority voting system, and, in this case, the strategy will follow the buy action because of the larger weight sum. To determine the optimal trend reversal point, we calculate the weighted average of forecasted trend reversal points by the first two thresholds that recommended the optimal trading action (i.e., 'buy' in this particular case). Assume that the current DCC point in this example coincides with data point 290 and the forecasted trend reversal points by thresholds [0.3, 0.35] are 312 and 300, respectively. We thus make a decision to perform a buy action at data point 305, which is the weight sum of their forecasted trend reversal points derived by applying Equation 6.1 calculated as $\frac{(0.3\times312)+(0.35\times300)}{(0.3+0.35)}$.

6.1.2 Genetic Operators

We use three operators namely elitism, uniform crossover and uniform mutation. For elitism, we copy the chromosome with the best fitness value into the next generation. For uniform crossover and uniform mutation, individuals from the population are selected into a mating pool. From the pool, through tournament selection, individuals that best favour the optimisation goal are selected as parents of individuals for the next generation. In this work we select as parent, individual in the pool with highest fitness. In uniform crossover both parents contribute their genes where each gene has a fixed probability of 0.5 of being swapped. In uniform mutation operation, the selected parent's gene have a fixed probability of 0.5 of being swapped as well. Figures 6.3 and 6.4 illustrate our uniform crossover and uniform mutation respectively.

Algorithm 6.2 Pseudocode for Multi-threshold Optimisation

```
Require: Initialise base_quantity = budget, quote_quantity = 0.0
Require: current_price = 0.0, LastUpPrice = 0.0
Require: Initialise weight values $W_1$, $W_2$, $W_3$ ... $W_{N_\theta}$ according to Algorithm 6.1
Require: Get forecast model $F_1$, $F_2$, $F_3$ ... $F_{N_\theta}$ for each threshold
  for i = 0; i < $dataset\_length$ ; i++ do
      Initialise forecast and action dictionary: Dict = empty
      Initialise weights for buy and sell: $W_B = W_s = 0$
      Initialise buy and sell trend reversal list: $List_B = List_S$ = empty
      for j = 0; j < $N_\theta$; j++ do
          Initialise trend reversal point: TRP = 0.0
          if event is upturn && DCC_point then
              $TRP \leftarrow F_j$
              $Insert\ TRP\ into\ List_S$
              $W_S \leftarrow W_S + W_j$
          else if event is downward trend && DCC_point then
              $TRP \leftarrow F_j$
              $Insert\ TRP\ into\ List_B$
              $W_B \leftarrow W_B + W_j$
          end if
      end for
      if $W_S > W_B$ then
          $TRP_{optimal_i} \leftarrow$ $optimise\ List_S$ according to Equation 6.1
          $Insert\ TRP_{optimal_i}\ and\ Sell\ into\ Dict\ at\ position\ i$
      else
          $TRP_{optimal_i} \leftarrow$ $optimise\ List_B$ according to Equation 6.1
          $Insert\ TRP_{optimal_i}\ and\ Buy\ into\ Dict\ at\ position\ i$
      end if
      if Dict[i] is not empty then
          if Dict[i] == $Sell$ then
              $current\_price \leftarrow dataset\_length_{i[ask]}$
              Trade with Sell Rule [See Algorithm 6.3]
          else if Dict[i] == $Buy$ then
              $current\_price \leftarrow dataset\_length_{i[bid]}$
              Trade with Buy Rule [See Algorithm 6.4]
          end if
      end if
  end for
  $Wealth \leftarrow base\_quantity - budget$
  $Return \leftarrow 100 \times \frac{Wealth}{budget}$
```

Algorithm 6.3 Trading rules used for selling the base currency

```
Require: Sell rule
  if $base\_quantity > 0$ then
      $base\_quantity \leftarrow base\_quantity$ - $transaction\_Cost$
      $quote\_quantity \leftarrow base\_quantity \times current\_price$
      $base\_quantity \leftarrow 0.0$
      $LastUpPrice \leftarrow current\_price$
  else Hold
  end if
```

Algorithm 6.4 Trading rules used for buying the base currency

```
Require: Buy rule
  if $quote\_quantity > 0$ && current_price < LastUpPrice then
      $quote\_quantity \leftarrow quote\_quantity$ - $transaction\_Cost$
      $base\_quantity \leftarrow \frac{quote\_quantity}{current\_price}$
      $quote\_quantity \leftarrow 0.0$
  else Hold
  end if
```

We measure the quality of our GA individual using Sharpe ratio presented in Section 5.5. We choose Sharpe ratio because it is an aggregate metric of risk-adjusted return, as it takes into account both the return and the risk of a given trading strategy.

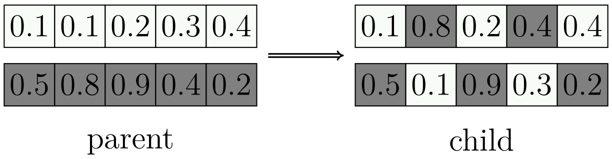


Figure 6.3: A sample uniform crossover operation by our GA. Either of the children is randomly selected for the next generation

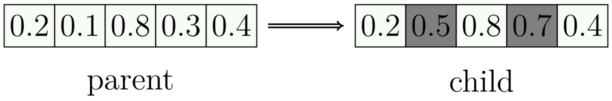


Figure 6.4: A sample uniform mutation operation by our GA .

6.2 Experimental Setup

Data

The same 10-minute interval high frequency data already described in Chapter 5 are used for experimenting our proposed multi-threshold strategy. We considered each month in the period as a separate physical-time dataset. In our tuning phase, we used 200 DC datasets for tuning (i.e., 5 DC thresholds × 20 currency pairs × first 2 months of our physical-time data). For the rest of the experiment, we use 1000 DC datasets (i.e. 5 DC thresholds × 20 currency pairs × remaining 10 months of our physical time datasets). The tuning and non-tuning DC datasets were split in 70:30 ratio as training and testing sets.

Parameter tuning

Since we are building on our previous contribution in Chapter 5 and the trend reversal forecasting model should remain unchanged, we use the same parameter setup for our classifier and symbolic regression GP models. Auto-WEKA execution time to select and configure tailored classification model is set to 60 minutes and Table 6.1 presents the GP configuration to evolve the symbolic regression model for estimating the OS event length.

To determine the values for the parameters in our GA algorithm, we performed parameter tuning using I/F-Race package (López-Ibánez et al. 2011), already described in Chapter 4. The tuned parameters are population size, generation size,

Table 6.1: Regression GP experimental parameters for detecting DC-OS relationship, determined using I/F-Race.

Parameter	
Population	500
Generation	37
Tournament size	3
Crossover probability	0.98
Mutation probability	0.02
Maximum depth	3
Elitism	0.10

tournament size, crossover probability, mutation probability and elitism. Table 6.2 present the value of our tuned parameters. We did not tune the number of thresholds instead we choose the same number of thresholds as our previous contributions in Chapter 4 (5 threshold) to facilitate comparison.

Table 6.2: GA experimental parameters for multi-threshold trading strategy determined using I/F-Race.

Parameter	
Population size	500
Generation size	50
Tournament size	7
Crossover probability	0.90
Mutation probability	0.10
Elitism	1

Trading Experimental Setup

We embedded the 5 trend reversal forecasting models in the trading strategy already described in Section 6.1.1. The strategy combines these models to make trading decisions. Our goal in this contribution is to investigate whether our multi-threshold trading strategy can outperform the best performing single threshold strategy. Thus, we compare the performance of our multi-threshold strategy against the 5 best performing single threshold strategies. The single thresholds are the same thresholds that make up the genes of the GA chromosome.

6.3 Trading Results

In this section, we present the summary of our experimental result. As a reminder, the goal of this contribution is to demonstrate that by optimising recommendations from multiple thresholds using machine learning techniques we can further improve profitability and risk, statistically outperforming single threshold strategies.

