Deep Reinforcement Learning for Trading Strategy Development on High-Frequency Currency Data Using Directional Changes Sampling

George Rayment

A thesis submitted for the degree of

Doctor of Philosophy in Computational Finance

School of Computer Science and Electronic Engineering

University of Essex

November 24, 2025

Abstract

High-frequency trading in the foreign exchange market presents unique challenges, requiring sophisticated techniques to address its complexities. This thesis investigates the application of deep reinforcement learning to algorithmically trade high-frequency currency data. Traditional trading algorithms typically use fixed interval sampling in both manual and automated trading strategies. While effective for less noisy price movements and longer trade durations, this approach can miss significant price shifts in high-frequency scenarios which limits profitability. To overcome these limitations, this thesis employs the directional change (DC) sampling paradigm, which captures significant price movements more effectively. By combining DC sampled data and deep reinforcement learning to train trading agents, the research explores whether this approach outperforms traditional fixed interval methods when trading at high frequencies. The initial investigation develops the Filtered Deep Reinforcement Learning (FDRL) trading framework, using deep reinforcement learning to create semi-autonomous trading agents. Results show that FDRL is effective at fixed transaction costs but requires rule-based interventions to manage trades. To enhance autonomy, the Positionally Aware Deep Reinforcement Learning (PADRL) framework is introduced, incorporating real-time positional awareness to eliminate the need for rule-based filters, further improving performance. The final contribution of the Spread Aware Deep Reinforcement Learning (SADRL) framework, refines the PADRL approach by using the bid-ask spread as opposed to fixed transaction costs, making the strategy more realistic and applicable to real trading environments. Each algorithm iteration demonstrates improved

performance over traditional benchmarks like buy-and-hold and technical analysis. Financial metrics including Total Return, Maximum Drawdown and Calmar Ratio validate the superior performance of these deep reinforcement learning-based strategies, demonstrating their potential for advancing high-frequency trading in the foreign exchange market.

Acknowledgements

I would firstly like to extend my deepest thanks to my supervisor, Dr. Michael Kampouridis, whose patience and expertise turned what is often regarded as a daunting challenge into an enjoyable and highly rewarding experience. I would also like to thank my family, who have supported me throughout the whole journey with their unwavering encouragement, understanding and belief in my abilities.

Contents

List of Tables

List of Algorithms

List of Figures

Chapter 1

Introduction

The foreign exchange (FX) market operates as a decentralised and continuously active global marketplace for the exchange of currencies. The market opens at 10pm GMT on Sunday at the start of the Australian session and closes at 10pm GMT on Friday at the end of the US session. In 2019 it was estimated that the currency market hit a total value of 2.4 quadrillion and had a daily trading volume of 6.6 trillion USD [1]. As well as the general necessity to exchange currencies to purchase goods and services in native currencies, the potential for financial gain is an attraction for many traders and has catalysed the development of various different trading strategies, designed to capitalise on the fluctuations in the value of one currency against another, commonly referred to as a currency pair. The strategies developed by traders rely on an understanding of the price behaviour and market mechanics to anticipate future changes in price [2]. The ability to accurately predict price changes is a fundamental component of profitable trading, but an accompanying strategy that makes use of price predictions is just as important to translate accurate financial forecasting into actual returns.

Financial forecasting consists of two crucial elements: data curation and data modelling. The curation process involves the extraction and computation of financial indicators that numerically represent the movements of currency prices. Effective forecasting models rely heavily on the quality of the data they are trained on, so the effective curation of the data provides many benefits downstream. Once

a trader is in possession of a reliable forecasting model, they then have the choice of developing a trading strategy. Trading strategies are employed to make trading decisions based on the outputs of financial models that have been trained by the curated data [3], they are often considered as a rule-based wrapper around the forecasting model. The deep reinforcement learning techniques used in this thesis take this a step further by integrating both the financial forecasting and trading strategy into a single decision making agent. A financial forecasting model and an algorithmic trading strategy, whether separate or together, form the foundations of algorithmic trading frameworks. Algorithmic trading frameworks offer opportunities to refine both processes through advanced techniques, particularly when working with high-frequency FX data.

High-frequency trading (HFT) is especially attractive due to the increased granularity of price movements observed over shorter time frames. This concept is often described through the metaphor of a "larger coastline", where more frequent, smaller changes in price create a longer and more detailed price curve, offering traders a greater number of opportunities to capitalise on price changes. As a result, HFT presents significant profit potential. The increase in trading opportunity does not come without added challenges though, noisier price signals necessitate the use of sophisticated techniques to mitigate uncertainty and improve the accuracy of the trading decisions.

FX data, at its most fundamental level is composed of tick prices, these are records of executed bid and ask prices over time for a specific currency pair. These data points are irregularly spaced in time, which typically requires sampling before any meaningful financial indicators can be extracted. Fixed interval sampling, the most common approach, captures prices at regular time intervals, such as every 10 minutes for intra-day analysis. Non-linear and non-stationary characteristics of FX data mean that fixed interval sampling can miss significant price movements that occur between intervals [4]. To address these limitations, event-driven sampling methods, such as directional changes (DC) sampling, offer a more dynamic

alternative. DC sampling focuses on identifying key price movements by setting a threshold (θ), representing a significant percentage change in the market price. When price movements exceed this threshold, they are recorded as alternating upward or downward changes, followed by an "overshoot" event, capturing the extent to which the price continues in this direction after triggering a new sample [5].

The application of directional change sampling complements machine learning (ML) techniques by enhancing the clarity and relevance of price movements, focusing on events of meaningful magnitude rather than arbitrary time-based intervals. In the context of ML-driven trading, clean and information-rich data is vital for the successful application of these algorithms. Reinforcement learning (RL), a branch of ML, frames the trading problem as one in which an agent interacts with an environment, making decisions that impact future states while receiving rewards based on the outcomes of its actions. Deep reinforcement learning (DRL), which uses deep neural networks, extends this approach by enabling the agent to learn more complex policies for navigating an environment [6, 7]. Through the integration of DC sampling and ML techniques, particularly RL and DRL, this work aims to develop advanced high-frequency trading strategies capable of exploiting the intricate dynamics of the FX market.

1.1 Motivation

The ability to accurately predict price movements in financial markets offers a compelling financial incentive. This financial incentive attracts many participants and as a result builds a complex dynamic system that often demonstrates patterns that academic researchers and hedge funds alike have been trying to decode for decades. A key challenge in algorithmic trading is finding effective strategies that maximise profit while simultaneously keeping risk to a minimum. As advancements in machine learning (ML) and, more specifically, deep learning continue at a rapid pace, these techniques are being increasingly applied to a variety of domains, including high-frequency trading (HFT). Recent breakthroughs in deep learning,

particularly in the fields of natural language processing (NLP) and image generation, have demonstrated the ability of these methods to handle complex, sequential data. Many parallels can be drawn between these datasets and the sequential and interdependent nature of financial price data. Deep reinforcement learning has therefore emerged as a powerful tool for training agents to take actions within an environment that represented a market of buyers and sellers. Despite its growing success in other fields, the application of DRL to HFT remains relatively under-explored, with a limited body of literature addressing its potential in this space.

A key principle in machine learning is that the quality of a model's output is directly influenced by the quality of its input data. This principle is particularly relevant in the context of high-frequency trading, where the data is inherently noisy and often obscures the underlying trends. The noisy nature of HFT data presents a unique challenge, and addressing this challenge is a central motivation for this thesis. By applying the event-based sampling algorithm of directional changes (DC) sampling, this work aims to investigate how these methods enhance the performance of trading agents in HFT environments by bringing greater clarity to the movements of price data. Unlike more conventional time-based sampling methods, DC sampling focuses on significant price movements, which can provide machine learning models with cleaner and more meaningful data, ultimately improving model performance.

This thesis explores the intersection of machine learning, specifically deep reinforcement learning and event-based sampling, using the directional change (DC) approach to develop profitable trading systems on high frequency FX data. The work seeks to demonstrate that deep reinforcement learning algorithms, when combined with data prepared to maximise clarity, can be successfully applied to high-frequency financial data. In doing so, this research contributes to the growing understanding of how advanced machine learning techniques can be extended to the fast-paced and volatile environment of high-frequency trading.

1.2 Thesis Overview

This thesis is structured into seven chapters. This first chapter provides an introduction to the thesis, laying the foundation for the topics and themes explored in the subsequent chapters. Chapter 2 provides the fundamentals of financial forecasting, outlining the essential concepts and techniques upon which this research was formulated. Chapter 3 outlines the essential background knowledge of the machine learning techniques used in this thesis as well as reviewing the literature relevant to this work. Chapters 4, 5 and 6 provide the contributions of this thesis to the literature, detailing the development of the FDRL, PADRL and SADRL high-frequency trading algorithms. Chapter 7 is the final chapter, in which the main findings are summarised, conclusions are drawn and directions of future research to further advance the work presented in this thesis are suggested.

1.3 Publications

The following publications listed below contain the peer reviewed content from which this thesis was written:

- Preliminary work for the following thesis was conducted in [8].
- Chapter 4 is based on the content from [9] which includes the FDRL framework.
- Chapter 5 is based on the content from [10] which includes the PADRL framework.
- Chapter 6 forms the foundation for a journal paper currently under development.

Chapter 2

Financial Forecasting Fundamentals

This chapter establishes the theoretical framework that underpins the remainder of this thesis by providing context for the research conducted. A ground up approach is taken by first investigating the fundamental concepts of financial forecasting, before moving onto the details of sampling paradigms. The sampling paradigms discussed are centred around fixed interval and DC sampling, with other paradigms also being referenced to give further context to both. Once the fundamentals of financial forecasting and price sampling have been covered, the topic of technical analysis is covered, providing the equations used to calculate both the fixed interval and DC based indicators.

2.1 Financial Forecasting Concepts

Financial forecasting refers to the concept of predicting future financial outcomes for a financial instrument using historical data [11]. The scope of this definition is broad and could refer to anything from inflation rates [12,13] to stock prices [14,15] to monthly personal spending [16,17]. In any case, the concept of using all available and relevant information at a given moment in time to inform future predictions about a financial variable remains the same. The particular variety of financial

forecasting used in this thesis is specific to the financial markets and therefore refers to the prediction of currency prices in relation to another currency over a period of time.