Table 6.3 presents returns of single threshold and multi-threshold trading strategies calculated monthly. In this table, cases where 0.00 is reported as return indicates that the strategy is passive (i.e., hold action). Trading return results show that the multi-threshold strategy has the highest return (1.15%), which is over 100% better than the best single threshold strategy that recorded return of 0.53%. The 1.15% return is earned over a month period, annualising the return results in 14.707% return in a year. The result of the multi-threshold strategy is also the best per currency pair (highlighted in bold in Table 6.3. Table 6.4 present the non-parametric Friedman test with the Hommel post-hoc test to determine if the differences in performance are statistically significance. The null hypothesis is again that the strategies come from the same continuous distribution. As we could observe, the best ranking strategy was the multi-threshold strategy, and it statistically outranked the 5 single threshold strategies at the 5% significance level in all pairs.

We evaluated our risk adjusted return over the transactions that occurred in the 10-minutes monthly dataset. Table 6.5 presents the result, and it shows that multi-threshold strategy outperformed single threshold strategy in all 20 currency pairs.

The Sharpe ratio of 0.78 is over 200% better than the Sharpe ratio of the best single threshold strategy. We also tested the statistical significance of the Sharpe ratio result using Friedman nonparametric test. The null hypothesis is that the strategies come from the same continuous distribution. We reject the null hypothesis because the statistical test results presented in Table 6.6 shows that multi-threshold strategy outperformed the 5 single threshold strategies.

We also performed risk analysis, measuring maximum drawdown and standard deviation of our daily return. Table 6.7 presents the maximum drawdown results, where the lower the drawdown the better the result. Our multi-threshold strategy recorded the lowest overall average maximum drawdown (0.02). On average, the risk was 10 times lower than trading using single threshold strategies. We also perform Friedman test and Table 6.8 presents the result that shows that multi-threshold strategy statistically outperformed all single threshold strategies at the 5% significance level.

Finally, Table 6.9 presents our standard deviation results. The results are not as homogenous as in the previous tables, where the multi-threshold strategy is ranking first across all datasets. Nevertheless, the multi-threshold strategy remained ranking the highest for the number of currency pairs(7), it has the lowest average standard deviation (0.1638). We also performed Friedman statistically test, presented in Table 6.10. The results show that multi-threshold strategy ranks first overall, although the performance was not statistically significant against any of the single threshold strategies. It appears that the Sharpe ratio, which is the fitness function of our GA and thus drives the search, is heavily affected by the non-homogeneity of the standard deviation result, where we are unable to record statistically significant result against the single threshold strategies. In terms of standard deviation, it appears that the profit volatility is relatively similar across the different strategies even though we see slight improvements using multi-threshold strategy. It is also important to remember that in terms of a different risk metric (MDD), we have observed that the multi-threshold strategy is outperforming the individual thresholds across all currency pairs.

Table 6.3: Average return result (%) for trading strategies of individual single threshold strategies and multi-threshold strategy. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. 5 DC dataset were generated using 5 dynamically generated thresholds tailored to each DC dataset. Best value for each row (currency pair) is shown in boldface.

Dataset	**Threshold1**	**Threshold2**	**Threshold3**	**Threshold4**	**Threshold5**	**Multi-threshold**
AUD_JPY	0.9032	1.1177	1.0361	1.0132	1.2644	**1.4018**
AUD_NZD	0.4716	0.4831	0.3926	0.3365	0.2377	**1.1877**
AUD_USD	0.3970	0.5281	0.5813	0.7310	0.7253	**0.8701**
CAD_JPY	0.8736	0.8969	0.8264	0.7082	0.7935	**1.3208**
EUR_AUD	0.6808	0.5261	0.3586	0.3850	0.3508	**1.0787**
EUR_CAD	0.4677	0.3900	0.3471	0.4886	0.4250	**0.9773**
EUR_CSK	0.0232	0.0372	0.0025	0.0474	0.0432	**0.3955**
EUR_GBP	0.2132	0.2712	0.0583	0.2139	0.2121	**0.8233**
EUR_JPY	0.5475	0.5171	0.4380	0.5985	0.5385	**0.8509**
EUR_NOK	0.2632	0.4388	0.3222	0.6373	0.2553	**0.8889**
EUR_USD	0.2139	0.2427	0.1494	0.1022	0.0777	**1.0474**
GBP_AUD	0.5770	0.3854	0.5816	0.7964	0.6471	**1.4298**
GBP_CHF	0.2575	0.0779	0.6074	0.1904	0.3013	**0.5371**
GBP_USD	0.1141	0.1997	0.0648	0.2228	0.1140	**0.8567**
NZD_USD	0.5130	0.5937	0.7069	0.7858	0.5984	**0.9422**
USD_CAD	0.2078	0.1658	0.4274	0.3773	0.4194	**0.8522**
USD_JPY	0.4411	0.6448	0.3829	0.3914	0.3428	**1.2062**
USD_NOK	0.3836	0.4253	1.0093	0.4595	0.4502	**1.5360**
USD_SGD	0.1525	0.1325	0.2305	0.2991	0.3777	**0.7704**
USD_ZAR	1.5811	1.4437	1.8097	1.7583	1.4155	**4.1808**
Average	0.4641	0.4759	0.5167	0.5271	0.4795	**1.1577**

6.3.1 Computational time

Table 6.11 presents the average computational time for multi-threshold strategy in comparison to single threshold strategy. The results show an increase in computation time taken by multi-threshold strategy. This is expected since it includes the time required to train multiple classification models. Additional time is also used in

Table 6.4: Statistical test results for average returns according to the non-parametric Friedman test with the Hommel post-hoc test of multi-threshold (c) vs other single threshold based trading strategies. 10-minute interval out-of-sample date. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
Multi-threshold (c)	1.0500	-
Threshold4	**3.3000**	1.4284E-4
Threshold2	**3.9999**	1.2302E-6
Threshold3	**4.1000**	7.5910E-7
Threshold1	**4.2500**	2.5353E-7
Threshold5	**4.3000**	1.5846E-7

Table 6.5: Average Sharpe ratio result for trading strategies of individual single threshold strategies and multi-threshold strategy. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. 5 DC dataset were generated using 5 dynamically generated thresholds tailored to each DC dataset. Best value for each row (currency pair) is shown in boldface.

Dataset	**Threshold1**	**Threshold2**	**Threshold3**	**Threshold4**	**Threshold5**	**Multi-threshold**
AUD_JPY	0.3469	0.2082	0.2335	0.2245	0.2451	**0.7026**
AUD_NZD	0.2565	0.2129	0.2289	0.1246	0.3348	**0.7912**
AUD_USD	0.2749	0.2183	0.3149	0.3396	0.3702	**0.8014**
CAD_JPY	0.2614	0.1879	0.3268	0.1664	0.2708	**0.6804**
EUR_AUD	0.2812	0.2310	0.2358	0.2855	0.2961	**0.9101**
EUR_CAD	0.3972	0.1807	0.2865	0.3229	0.2964	**0.7496**
EUR_CSK	0.0970	0.1190	0.0370	0.1893	-0.0555	**1.2658**
EUR_GBP	0.0845	0.0035	0.1589	0.1077	0.2287	**0.7330**
EUR_JPY	0.3539	0.3183	0.3371	0.4049	0.2846	**1.0389**
EUR_NOK	0.1292	0.2177	0.2578	0.2430	0.2778	**0.5835**
EUR_USD	0.2370	0.1381	0.1073	0.1328	0.1258	**0.5673**
GBP_AUD	0.2579	0.2179	0.2326	0.2619	0.3402	**0.9387**
GBP_CHF	0.2793	0.0216	0.3019	0.2840	0.2367	**0.7413**
GBP_USD	0.0779	0.2178	0.1344	0.2539	0.1855	**0.6961**
NZD_USD	0.1753	0.2463	0.2388	0.3418	0.2365	**0.6223**
USD_CAD	0.1780	0.3044	0.3232	0.3508	0.2181	**0.6328**
USD_JPY	0.2140	0.2205	0.0582	0.2940	0.2303	**0.6499**
USD_NOK	0.2614	0.2526	0.3395	0.1712	0.2156	**0.7604**
USD_SGD	0.0434	0.1260	0.1219	0.1236	0.1910	**0.7305**
USD_ZAR	0.2555	0.2741	0.2420	0.2576	0.2401	**0.9430**
Average	0.2231	0.1958	0.2259	0.2440	0.2384	**0.7769**

Table 6.6: Statistical test results for average Sharpe ratio according to the non-parametric Friedman test with the Hommel post-hoc test of multi-threshold (c) vs other single threshold based trading strategies. 10-minute interval out-of-sample date. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
Multi-threshold (c)	1.0000	-
Threshold4	**3.4500**	3.4541E-5
Threshold5	**3.7000**	1.0046E-5
Threshold3	**4.1000**	4.8184E-7
Threshold1	**4.2000**	2.5353E-7
Threshold2	**4.5500**	9.8300E-9

Table 6.7: Average Maximum drawdown (%) result for trading strategies of individual single threshold strategies and multi-threshold strategy. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. 5 DC dataset were generated using 5 dynamically generated thresholds tailored to each DC dataset. Best value for each row (currency pair) is shown in boldface.