Predicting currency prices is not a trivial task and often the first question raised by academics is if it is even possible. This has always been quite a prominent question within the literature, so much so that it has given rise to the concept of the Efficient Market Hypothesis (EMH), introduced by Fama in 1970 [18]. The Efficient Market Hypothesis states that 'A market in which prices always "fully reflect" available information is called "efficient"'. This hypothesis therefore claims that, assuming symmetric information across all interested parties, the price would always be a perfect representation of an asset at that point in time and any profit made from the market would be short-lived and the result of guesswork. It has been debated however how theoretical this hypothesis is as many have provided reasons against it, namely imperfect information efficiency [19] and imperfect competition [20], with other work suggesting the EMH is subject to certain conditions such as specific time frames [21]. [22] investigates the history of the EMH and concludes that it exists in many forms and to make such a definitive statement goes against the very nature of economics, a social science full of nuance and edge cases. The conclusion therefore gives rise to the existence of different gradings of the EMH, commonly referred to in three denominations of the EMH [23]. The 'Weak form EMH' [24] suggests that the price today reflects all past data and technical analysis is therefore unhelpful to investors. The 'Semi-strong form EMH' [25] claims that public information is already accounted for in the price of an asset, and therefore technical and fundamental analysis of an asset provides no edge to the investor and only insider information is effective for forecasting price movements. The 'Strong form EMH' [26] states that all information is represented by the price of an asset and no information would give an investor an advantage.

The random walk theory is a consequence of the EMH that claims that if markets are always efficient then all price movements should follow a random

walk [27]. If this theory were to be true, it would again render techniques like technical analysis ineffective for building reliable trading strategies. However, as investigated later in this chapter, several studies have demonstrated the value of technical analysis in predicting market movements. Lo and MacKinlay [28] for instance, provided compelling evidence that historical prices could be used to forecast future profits, challenging the long-held belief in the random walk theory. The goal therefore when building a trading strategy is to interpret information in a way that an advantage over other traders can be obtained, this advantage can then be translated into long term positive returns. One technique for gaining this advantage is the use of machine learning, which has provided a means to identify patterns in data considered invisible to humans. Before discussing the use of machine learning in Section 3.3, sampling techniques and technical analysis are first discussed in order to investigate how financial information is prepared before it is used to identify patterns.

2.2 Sampling Algorithms

The following section outlines the differences between fixed interval sampling and event-based sampling methods. Directional change sampling is the event-based sampling method used in this thesis and is therefore analysed in more depth than other available event-based methods. The purpose of this section is to demonstrate the differences between fixed interval sampling and DC sampling with the intention of providing the required background knowledge for all references to DC sampling within the remainder of this thesis. The literature review of the applications of DC sampling will be covered in Section 3.3 along with applications of machine learning to trading.

Tick Data Before developing an understanding of price sampling, it must first be understood that the raw data that is being sampled. Each transaction is represented as a set of an ask price, a bid price and a timestamp. The bid price