Dataset	**Threshold1**	**Threshold2**	**Threshold3**	**Threshold4**	**Threshold5**	**Multi-threshold**
AUD_JPY	0.7447	0.2441	0.2796	0.2773	0.5053	**0.0262**
AUD_NZD	0.3235	0.2914	0.2642	0.2910	0.1143	**0.0177**
AUD_USD	0.2810	0.1617	0.2001	0.3173	0.2748	**0.0261**
CAD_JPY	0.1864	0.2537	0.2720	0.1687	0.3897	**0.0124**
EUR_AUD	0.5627	0.4365	0.0977	0.2828	0.2400	**0.0087**
EUR_CAD	0.2956	0.3051	0.0928	0.0964	0.1094	**0.0293**
EUR_CSK	0.0007	0.0302	0.0076	0.0383	0.0706	**0.0000**
EUR_GBP	0.1635	0.2833	0.0445	0.1823	0.1492	**0.0077**
EUR_JPY	0.2932	0.3777	0.2856	0.3772	0.3927	**0.0391**
EUR_NOK	0.1774	0.2326	0.1978	0.4144	0.0894	**0.0112**
EUR_USD	0.1499	0.2006	0.0973	0.0832	0.0487	**0.0303**
GBP_AUD	0.3190	0.2690	0.4058	0.4627	0.3772	**0.0074**
GBP_CHF	0.1020	0.1239	0.4167	0.1069	0.0923	**0.0035**
GBP_USD	0.1311	0.1223	0.0753	0.1477	0.0598	**0.0057**
NZD_USD	0.1884	0.1971	0.2058	0.2131	0.1720	**0.0342**
USD_CAD	0.1451	0.0469	0.2685	0.1434	0.3030	**0.0353**
USD_JPY	0.2563	0.3516	0.2688	0.3132	0.1097	**0.0160**
USD_NOK	0.2655	0.3375	0.6848	0.3467	0.3476	**0.0243**
USD_SGD	0.0383	0.0354	0.1351	0.1351	0.1890	**0.0071**
USD_ZAR	1.1300	1.0217	1.3708	1.3680	1.0740	**0.0196**
Average	0.2877	0.2661	0.2835	0.2883	0.2554	**0.0181**

Table 6.8: Statistical test results for average maximum drawdown according to the non-parametric Friedman test with the Hommel post-hoc test of multi-threshold (c) vs other single threshold based trading strategies. 10-minute interval out-of-sample date. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
Multi-threshold (c)	1.0000	-
Threshold5	**3.7500**	3.3460E-6
Threshold3	**3.9000**	1.8983E-6
Threshold1	**3.9000**	1.8983E-6
Threshold2	**4.0000**	1.2655E-6
Threshold4	**4.4500**	2.7455E-8

Table 6.9: % Average Standard Deviation (SD) result for trading strategies of individual single threshold strategies and multi-threshold strategy. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. 5 DC dataset were generated using 5 dynamically generated thresholds tailored to each DC dataset. Best value for each row (currency pair) is shown in boldface.

Dataset	**Threshold1**	**Threshold2**	**Threshold3**	**Threshold4**	**Threshold5**	**Multi-threshold**
AUD_JPY	0.4511	0.5528	0.5686	0.5502	**0.4419**	0.5334
AUD_NZD	0.2481	0.1745	0.1339	**0.0882**	0.0939	0.1048
AUD_USD	0.2142	0.2355	0.2512	0.2939	0.3751	**0.1945**
CAD_JPY	0.3759	0.3256	0.3656	0.2963	**0.2130**	0.3167
EUR_AUD	0.2798	0.2698	0.1849	0.1667	**0.1491**	0.1571
EUR_CAD	0.2184	**0.1827**	0.1910	0.2509	0.1993	0.2714
EUR_CSK	0.0144	0.0247	**0.0087**	0.0301	0.0380	0.0418
EUR_GBP	0.0898	0.1465	**0.0250**	0.0846	0.0802	0.0812
EUR_JPY	0.2309	0.2323	0.2104	0.2762	0.2718	**0.1408**
EUR_NOK	0.1155	0.1676	0.0993	0.1956	0.1349	**0.0863**
EUR_USD	0.0898	0.1326	0.0859	**0.0577**	0.0311	0.0884
GBP_AUD	0.2618	**0.1671**	0.2520	0.3044	0.2601	0.1867
GBP_CHF	**0.1021**	0.1575	0.2216	0.1277	0.1421	0.1647
GBP_USD	0.1104	0.1164	0.0841	0.1188	**0.0993**	0.1073
NZD_USD	0.2209	0.2201	0.2944	0.3090	0.2113	**0.1483**
USD_CAD	0.1218	**0.0676**	0.2414	0.1647	0.2023	0.1356
USD_JPY	0.2053	0.2749	0.2051	0.1725	0.1658	**0.1186**
USD_NOK	0.1543	0.1999	0.4327	0.1649	0.2033	**0.1299**
USD_SGD	**0.0651**	0.0721	0.0868	0.1636	0.1629	0.0925
USD_ZAR	0.4146	0.4244	0.5034	0.6268	0.3746	**0.1764**
Average SD	0.1992	0.2072	0.2223	0.2221	0.1925	**0.1638**

Table 6.10: Statistical test results for average Standard deviation according to the non-parametric Friedman test with the Hommel post-hoc test of multi-threshold (c) vs other single threshold based trading strategies. 10-minute interval out-of-sample date. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p value is lower than 0.05.

Trading strategies	Average Rank	$Adjust_{pHomm}$
Multi-threshold (c)	2.6999	-
Threshold5	3.0500	0.5541
Threshold1	3.5500	0.3016
Threshold3	3.6500	0.2262
Threshold2	3.8500	0.2010
Threshold4	4.2000	0.0561

Table 6.11: Average computational times per trend for single threshold strategy and multi-threshold strategy

Trading strategies	Single threshold	Multi-threshold
Classification	~ 65 mins	~ 330 mins
Estimation	~ 5.45 mins	~ 5.45 mins
GA optimisation	—-	~ 7 mins
Trading	~ 3 secs	~ 9 secs

training our GA based strategy. The computation time was measured on a non-dedicated[1] Red Hat Enterprise Linux (Maipo) with a 24 core, 2.53 GHz processor and 24 Gigabit memory. Although auto-WEKA, the tool for our classification step can be executed using multiple threads of concurrent execution, we chose to run serial mode using a single CPU core due to limitation on hardware resources. Beside the classification step, we acknowledge that improvements can be made in computation time through parallelisation of the different steps that make up the trading strategy framework (Brookhouse, Otero and Kampouridis 2014; Ong and Schroder 2020). We do not consider the additional time to be a significant drawback as the framework

[1] There were other processes unrelated to the experiment running on the server at the time the experiments were performed

is used off-line, therefore the significant improvement observed in trading results, outweigh any extra computational time needed.

6.4 Summary

Based on our experimental results, we are able to reach the following conclusion.

Viewing data from multiple perspectives augments insight into price movement. As we observed in Tables 6.3 and 6.5, profit obtained trading using a multi-threshold strategy outperformed single threshold 2 and 4 folds respectively. The statistical test performed showed that that the increase in profit is statistically significant. In addition, having better insight into price movement enables traders make better decisions without increasing risk. We were able to achieve afore mentioned profit without increasing risk. As we observed in Table 6.10, although multi-threshold strategy was unable to statistically outperform single thresholds in standard deviation risk measure, it was ranked first, and we consider this to be a positive result.

Optimisation of individual threshold recommendation is beneficial. Optimising both the trading actions and the forecasted trend reversal point from multiple thresholds using machine learning techniques is an effective way of developing profitable strategies without increasing risk. We also observed that Genetic algorithm, an optimisation technique is a tool that can be used in performing multiple recommendation optimisation successfully.

Chapter 7

Conclusion

In this thesis, we focused our research on: (1) extending the types of discoverable relationships between DC and OS event length; (2) Identifying two kinds of DC trends; DC trend that compose of DC and OS event and DC trend of only DC event; and (3) developing a novel trading strategy that optimises recommendation from individual DC thresholds.

The aim in (1) was to discover equations that express richer relationships between DC event length and OS event length using symbolic regression GP approach. Previous approaches discovered linear relationships. With our approach, we were able to discover more complex relationships that could be linear or non-linear tailored to a specific dataset. This resulted in an improvement in OS event length estimation accuracy and, consequently, led to improvements in DC trend reversal forecasting accuracy[1]. In addition, the improved forecasting model was embedded in a trading strategy and resulted in increased trading returns at reduced risk.

To further improve our DC trend reversal forecasting accuracy, in (2) we made the

[1] By adding estimated OS event length to the DC event length known at DCC point we can forecast trend reversal point

distinction between DC trends that end at the direction change confirmation point and others that continue beyond the said point. We extended the trend reversal forecasting algorithm by introducing a classification step that categorises DC trends into two kinds, 1) composed of DC and OS event, and 2) composed of only DC event. This knowledge significantly improved trend reversal estimation accuracy as OS event length estimation was calculated only when a trend is categorised to have one. Additionally, we dynamically selected threshold for sampling event series from a pool of thresholds by choosing the threshold with the best trend reversal forecasting accuracy in training.

In (3), we developed a GA based trading strategy that optimised trading actions and trend reversal point recommendations from multiple thresholds. Our approach was compared with results from a single threshold trading strategy. Results showed that further increase in profitability and reduction in risk is achieved by the multi-threshold trading.

7.1 Contributions

To tackle the problem of forecasting trend reversal in directional changes, we started by proposing a novel symbolic regression GP (SRGP) that estimated the length of an OS event based on the relationships between DC and OS event lengths. Our approach led to the improvement in OS event length estimation accuracy (Adegboye, Kampouridis and Johnson 2017). The estimation error of our SRGP was compared to those of other OS event length estimation algorithms in the literature (Glattfelder, Dupuis and Olsen 2011; Kampouridis and Otero 2017). The results showed that our SRGP statistically significantly outperformed them. Given the enhanced OS event length estimation to DC event length, we were also able to improve the accuracy at forecasting DC trend reversal. The algorithm was introduced as the forecasting engine in an existing multi-threshold trading strategy. The trading strategy recorded statistically significant profit in comparison to technical indicator based strategies, buy-and-hold and the original version of the strategy that used a different DC-based forecasting engine.

Despite the significant improvements by our approach for forecasting trend reversal, we identified two limitations which we addressed in our second contribution (Adegboye and Kampouridis 2021; Adegboye, Kampouridis and Otero 2021). The first limitation was the assumption that all DC trends are composed of DC and OS events. Empirical observation showed DC event series could have as little as 14.77% OS events even though this is threshold dependent. Therefore, OS event estimation should be done only when a DC trend is expected to have an OS event. The second limitation was the use of the same fixed-sized thresholds across the dataset which will not necessarily capture the most significant price events. We tackled these limitations by introducing a classification step before estimating OS event length. We estimated OS event length only when DC trends are classified to consist of DC and OS events. Otherwise, trends are considered to have only DC events. We then sampled events-series using tailored thresholds. We generated a pool of thresholds and evolved SRGPs for each threshold under perfect foresight. From the pool, we selected the threshold associated with an SRGP that had the least root mean squared error (RMSE) as the trading threshold.

Our results showed that this approach led to further statistically significant improvement to trend reversal forecasting accuracy after comparing with other known DC-based trend reversal forecasting algorithms. The results showed that improvement to trend reversal forecasting accuracy was achieved and confirmed the importance of carefully selecting thresholds and estimating OS event length only when it is known that an OS event exists.

To show that our improved trend reversal forecasting algorithm led to improved trading returns and reduced risk, we developed a new trading strategy. We tested 12 versions of the strategy embedded with different DC trend reversal forecasting algorithms. The first three forecasting algorithms assumed that all DC trends have corresponding OS event and estimated OS event length according to the approaches proposed by Glattfelder, Dupuis and Olsen (2011); Kampouridis and Otero (2017); Adegboye, Kampouridis and Johnson (2017). The second three set of forecasting algorithms introduced our classification step to the three aforementioned forecasting approaches and estimated OS event length only when a DC trend is classified to have an OS event. The third three set of forecasting algorithms probabilistically categorised DC trends and estimated OS event length according to the approaches proposed by Glattfelder, Dupuis and Olsen (2011); Kampouridis and Otero (2017); Adegboye, Kampouridis and Johnson (2017), respectively. The last three set of forecasting algorithms sampled event series using approaches proposed by Glattfelder, Dupuis and Olsen (2011); Kampouridis and Otero (2017); Adegboye, Kampouridis and Johnson (2017), then ignored the OS events, trading at the DCC point. We measured average return, Sharpe ratio and Maximum Drawdown (MDD). In general, the return of the variants that combined a classification model with an OS event length estimation model outperformed the others. Specifically, the Sharpe ratio result of our strategy (i.e., combined classification and SRGP models) statistically significantly outranked all the other strategies. Similarly, the Sharpe ratio result statistically outperformed seven technical analysis based strategies including buy-and-hold. The risk measure comparison (i.e., MDD) showed that our strategy approach outranked other strategies, but the performance was not statistically significant in comparison to other versions of the strategy that introduced classification. The results showed that finding richer relationships between DC and OS event lengths is beneficial for maximising returns and reducing risk.

Finally, as our third contribution in Chapter 6, we proposed a multi-threshold trading strategy that optimised trading recommendations from individual thresholds. The proposed trading strategy addresses the limitation of our single threshold based trading strategy, which could act on only a single type of event. The ability of a strategy to perceive and act on different types of events has the added advantage of making robust trading decision as the information from multiple thresholds is used to make decision. This was evident in the trading result reported where the multi-threshold trading strategy statistically significantly outperformed individual thresholds in both profit and risk measures.

7.2 Future Research

Although we were able to significantly improve trend reversal forecasting, future investigation is required in two areas for the approach to realise its full forecasting potential. The classification step consumes around 96% of the computational time that is required to create a forecasting model. It would be relevant to investigate alternative classification techniques that can generate classification models of comparable accuracy at reduced computational time. By doing this, the forecasting algorithm will be able to transition from an offline process to an online one. We also leave for future work the experimentation with Auto-WEKA in the multi-threaded mode for improvement in computational time spent on the classification step.

We successfully forecasted DC trend reversal using 10-minutes physical time Forex data. It is yet to be seen whether similar performance can be achieved in other markets (i.e., commodities, bond, indices and stocks, cryptocurrency) or not. It will therefore be interesting to investigate DC trend reversal forecasting in other markets using our approach. The data used for this work might be relatively old (i.e., between 5 and 7 years), it will be interesting to experiment with more recent data. Additionally, our work focused on trend reversal forecasting improvement, it will be worthwhile to evaluate the robustness of our approach at forecasting trending reversal in higher frequency data such as 1-minute physical time data and tick-data.