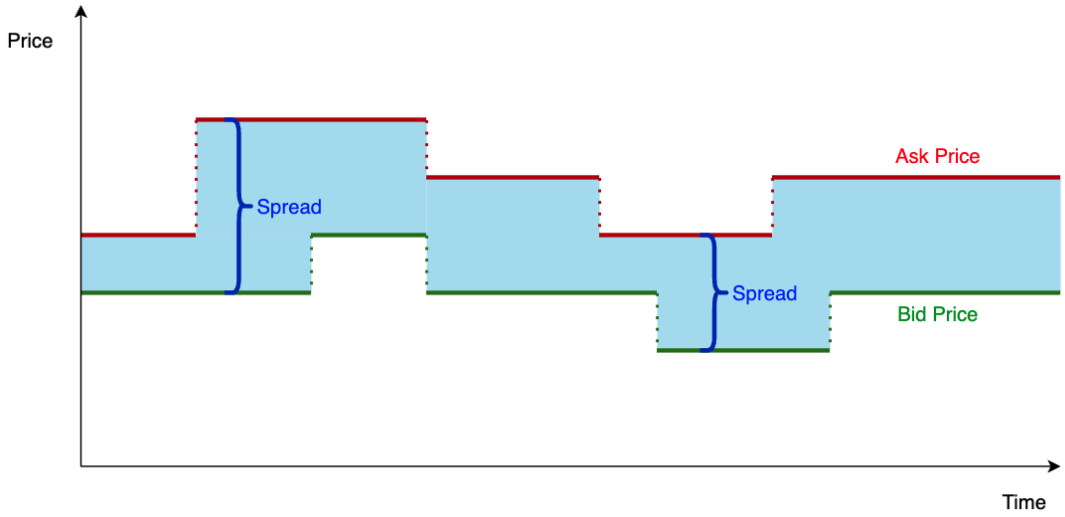

Figure 2.1: Diagram of Bid and Ask Spread changing over time

is the highest price at which someone is willing to buy a financial asset and the ask price is the lowest price at which someone is willing to sell a financial asset. When a transaction occurs, the bid and ask prices are both recorded and the time at which this transaction occurs is recorded. The recorded timestamp is usually accurate down to a thousandth of a second, this gives an idea of how quickly these transactions occur. The difference between the ask and the bid price is referred to as the spread (see Figure 2.1). The spread represents the value gained by the market maker for facilitating a trade, spread is a necessary requirement to the trading process as it acts as a profit incentive for market makers. Market makers hold the primary responsibility of sustaining liquidity within the market, as well as providing traders with up-to-date prices and other information about the price. Market makers are able to offer lower spreads in more liquid markets [29]. The spread varies over time but is often a fraction of a price move, therefore traders often average out the ask and bid price to give them a 'mid-price'. The mid-price (see Equation 2.1) is a useful abstraction of the bid and ask price as it creates a univariate sequence of prices that represent the change in price over time.

$$\text{Mid-Price} = \frac{(\text{Ask Price} + \text{Bid Price})}{2} \tag{2.1}$$

Each transaction is recorded and generates a tick, a set of ticks that have been recorded over time produces an irregularly spaced and noisy set of prices. This presents the requirement for sampling the data in order to create a more distinct representation of past price movements, with the intention of gaining a clearer understanding of the latent price behaviour. The ability to understand the price behaviour of a particular financial instrument is key to successful price prediction. The more successfully the price can be predicted, the more financial reward can be obtained from trading activity within the market.

Fixed Interval Sampling When using fixed interval techniques to sample price data, traders like to pick a certain fixed interval to trade at. Fixed interval sampling periods can range from one second to one year and in some cases even longer. This sampling algorithm works by identifying a sampling interval and a starting timestamp, the algorithm then keeps track of the most recent tick price to be posted until it is one interval away from the starting timestamp. The most recent tick before the new interval is recorded as the price at that point in time, this process then repeats as new prices are posted.

Fixed interval sampling is the most popular choice of sampling method as it is simple and can be very effective. With the simplicity of this method however, there are some drawbacks. Due to the nature of the fixed interval sampling method, various significant price moves can go undetected. Since the algorithm is naively sampling at a regular interval, if the price were to change drastically and then return to a similar price level between two sampling points, that price move is missed by the algorithm and not reflected in the resultant series of sampled prices. This missed price movement constitutes a loss in potential profit for a trader using fixed interval sampling as they are unaware of any price movement outside of the sampled set of prices. Techniques used to counteract this issue will be covered in the following Event-Based Sampling section.

Event-Based Sampling The limitations of fixed interval sampling become especially evident in high-frequency trading (HFT) scenarios, where its rigid sampling intervals can lead to substantial information loss. In such cases, the method's inability to capture the finer price dynamics can result in a distorted view of market movements. Significant price changes that occur between the fixed sampling points may go unnoticed, causing traders to miss critical opportunities. This can be particularly detrimental in markets where price action is volatile, such as the foreign exchange market, and split-second decisions can be the difference between a profit and a loss. To address this issue, alternative sampling methods that focus on event-driven sampling have gained traction. Techniques such as "Important Points" [30], "Turning Points" [31], "Perceptually Important Points" [32] and the "Zigzag" [33] model have been introduced to mitigate the issues inherent in fixed interval sampling. These methods focus on capturing data only when meaningful price events occur, thus retaining more relevant and actionable market information.

One event-based approach that has attracted considerable academic interest is the directional changes (DC) framework, first introduced by Guillaume et al. in 1997 [5]. This method differs fundamentally from fixed interval sampling by focusing on the significant directional shifts in price, rather than arbitrary time-based intervals. The DC framework has proven particularly effective in capturing meaningful price movements, especially in high-frequency environments where market behaviour is fast-paced and complex. Recent studies, such as [34–36], have further validated the effectiveness of the DC approach in both predicting market regime changes and enhancing trading strategies. By concentrating on price movements that exceed a certain threshold, the DC method filters out noise and allows traders to focus on the most significant changes in price. As a result, it has become a valuable tool when preparing data for training machine learning models, offering a more nuanced and information-rich approach to price data analysis compared to the traditional fixed interval paradigms.

Directional Change (DC) Sampling relies on identifying significant changes in the market and using the occurrence of these changes to trigger a sampling mechanism. This whole DC sampling process yields a contiguous set of 'trends'. Each trend consists of a directional change (DC) event and an overshoot (OS) event. These two events can be defined by three points: the DC start point, DC confirmation point (DCC), and the DC extreme point (DCE) as shown in Figure 2.2. The DC start point of one trend is the DCE point of the previous trend, hence the contiguous nature of the set of trends. Each trend is classified as either an upturn or a downturn. Upturns and downturns alternate throughout the course of the data, and the sampling algorithm is initialised at the DCC point of an upturn trend.