The final area of research that can be investigated further is the enhancement of our GA-based trading strategy framework. A potential limitation of our current framework is the selection of the best 5 thresholds as genes of our GA individuals. It could be the case that few or more genes are required. Therefore, further studies investigating variable-sized individuals in the GA population are warranted. This can be realised in such a way that chromosome size is one of the optimisation objectives. Additionally, in our current implementation, the total budget was used for every transaction. Future studies should address it by optimising minimum and maximum quantity to trade per transaction and dynamically adjust the value during trading using a reward/penalty system. Finally, future studies could focus on combining intrinsic and physical time scale approaches in a trading strategy to elucidate whether the combination could lead to improvement in trading returns and reduction in risk or not. It could be done by optimising recommendations from both technical indicators and multiple thresholds in the GA based trading strategy.

Bibliography

Abu-Mostafa, Y. S. and Atiya, A. F. (1996). Introduction to financial forecasting. *Applied Intelligence*, 6(3), pp. 205–213.

Achelis, S. B. (2001). *Technical Analysis from A to Z.* McGraw Hill New York.

Adegboye, A. and Kampouridis, M. (2021). Machine learning classification and regression models for predicting directional changes trend reversal in FX markets. *Expert Systems with Applications*, 173, p. 114645.

Adegboye, A., Kampouridis, M. and Johnson, C. G. (2017). Regression genetic programming for estimating trend end in foreign exchange market. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, pp. 1–8.

Adegboye, A., Kampouridis, M. and Otero, F. (2021). Improving trend reversal estimation in forex markets under a directional changes paradigm with classification algorithms. *International Journal of Intelligent Systems.*

Ali, S. and Smith, K. A. (2006). On learning algorithm selection for classification. *Applied Soft Computing*, 6(2), pp. 119–138.

Alkhamees, N. and Fasli, M. (2017a). A directional change based trading strategy with dynamic thresholds. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, pp. 283–292.

Alkhamees, N. and Fasli, M. (2017b). Event detection from time-series streams using directional change and dynamic thresholds. In *2017 IEEE International Conference on Big Data (Big Data)*, IEEE, pp. 1882–1891.

Aloud, M. (2016a). Directional-change event trading strategy: Profit-maximizing learning strategy. *The Seventh International Conference on Advanced Cognitive Technology and Applications*.

Aloud, M. (2017). Investment opportunities forecasting: a genetic programming-based dynamic portfolio trading system under a directional-change framework. *Journal of Computational Finance*, 22(1), pp. 1–35.

Aloud, M. and Fasli, M. (2013). The impact of strategies on the stylized facts in the fx market. Tech. rep., Citeseer.

Aloud, M., Tsang, E. and Olsen, R. (2014). Modeling the fx market traders' behavior: An agent-based approach. *Banking, Finance, and Accounting: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*, p. 350.

Aloud, M. E. (2016b). Profitability of directional change based trading strategies: The case of saudi stock market. *International Journal of Economics and Financial Issues*, 6(1).

Aloud, M. E. (2016c). Time series analysis indicators under directional changes: The case of saudi stock market. *International Journal of Economics and Financial Issues*, 6(1).

Angeline, P. J. (1996). An investigation into the sensitivity of genetic programming to the frequency of leaf selection during subtree crossover. In *Proceedings of the 1st annual conference on genetic programming*, MIT Press, pp. 21–29.

Antony, A. (2020). Behavioral finance and portfolio management: Review of theory and literature. *Journal of Public Affairs*, 20(2), p. e1996.

Ao, H. (2018). *A Directional Changes based study on stock market*. Ph.D. thesis, University of Essex.

Assaad, R. H. and Fayek, S. (2021). Predicting the price of crude oil and its fluctuations using computational econometrics: deep learning, lstm, and convolutional neural networks. *Econometric Research in Finance*, 6(2), pp. 119–137.

Atilgan, E. and Hu, J. (2018). First-principle-based computational doping of srtio$_3$ using combinatorial genetic algorithms. *Bulletin of Materials Science*, 41(1), pp. 1–9.

Attigeri, G. V., MM, M. P., Pai, R. M. and Nayak, A. (2015). Stock market prediction: A big data approach. In *TENCON 2015-2015 IEEE Region 10 Conference*, IEEE, pp. 1–5.

Azzini, A., da Costa Pereira, C. and Tettamanzi, A. G. (2010). Modeling turning points in financial markets with soft computing techniques. In *Natural Computing in Computational Finance*, Springer, pp. 147–167.

Bach, E. and Ghil, M. (2022). A multi-model ensemble kalman filter for data assimilation and forecasting. *arXiv preprint arXiv:220202272*.

Bakhach, A. (2018). *Developing trading strategies under the Directional Changes framework, with application in the FX market*. Ph.D. thesis, University of Essex.

Bakhach, A., Tsang, E. P. and Ng, W. L. (2015). Forecasting directional changes in financial markets. *Working Paper WP075-15 Centre for Computational Finance and Economic Agents (CCFEA), University of Essex, Tech Rep.*

Bakhach, A., Tsang, E. P. K. and Jalalian, H. (2016). Forecasting directional changes in FX markets. In *IEEE Symposium on Computational Intelligence for Financial Engineering & Economics (IEEE CIFEr'16)*, Athens Greece: IEEE, pp. 6–9.

Bakhach, A., Tsang, E., Ng, W. L. and Chinthalapati, V. R. (2016). Backlash agent: A trading strategy based on directional change. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, IEEE, pp. 1–9.

Bakhach, A., Chinthalapati, V., Tsang, E. and El Sayed, A. (2018). Intelligent dynamic backlash agent: A trading strategy based on the directional change framework. *Algorithms*, 11(11), p. 171.

Bakhach, A. M., Tsang, E. P. and Raju Chinthalapati, V. (2018). Tsfdc: A trading strategy based on forecasting directional change. *Intelligent Systems in Accounting, Finance and Management*, 25(3), pp. 105–123.

Balasubramaniam, K. (2021). Buying and selling in the forex market. Accessed: 2021-07-03.

Banzhaf, W., Nordin, P., Keller, R. E. and Francone, F. D. (1998). *Genetic programming: an introduction*, vol. 1. Morgan Kaufmann Publishers San Francisco.

Bao, D. and Yang, Z. (2008). Intelligent stock trading system by turning point confirming and probabilistic reasoning. *Expert Systems with Applications*, 34(1), pp. 620–627.

Basgalupp, M. P. et al. (2020). An extensive experimental evaluation of automated machine learning methods for recommending classification algorithms. *Evolutionary Intelligence*, pp. 1–20.

Beniwal, S. and Arora, J. (2012). Classification and feature selection techniques in data mining. *International journal of engineering research & technology (ijert)*, 1(6), pp. 1–6.

Bergstra, J., Bardenet, R., Bengio, Y. and Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.

Bernanke, B. S. (2017). Federal reserve policy in an international context. *IMF Economic Review*, 65(1), pp. 1–32.

Birattari, M., Yuan, Z., Balaprakash, P. and Stützle, T. (2010). F-race and iterated f-race: An overview. In *Experimental methods for the analysis of optimization algorithms*, Springer, pp. 311–336.

Bisig, T., Dupuis, A., Impagliazzo, V. and Olsen, R. (2012). The scale of market quakes. *Quantitative Finance*, 12(4), pp. 501–508.

Bordo, M. D. (2019). *The Operation and Demise of the Bretton Woods System, 1958–1971*. Yale University Press.

Braga, I., do Carmo, L. P., Benatti, C. C. and Monard, M. C. (2013). A note on parameter selection for support vector machines. In *Mexican International Conference on Artificial Intelligence*, Springer, pp. 233–244.

Brookhouse, J., Otero, F. E. and Kampouridis, M. (2014). Working with opencl to speed up a genetic programming financial forecasting algorithm: initial results.

In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pp. 1117–1124.

Caginalp, G. and Balevonich, D. (2003). A theoretical foundation for technical analysis. *Journal of Technical Analysis*, 59(5-22).

Carr, J. (2014). An introduction to genetic algorithms. *Senior Project*, 1(40), p. 7.

Cavalcante, R. C., Brasileiro, R. C., Souza, V. L., Nobrega, J. P. and Oliveira, A. L. (2016). Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 55, pp. 194–211.

Chang, J.-F. and Huang, Y.-M. (2014). Pso based time series models applied in exchange rate forecasting for business performance management. *Electronic Commerce Research*, 14(3), pp. 417–434.