Once the DC sampling algorithm has been initialised at the DCC point of an upturn trend, the OS phase of the first upturn event has been triggered. From this point, any price data that is observed is in this OS phase unless the price changes significantly enough to trigger the retroactive classification of a trend in the opposite direction (i.e. a downturn). The significance of this price change is determined by the DC threshold (θ). The DC threshold is the percentage by which the price has to change in order to trigger an opposite trend and is selected prior to sampling by the trader. The DC threshold is therefore often considered a parameter of the trading algorithm that can be manipulated during optimisation. Once the significant price change is observed, the highest price since the DCC point of the previous trend is considered the DCE point of the previous trend and the DC start point of this new downturn trend. Any price data observed since that DC start point and the current triggering point is then considered price data during the DC event of the downturn event. Since the DCC point of the opposite trend has now been identified and noted, the current state now corresponds to the OS portion of a downturn. This process then propagates forwards through the data until there are no more new prices to sample. The pseudo-code for this DC sampling process is defined in Algorithm 1.

Algorithm 1 DC Sampling Algorithm

Initialisation: Initialise variables: event is Upturn event, $p^h = p^l = p(t_0), \theta \geq 0, t_0^{dc} = t_1^{dc} = t_0^{os} = t_1^{os} = t_0$

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 a Downturn Event
    $t_0^{dc} \leftarrow t$ // Start time for a Downturn Event
    $t_1^{os} \leftarrow t - 1$ // End time for an Upturn Overshoot Event
  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 an Upturn Event
    $t_1^{dc} \leftarrow t$ // End time for an Upturn Event
    $t_0^{os} \leftarrow t + 1$ // Start time for an Upturn Overshoot Event
  else if $p^l > p(t)$ **then**
    $p^l \leftarrow p(t)$ // Price at start of an Upturn Event
    $t_0^{dc} \leftarrow t$ // Start time for an Upturn Event
    $t_1^{os} \leftarrow t - 1$ // End time for a Downturn Overshoot Event
  end if
end if