Chen, J. and Tsang, E. P. (2020). *Detecting Regime Change in Computational Finance: Data Science, Machine Learning and Algorithmic Trading.* CRC Press.

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

Chen, W., Jiang, M., Zhang, W.-G. and Chen, Z. (2021). A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Information Sciences*, 556, pp. 67–94.

Chung, F.-L., Fu, T. C., Luk, R. and Ng, V. (2001). Flexible time series pattern matching based on perceptually important points. *International Joint Conference on Artificial Intelligence Workshop on Learning from Temporal and Spatial Data*, pp. 1–7.

Clegg, J., Walker, J. A. and Miller, J. F. (2007). A new crossover technique for cartesian genetic programming. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, ACM, pp. 1580–1587.

Colby, R. W. (2003). *The encyclopedia of technical market indicators*. McGraw-Hill.

Cont, R., Potters, M. and Bouchaud, J.-P. (1997). Scaling in stock market data: stable laws and beyond. In *Scale invariance and beyond*, Springer, pp. 75–85.

Critchley, H. D. and Garfinkel, S. N. (2018). The influence of physiological signals on cognition. *Current Opinion in Behavioral Sciences*, 19, pp. 13–18.

Dao, T. M., McGroarty, F. and Urquhart, A. (2019). The brexit vote and currency markets. *Journal of International Financial Markets, Institutions and Money*, 59, pp. 153–164.

Datta, R. K., Sajid, S. W., Moon, M. H. and Abedin, M. Z. (2021). Foreign currency exchange rate prediction using bidirectional long short term memory. In *The Big Data-Driven Digital Economy: Artificial and Computational Intelligence*, Springer, pp. 213–227.

Degiannakis, S. and Filis, G. (2019). Forecasting european economic policy uncertainty. *Scottish Journal of Political Economy*, 66(1), pp. 94–114.

Deng, S. and Sakurai, A. (2013). Foreign exchange trading rules using a single technical indicator from multiple timeframes. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, IEEE, pp. 207–212.

D'haeseleer, P. (1994). Context preserving crossover in genetic programming. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, IEEE, pp. 256–261.

Dixon, M. F., Halperin, I. and Bilokon, P. (2020). *Machine Learning in Finance.* Springer.

Dorigo, M. and Stützle, T. (2004). *The Ant Colony Optimization Metaheuristic.* pp. 25–64.

Du Plessis, A. W. (2012). *The effectiveness of a technical analysis strategy versus a buy-and-hold strategy on the FTSE/JSE top 40 index shares of the JSE Ltd: the case of the Moving Average Convergence Divergence Indicator.* Ph.D. thesis, University of Johannesburg.

Dymova, L., Sevastjanov, P. and Kaczmarek, K. (2016). A forex trading expert system based on a new approach to the rule-base evidential reasoning. *Expert Systems with Applications*, 51, pp. 1–13.

Edwards, R. D., Magee, J. and Bassetti, W. (2012). *Technical analysis of stock trends.* CRC Press.

Eom, C., Choi, S., Oh, G. and Jung, W.-S. (2008). Hurst exponent and prediction based on weak-form efficient market hypothesis of stock markets. *Physica A: Statistical Mechanics and its Applications*, 387(18), pp. 4630–4636.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2), pp. 383–417.

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

Fernandez-Rodrıguez, F., Gonzalez-Martel, C. and 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), pp. 89–94.

Fernando, J. (2021). Gross domestic product (gdp). Accessed: 2021-07-04.

Feurer, M., Springenberg, J. and Hutter, F. (2015). Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29.

Feurer, M. et al. (2015). Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, pp. 2962–2970.

Flanagan, R. and Lacasa, L. (2016). Irreversibility of financial time series: a graph-theoretical approach. *Physics Letters A*, 380(20), pp. 1689–1697.

Fogel, L. J., Owens, A. J. and Walsh, M. J. (1966). Intelligent decision making through a simulation of evolution. *Behavioral science*, 11(4), pp. 253–272.

Folger, J. (2020 ([ONLINE]: accessed December 26, 2020)). Should you trade forex or stocks? https://www.investopedia.com/articles/forex/11/forex-or-stocks.asp.

Fontanills, G. A. and Gentile, T. (2002). *The stock market course*, vol. 117. John Wiley & Sons.

Gabrielli, D. and Valente, C. (2011). Which random walks are cyclic? *arXiv preprint arXiv:11075336*.

Garber, P. M. (2007). 9. the collapse of the bretton woods fixed exchange rate system. In *A retrospective on the Bretton Woods system*, University of Chicago Press, pp. 461–494.

Giannellis, N. and Koukouritakis, M. (2013). Exchange rate misalignment and inflation rate persistence: Evidence from latin american countries. *International Review of Economics & Finance*, 25, pp. 202–218.

Glattfelder, J., Dupuis, A. and Olsen, R. (2008). An extensive set of scaling laws and the FX coastline. *Centre for Computational Finance and Economics Agents, Working Paper Series WP025-08*.

Glattfelder, J., Dupuis, A. and Olsen, R. (2011). Patterns in high-frequency fx data: discovery of 12 empirical scaling laws. *Quantitative Finance*, 11(4), pp. 599–614.

Grossman, S. J. and Stiglitz, J. E. (1980). On the impossibility of informationally efficient markets. *The American economic review*, 70(3), pp. 393–408.

Guillaume, D. M. et al. (1997). From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets. *Finance and stochastics*, 1(2), pp. 95–129.

Gypteau, J., Otero, F. E. and Kampouridis, M. (2015). Generating directional change based trading strategies with genetic programming. In *European Conference on the Applications of Evolutionary Computation*, Springer, pp. 267–278.

Habib, M. M., Mileva, E. and Stracca, L. (2017). The real exchange rate and economic growth: Revisiting the case using external instruments. *Journal of International Money and Finance*, 73, pp. 386–398.

Hall, M. et al. (2009). The weka data mining software: An update. *SIGKDD Explor Newsl*, 11(1), pp. 10–18.

Harries, K. and Smith, P. (1997). Exploring alternative operators and search strategies in genetic programming. *Genetic Programming*, 97, pp. 147–155.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). Unsupervised learning. In *The elements of statistical learning*, Springer, pp. 485–585.

Hayes, A. (2021). What is a trading strategy. Accessed: 2021-07-03.

Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1), pp. 66–73.

Holmes, G., Donkin, A. and Witten, I. H. (1994). Weka: A machine learning workbench. In *Proceedings of ANZIIS'94-Australian New Zealnd Intelligent Information Systems Conference*, IEEE, pp. 357–361.

Hongguang, L. and Ping, J. (2015). Generating intraday trading rules on index future markets using genetic programming. *International Journal of Trade, Economics and Finance*, 6(2), p. 112.

Hu, Y. et al. (2015). Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Applied Soft Computing*, 36, pp. 534–551.

Huang, J., Chai, J. and Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, 14, pp. 1–24.

Igwe, I. O. (2018). History of the international economy: The bretton woods system and its impact on the economic development of developing countries. *Athens JL*, 4, p. 105.

Islam, M. et al. (2020). A review on recent advancements in forex currency prediction. *Algorithms*, 13(8), p. 186.

Jensen, M. C. (1978). Some anomalous evidence regarding market efficiency. *Journal of financial economics*, 6(2-3), pp. 95–101.

Jin, H., Song, Q. and Hu, X. (2019). Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1946–1956.

Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3), pp. 401–422.

Kablan, A. and Ng, W. L. (2011). Intraday high-frequency fx trading with adaptive neuro-fuzzy inference systems. *International Journal of Financial Markets and Derivatives*, 2(1-2), pp. 68–87.

Kaltwasser, P. R. (2010). Uncertainty about fundamentals and herding behavior in the forex market. *Physica A: Statistical Mechanics and its Applications*, 389(6), pp. 1215–1222.

Kampouridis, M., Adegboye, A. and Johnson, C. (2017). Evolving directional changes trading strategies with a new event-based indicator. In *Asia-Pacific Conference on Simulated Evolution and Learning*, Springer, pp. 727–738.

Kampouridis, M. and Otero, F. E. (2017). Evolving trading strategies using directional changes. *Expert Systems with Applications*, 73, pp. 145–160.

Kamruzzaman, J., Sarker, R. A. and Ahmad, I. (2003). Svm based models for predicting foreign currency exchange rates. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, IEEE, pp. 557–560.

Karegowda, A. G., Manjunath, A. and Jayaram, M. (2010). Comparative study of attribute selection using gain ratio and correlation based feature selection. *International Journal of Information Technology and Knowledge Management*, 2(2), pp. 271–277.

Kenton, W. (2021). Balance of trade. Accessed: 2021-07-04.

Khanmohammadi, S. and Rezaeiahari, M. (2014). Ahp based classification algorithm selection for clinical decision support system development. *Procedia Computer Science*, 36, pp. 328–334.

Kinnear, K. E. (1994). Fitness landscapes and difficulty in genetic programming. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, IEEE, pp. 142–147.

Komer, B., Bergstra, J. and Eliasmith, C. (2014). Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In *ICML workshop on AutoML*, pp. 2825–2830.

Korczak, J., Hernes, M. and Bac, M. (2016). Fundamental analysis in the multi-agent trading system. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, IEEE, pp. 1169–1174.

Kotsiantis, S. B., Zaharakis, I., Pintelas, P. et al. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1), pp. 3–24.

Kotthoff, L., Thornton, C. and Hutter, F. (2017). User guide for auto-weka version 2.6. *Dept Comput Sci, Univ British Columbia, BETA lab, Vancouver, BC, Canada, Tech Rep*, 2.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*, vol. 1. MIT press.

Krol, R. (1992). Trends, random walks and persistence: an empirical study of disaggregated us industrial production. *The Review of Economics and Statistics*, pp. 154–159.

Kyriazis, N. A. (2019). A survey on efficiency and profitable trading opportunities in cryptocurrency markets. *Journal of Risk and Financial Management*, 12(2), p. 67.

Langdon, W. B. (1996). Evolution of genetic programming populations. *University College London Technical Report RN/96/125.*

Langdon, W. B. (1998). Genetic programming—computers using "natural selection" to generate programs. In *Genetic Programming and Data Structures*, Springer, pp. 9–42.

Langdon, W. B. (2000). Size fair and homologous tree crossovers for tree genetic programming. *Genetic programming and evolvable machines*, 1(1-2), pp. 95–119.

Levy, R. A. (1967). Random walks: Reality or myth. *Financial Analysts Journal*, 23(6), pp. 69–77.

Liu, Y., Gong, D., Sun, J. and Jin, Y. (2017). A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Transactions on Cybernetics*, 47(9), pp. 2689–2702.

Lo, A. W. and MacKinlay, A. C. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *The review of financial studies*, 1(1), pp. 41–66.

Lo, A. W., Mamaysky, H. and Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The journal of finance*, 55(4), pp. 1705–1770.

López-Ibánez, M., Dubois-Lacoste, J., Stützle, T. and Birattari, M. (2011). The irace package, iterated race for automatic algorithm configuration. Tech. rep., Citeseer.

Lui, Y.-H. and Mole, D. (1998). The use of fundamental and technical analyses by foreign exchange dealers: Hong kong evidence. *Journal of International money and Finance*, 17(3), pp. 535–545.

Ma, J., Xiong, X., He, F. and Zhang, W. (2017). Volatility measurement with directional change in chinese stock market: Statistical property and investment strategy. *Physica A: Statistical Mechanics and its Applications*, 471, pp. 169–180.

Macedo, L. L., Godinho, P. and Alves, M. J. (2020). A comparative study of technical trading strategies using a genetic algorithm. *Computational Economics*, 55(1), pp. 349–381.

Maher, M. and Sakr, S. (2019). Smartml: A meta learning-based framework for automated selection and hyperparameter tuning for machine learning algorithms. In *EDBT: 22nd International Conference on Extending Database Technology*.

Mahmoud, H. F. et al. (2021). Parametric versus semi and nonparametric regression models. *International Journal of Statistics and Probability*, 10(2), pp. 1–90.

Malkiel, B. G. (1999). *A random walk down Wall Street: including a life-cycle guide to personal investing*. WW Norton & Company.

Mandelbrot, B. (1967). The variation of some other speculative prices. *The Journal of Business*, 40(4), pp. 393–413.

Mandelbrot, B. and Taylor, H. M. (1967). On the distribution of stock price differences. *Operations research*, 15(6), pp. 1057–1062.

Maxwell, S. and Koza, J. R. (1996). Why might some problems be difficult for genetic. *Late Breaking Papers at the Genetic Programming 1996*, pp. 125–128.

McCullagh, P. (2002). What is a statistical model? *The Annals of Statistics*, 30(5), pp. 1225–1310.

Metaxiotis, K. and Liagkouras, K. (2012). Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. *Expert Systems with Applications*, 39(14), pp. 11685–11698.

Michie, D., Spiegelhalter, D. J. and Taylor, C. C. (1994). Machine learning, neural and statistical classification. *Paramount Publishing International*.

Miller, J. F. (1999). An empirical study of the efficiency of learning boolean functions using a cartesian genetic programming approach. In *Proceedings of the genetic and evolutionary computation conference*, vol. 2, pp. 1135–1142.

Miller, J. F. (2019). Cartesian genetic programming: its status and future. *Genetic Programming and Evolvable Machines*, pp. 1–40.

Mohr, F., Wever, M. and Hüllermeier, E. (2018). Ml-plan: Automated machine learning via hierarchical planning. *Machine Learning*, 107(8), pp. 1495–1515.

mortgage strategy (2019). Historical interest rates in the uk, 1979-2017: rise and fal. Accessed: 2021-07-02.

Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin.

Neely, C., Weller, P. and Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? a genetic programming approach. *Journal of financial and Quantitative Analysis*, 32(04), pp. 405–426.

Neely, C. J. (2003). Risk-adjusted, ex ante, optimal technical trading rules in equity markets. *International Review of Economics & Finance*, 12(1), pp. 69–87.

Nordin, P. (1994). A compiling genetic programming system that directly manipulates the machine code. *Advances in genetic programming*, 1, pp. 311–331.

Nti, I. K., Adekoya, A. F. and Weyori, B. A. (2019). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, pp. 1–51.

O'neill, M., Ryan, C., Keijzer, M. and Cattolico, M. (2003). Crossover in grammatical evolution. *Genetic programming and evolvable machines*, 4(1), pp. 67–93.

Ong, B. W. and Schroder, J. B. (2020). Applications of time parallelization. *Computing and Visualization in Science*, 23(1), pp. 1–15.

Palsma, J. and Adegboye, A. (2019). Optimising directional changes trading strategies with different algorithms. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, pp. 3333–3340.

Peng, C.-C., Wang, J.-G. and Yeh, C.-W. (2020). Survey of forex price forecasting approaches based on artificial neural networks. In *2020 IEEE 2nd Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS)*, IEEE, pp. 175–178.

Pesaran, M. and Pick, A. (2008). Forecasting random walks under drift instability. Tech. rep., CESifo Group Munich.

Petropoulos, A., Chatzis, S. P., Siakoulis, V. and Vlachogiannakis, N. (2017). A stacked generalization system for automated forex portfolio trading. *Expert Systems with Applications*, 90, pp. 290–302.

Petrov, V., Golub, A. and Olsen, R. (2019a). Instantaneous volatility seasonality of high-frequency markets in directional-change intrinsic time. *Journal of Risk and Financial Management*, 12(2), p. 54.

Petrov, V., Golub, A. and Olsen, R. (2020). Agent-based modelling in directional-change intrinsic time. *Quantitative Finance*, 20(3), pp. 463–482.

Petrov, V., Golub, A. and Olsen, R. B. (2019b). Intrinsic time directional-change methodology in higher dimensions. *Available at SSRN 3440628*.

Petrusheva, N. and Jordanoski, I. (2016). Comparative analysis between the fundamental and technical analysis of stocks. *Journal of Process Management New Technologies*, 4(2), pp. 26–31.

Picasso, A., Merello, S., Ma, Y., Oneto, L. and Cambria, E. (2019). Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications*, 135, pp. 60–70.