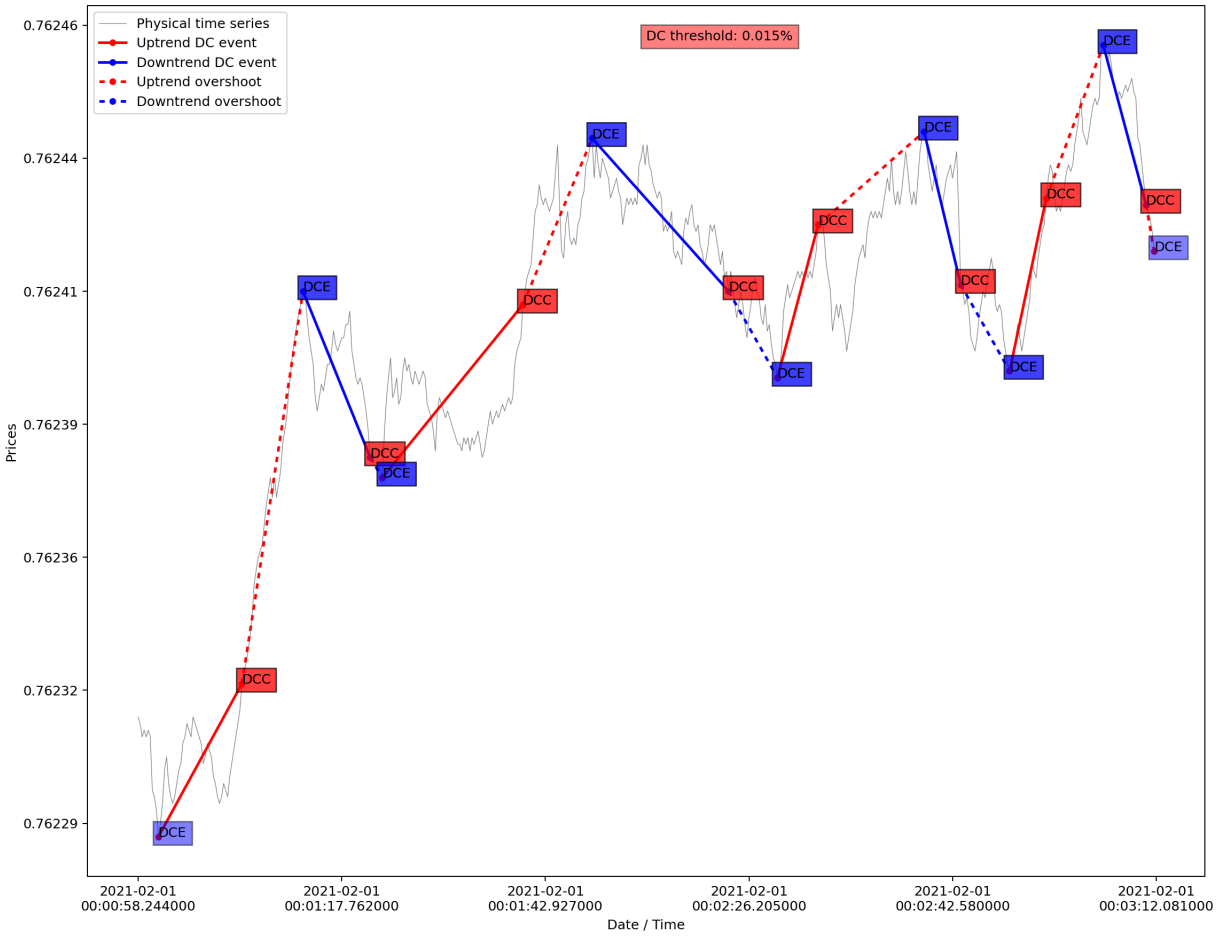

Figure 2.2: Directional Change Sampling Diagram of AUDUSD

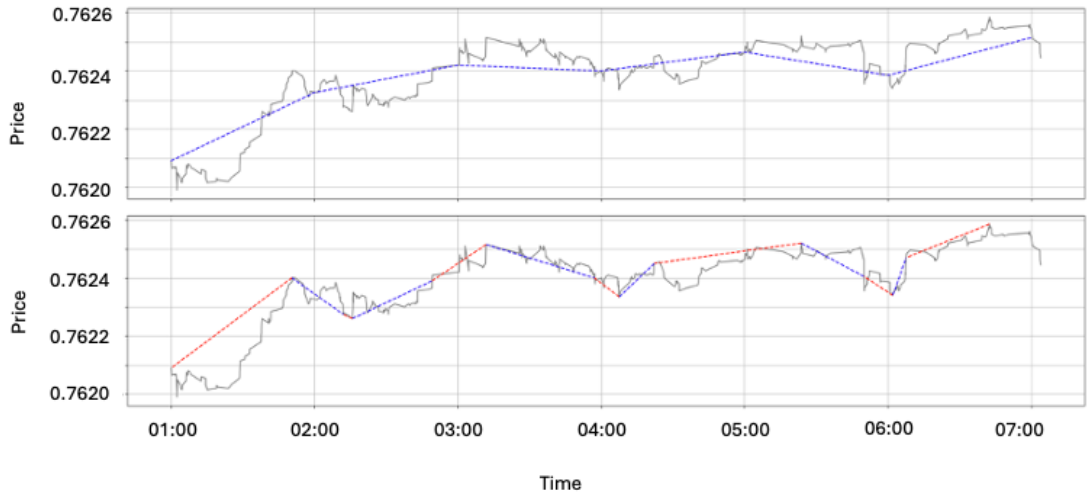

Figure 2.3: Fixed Sampling vs DC Sampling

The benefits of DC sampling are most apparent in the scenario shown in Figure 2.3. There are a number of occasions where there has been a significant move in the underlying tick price that has been missed by the fixed interval sampling approach. Of the six segments in both diagrams, the clearest example of the benefit of DC sampling is at the end of the first segment, where there is a spike in the price that is missed by fixed interval sampling but picked up by DC sampling and registered as the extreme point of of the first upturn. By taking a step back and observing the way the sample line follows the underlying price data as well, it is clear how much more closely the DC sampling algorithm follows the data, with all unsampled prices simply forming noise. The same however, cannot be said for the fixed interval sampling.

Within the paradigm of directional changes (DC) sampling, numerous properties emerge that are often obscured under traditional fixed interval sampling methodologies. These properties, referred to as "Scaling Laws" in the literature, represent empirically derived relationships that capture consistent patterns in price movements across a wide range of financial assets. The identification and formalisation of these scaling laws have played a crucial role in advancing the understanding of financial market dynamics under the DC framework. By revealing patterns that are otherwise hidden in time-based approaches, these scaling laws provide a more nuanced and informative methodology through which price behaviour can

be analysed and predicted.

The concept of scaling laws in the DC context was first empirically formalised by [5], who used high-frequency data from thirteen major foreign exchange markets to lay the foundations for the discovery of 12 distinct DC-based scaling laws in [37]. These laws describe a variety of regularities in market behaviour, including the proportional relationships between price movements in the OS move and the DC move, as well as the frequency and magnitude of these movements. Of the 12 scaling laws formalised by [37] the scaling law stating that $2DC \approx OS$ is perhaps the most well recognised. After [37], [38] and [39] identified four and five further scaling laws within the FX and stock markets respectively and more recently [35] and [36] identified five more scaling laws. By systematically observing these patterns, the authors have helped develop a foundational framework that has since been extensively used in the development of DC-based trading strategies such as in [40].

Directional Change (DC) sampling has also emerged as a promising approach for developing effective trading strategies without relying on traditional machine learning techniques. [41] uses the DC framework to categorise price fluctuations into significant market events. This work introduces the Scale of Market Quakes (SMQ) concept and offers a novel means of evaluating trading volatility and market behaviour. [42] also implements as DC based trading strategy that achieves promising profitability by leveraging the DC sampling framework. Their findings further support the effectiveness of DC-based methodologies by introducing three DC based strategies, each of which provides positive return on investment.

[43] demonstrated the potential of rule-based DC strategies, achieving forecasting accuracy greater than 80%. This concept was further developed through various implementations, including the Dynamic Backlash Agent (DBA) strategy [44], which showed high profitability potential with Alpha values over 10 in FX markets, though bid-ask spreads significantly impacted returns under certain conditions. [45] advanced the field by introducing a dynamic threshold method

that adapts to market conditions based on previous price movements, proving more effective at detecting significant market events compared to static thresholds. [46] also develops a DC based approach, their evaluation of FX market data demonstrates that strategies based on directional changes integrated with technical analysis improve both efficiency and sustainability in high-frequency trading.

Further research introduced the TSFDC strategy in [47] which incorporated a forecasting model to predict trend direction changes, demonstrating improved performance compared to existing DC-based strategies across multiple currency pairs. The Intelligent DBA (IDBA) strategy [48], which addressed the limitations of the original DBA by incorporating order size and risk management components, achieved approximately 30% annualised returns after accounting for bid-ask spreads. The evolution of DC-based trading culminated in [49]'s comprehensive approach, which combined intrinsic event-based time redefinition with agent-based modelling principles, resulting in a scalable trading model that generates significant profits.

2.3 Technical Analysis

Technical analysis is the application of statistical and visual patterns to historical price data in order to gain insight on the future movement of the price of an asset [50]. It is offered as an alternative or complementary approach to fundamental analysis, a topic that is out of scope for this thesis. A key component of technical analysis is the ability to visualise how price changes over time and using this to identify patterns. The invention of the candlestick for representing price changes (shown in figure 2.4) can be traced back to rice traders in 17th century Japan but was introduced to the western world by Nison [51]. The candlestick patterns outlined by Morris in [52] are examples of the structures multiple candlesticks can take and how certain configurations can be used to signify the change in direction of the market. The candlestick concept is applied in Chapter 6 on top of the DC sampling mechanism to help the machine learning algorithm learn.

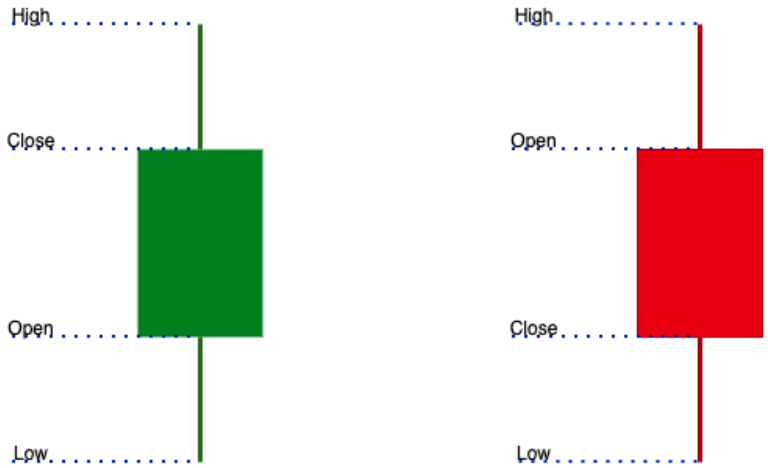

Figure 2.4: Candlesticks Diagram

Candlesticks are particularly useful for human traders who prefer trading based on visual patterns, but when making trading decisions it is useful to have statistical metrics that also support trading decisions, technical indicators are used to do this.

Technical Indicators Technical indicators have shown to be an effective way of improving the ability to predict the future movement of a price [53]. A series of sampled prices provide the information from which more insight can be extracted using technical indicators. Technical indicators are the result of statistical calculations based on the price of a financial instrument over a specific period of time. They take different forms depending on which feature of the price change they are trying to represent. The taxonomy applied in this thesis splits technical indicators into the four categories of trend-following, momentum, volatility and oscillators. Trend-following indicators are used to measure the strength and direction of a trend, momentum indicators are used to identify the strength and speed of various price moves, volatility indicators measure how erratic the price movements are within a given market and finally, oscillators are used to suggest trend reversals by identifying when a financial instrument is overbought or oversold.

Some of the most common technical indicators used include: Simple and Exponential Moving Averages, MACD (Moving Average Convergence Divergence),

RSI (Relative Strength Index), Bollinger Bands and Stochastic Oscillators. These can be calculated using the following equations.

Simple Moving Average The Simple Moving Average (SMA) calculates the average price over a specific period of time but summing up the series of prices over the specified period of time and dividing by the number of periods. By doing this iteratively for every new price it gives traders a clearer view of market direction and helps to filter out short-term noise.

$$\text{SMA} = \frac{1}{N} \sum_{t=1}^{N} X_t \tag{2.2}$$

where:

- N: period (the number of previous prices).
- X_t: price at time step t.

Exponential Moving Average The Exponential Moving Average (EMA), similarly to the simple moving average, calculates the average price over a specific period of time. The exponential moving average differs by giving more weight to more recent prices. This helps traders view clearer market direction as well as reacting more quickly to sharp changes in price, a limitation of the simple moving average.

$$\text{EMA}_t = \alpha \cdot X_t + (1 - \alpha) \cdot \text{EMA}_{t-1} \tag{2.3}$$

where:

- α: smoothing factor, calculated as $\frac{2}{1+N}$ where N is the period of the indicator.
- X_t: price at time step t.
- EMA_{t-1}: EMA value at the previous time step.