Plastun, A. (2017). Behavioral finance market hypotheses. *Financial Behavior: Players, Services, Products, and Markets*, p. 439.

Poli, R., Langdon, W. B., McPhee, N. F. and Koza, J. R. (2008). *A field guide to genetic programming*. Lulu. com.

Raftopoulos, S. (2003). The zigzag trend indicator. *TECHNICAL ANALYSIS OF STOCKS AND COMMODITIES-MAGAZINE EDITION-*, 21(11), pp. 26–33.

Rechenberg, I. (1973). Evolution strategy: Optimization of technical systems by means of biological evolution. *Fromman-Holzboog, Stuttgart*, 104, pp. 15–16.

Regnault, J. (1863). *Calcul des chances et philosophie de la bourse*. librairie Castel.

Reif, M., Shafait, F., Goldstein, M., Breuel, T. and Dengel, A. (2014). Automatic classifier selection for non-experts. *Pattern Analysis and Applications*, 17(1), pp. 83–96.

Remias, R. (2021). *President Trump's Tweets and their Effect on the Stock Market: The Relationship Between Social Media, Politics, and Emotional Economic Decision-Making*. Ph.D. thesis, Wittenberg University.

Rosca, J. P. (1995). Genetic programming exploratory power and the discovery of functions. In *Evolutionary Programming*, Citeseer, pp. 719–736.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:160904747*.

Sabharwal, A., Samulowitz, H. and Tesauro, G. (2016). Selecting near-optimal learners via incremental data allocation. In *AAAI*, pp. 2007–2015.

Safe, M., Carballido, J., Ponzoni, I. and Brignole, N. (2004). On stopping criteria for genetic algorithms. In *Brazilian Symposium on Artificial Intelligence*, Springer, pp. 405–413.

Sathya, R. and Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2), pp. 34–38.

Schabacker, R. (2005). *Technical Analysis and Stock Market Profits*. Harriman House Limited.

Schneeweis, T. (1983). Determinats of profitability: An international perspective. *Management International Review*, pp. 15–21.

Schoenauer, M. et al. (1996). Evolutionary identification of macro-mechanical models. *Advances in genetic programming*, 2, pp. 467–488.

Seth, S. (2021). Technical analysis strategies for beginners. Accessed: 2021-07-11.

Sewell, M. (2011). Characterization of financial time series. Research Note, RN/11/01, UCL Department of Computer Science.

Sezer, O. B., Gudelek, M. U. and Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90, p. 106181.

Shiryaev, A., Xu, Z. and Zhou, X. Y. (2008). Thou shalt buy and hold. *Quantitative finance*, 8(8), pp. 765–776.

Sidehabi, S. W., Tandungan, S. et al. (2016). Statistical and machine learning approach in forex prediction based on empirical data. In *2016 International Conference on Computational Intelligence and Cybernetics*, IEEE, pp. 63–68.

Sivanandam, S. and Deepa, S. (2008). Genetic algorithms. In *Introduction to genetic algorithms*, Springer, pp. 15–37.

Smith, M. R., Mitchell, L., Giraud-Carrier, C. and Martinez, T. (2014). Recommending learning algorithms and their associated hyperparameters. *arXiv preprint arXiv:14071890*.

Sparks, E. R. et al. (2015). Automating model search for large scale machine learning. In *Proceedings of the Sixth ACM Symposium on Cloud Computing*, ACM, pp. 368–380.

Stasinakis, C. and Sermpinis, G. (2014). Financial forecasting and trading strategies: a survey. In *Computational Intelligence Techniques for Trading and Investment*, Routledge, pp. 42–60.

Stawiarski, B. (2015). Selected techniques of detecting structural breaks in financial volatility. *e-Finanse: Financial Internet Quarterly*, 11(1), pp. 32–43.

Storn, R. and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), pp. 341–359.

Thornton, C., Hutter, F., Hoos, H. H. and Leyton-Brown, K. (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 847–855.

Tighe, D., Lewis-Morris, T. and Freitas, A. (2019). Machine learning methods applied to audit of surgical outcomes after treatment for cancer of the head and neck. *British Journal of Oral and Maxillofacial Surgery*, 57(8), pp. 771–777.

Tsang, E. and Chen, J. (2018). Regime change detection using directional change indicators in the foreign exchange market to chart brexit. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(3), pp. 185–193.

Tsang, E. P., Tao, R. and Ma, S. (2015). Profiling financial market dynamics under directional changes. *Quantitative finance, http://www tandfonline com/doi/abs/101080/146976882016*, 1164887.

Tsang, E. P. et al. (2000). Eddie in financial decision making. *Journal of Management and Economics*, 4(4), pp. 1–13.

Tsang, E. P., Tao, R., Serguieva, A. and Ma, S. (2017). Profiling high-frequency equity price movements in directional changes. *Quantitative finance*, 17(2), pp. 217–225.

Tun, Z. T. (2020). Pros & cons of a passive buy and hold strategy.

Wah, Y. B., Ibrahim, N., Hamid, H. A., Abdul-Rahman, S. and Fong, S. (2018). Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science & Technology*, 26(1).

Walczak, S. (2001). An empirical analysis of data requirements for financial forecasting with neural networks. *Journal of management information systems*, 17(4), pp. 203–222.

Wang, J.-Z., Wang, J.-J., Zhang, Z.-G. and Guo, S.-P. (2011). Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*, 38(11), pp. 14346–14355.

Wang, X. and Wang, Z. (2021). Analysis of the response of exchange rates to specific fomc announcements using high-frequency data. *Borsa Istanbul Review*.

Wang, Y., Wagner, N. and Rondinelli, J. M. (2019). Symbolic regression in materials science. *MRS Communications*, 9(3), pp. 793–805.

Wolpert, D. H., Macready, W. G. et al. (1995). No free lunch theorems for search. Tech. rep., Technical Report SFI-TR-95-02-010, Santa Fe Institute.

Wong, M. F., Fai, C. K., Yee, Y. C. and Cheng, L. S. (2019). Macroeconomic policy and exchange rate impacts on the foreign direct investment in asean economies. *International Journal of Economic Policy in Emerging Economies*, 12(1), pp. 1–10.

Woody, K. E. (2020). The new insider trading. *Ariz St LJ*, 52, p. 594.

Wooldridge, P. D. (2019). Fx and otc derivatives markets through the lens of the triennial survey. *BIS Quarterly Review, December*.

Xiao, C. (2015). *Using Machine Learning for Exploratory Data Analysis and Predictive Models on Large Datasets*. Master's thesis, University of Stavanger, Norway.

Yam, S., Yung, S. and Zhou, W. (2009). Two rationales behind the 'buy-and-hold or sell-at-once'strategy. *Journal of Applied Probability*, 46(3), pp. 651–668.

Ye, A., Chinthalapati, V. R., Serguieva, A. and Tsang, E. (2017). Developing sustainable trading strategies using directional changes with high frequency data. In *2017 IEEE International Conference on Big Data (Big Data)*, IEEE, pp. 4265–4271.

Yıldırım, D. C., Toroslu, I. H. and Fiore, U. (2021). Forecasting directional movement of forex data using lstm with technical and macroeconomic indicators. *Financial Innovation*, 7(1), pp. 1–36.

Yildirim, H. (2017). Behavioral finance or efficient market hypothesis? *International Journal of Academic Value Studies*, 3, pp. 151–158.

Yin, J., Si, Y.-W. and Gong, Z. (2011). Financial time series segmentation based on turning points. In *Proceedings 2011 International Conference on System Science and Engineering*, IEEE, pp. 394–399.

Yu, L., Wang, S. and Lai, K. K. (2007). *Foreign-exchange-rate forecasting with artificial neural networks*, vol. 107. Springer Science & Business Media.

Zafar, S. T. (2012). A systematic study to test the efficient market hypothesis on bse listed companies before recession. *International Journal of Management and Social Sciences Research*, 1(1), pp. 37–48.

Žegklitz, J. and Pošík, P. (2020). Benchmarking state-of-the-art symbolic regression algorithms. *Genetic Programming and Evolvable Machines*, pp. 1–29.