Moving Average Convergence/Divergence The Moving Average Convergence/Divergence (MACD) indicator is a momentum indicator that calculates the difference between a short-term and a long-term EMA, with an mid-term EMA used as a signal line to identify buy or sell opportunities. It is often used by traders to spot trend direction, trend strength and potential reversals by analysing crossovers and histogram momentum.

$$\text{MACD}_t = \text{EMA}_{\text{short}}(X_t) - \text{EMA}_{\text{long}}(X_t) \tag{2.4}$$

$$\text{MACD Signal Line}_t = \text{EMA}_{\text{signal}}(\text{MACD}_t) \tag{2.5}$$

where:

- $\text{EMA}_{\text{short}}(X_t)$: short-term Exponential Moving Average of the price X_t.
- $\text{EMA}_{\text{long}}(X_t)$: long-term Exponential Moving Average of the price X_t.
- $\text{EMA}_{\text{signal}}(\text{MACD}_t)$: Exponential Moving Average of the MACD line, used as the signal line.

Relative Strength Index The Relative Strength Index (RSI) is a momentum oscillator that measures the speed and change of price movements on a scale of 0 to 100 over a given period to identify overbought or oversold conditions typically above 70 and below 30 respectively. It is often used by traders to assess potential trend reversals, identify bullish or bearish divergences and confirm price momentum strength.

$$RS = \frac{\text{Average Gain}}{\text{Average Loss}} \tag{2.6}$$

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

where:

- **Average Gain**: average of all the upward price movements (up closes) over n periods.
- **Average Loss**: average of all the downward price movements (down closes) over n periods.

Bollinger Bands Bollinger Bands consist of a middle band and two outer bands set at two standard deviations above and below the middle band and are dynamically adjusted based on market volatility. They are often used by traders to identify overbought and oversold conditions, detect volatility contractions that signal potential breakouts and confirm trends based on price movement relative to the bands.

$$\text{Upper Band} = \text{SMA} + (2 \times \text{SD}) \tag{2.8}$$

$$\text{Lower Band} = \text{SMA} - (2 \times \text{SD}) \tag{2.9}$$

where:

- **SMA**: Simple Moving Average of the price.
- **SD**: standard deviation of the price over the same period as the SMA.

Stochastic Oscillator The Stochastic Oscillator is a momentum indicator that compares a price to the price range over a set period to identify overbought and oversold conditions, typically greater than 80 and less than 20 respectively. This is used by traders to detect potential trend reversals, measure price momentum and confirm divergences between price action and the oscillator, which can signal weakening trends.

$$\%K = \frac{\text{Closing Price} - \text{Lowest Low}}{\text{Highest High} - \text{Lowest Low}} \times 100 \tag{2.10}$$

$$\%D = \text{3-day SMA of \%K} \tag{2.11}$$

where:

- **Closing Price**: most recent closing price.
- **Lowest Low**: lowest price over the look-back period.
- **Highest High**: highest price over the look-back period.
- %D: 3-day Simple Moving Average (SMA) of %K.

Directional Change Indicators Technical indicators are most closely aligned with fixed time interval sampling methods but the same concept can be applied to data sampled under the DC paradigm. These indicators can also be tailored to represent the clearer signals provided under this sampling method. The DC specific indicators tend to use information gleaned from the scaling laws as this can demonstrate the state of the market within the framework defined by the scaling laws. The following includes the DC indicators used within this thesis.

TMV

TMV is the ratio of the whole price move to the DC threshold. The calculation for this indicator is defined in Equation 2.12.

$$TMV = \frac{|\Delta price|}{\theta} \tag{2.12}$$

where:

- $|\Delta price|$: absolute price change
- θ: DC threshold

OSV

OSV represents the percentage change between the current DCC price and the previous DCC price, normalized by the DC threshold, as shown in Equation 2.13.

$$OSV = \frac{\left(\frac{DCC_t - DCC_{t-1}}{DCC_{t-1}}\right)}{\theta} \tag{2.13}$$

where:

- DCC_t: current DCC price
- DCC_{t-1}: previous DCC price
- θ: DC threshold

$\mathbf{R_{DC}}$

R_{DC} is the number of ticks observed over the course of a trend, adjusted by the return of the trend, as shown in Equation 2.14.

$$R_{DC} = \frac{(TMV * \theta)}{\Delta Trend_t} \tag{2.14}$$

where:

- TMV: Total Move Value
- θ: DC threshold
- $\Delta Trend_t$: change in the trend at time t

$\mathbf{T_{DC}}$

T_{DC} is the number of ticks over the course of the trend, as shown in Equation 2.15.

$$T_{DC} = \Delta Trend_{no.ticks} \tag{2.15}$$

where:

- $\Delta Trend_{no.ticks}$: change in the number of ticks during the trend

$\mathbf{N_{DC}}$

N_{DC} is the number of ticks over the course of multiple trends, as shown in Equation 2.16.

$$N_{DC} = \sum_{i=0}^{n} Trend_{no.ticks_i} \tag{2.16}$$

where:

- $Trend_{no.ticks_i}$: number of ticks in trend i
- n: total number of trends considered

$\mathbf{C_{DC}}$

C_{DC} is the sum of $|TMV|$ over a certain number of trends, as shown in Equation 2.17.

$$C_{DC} = \sum_{i=0}^{n} |TMV|_i \tag{2.17}$$

where:

- $|TMV|_i$: absolute Total Move Value of trend i
- n: total number of trends considered

$\mathbf{A_T}$

A_T is the difference between the number of ticks spent on an uptrend and a downtrend over a certain number of trends, as shown in Equation 2.18.

$$A_T = \sum_{i=0}^{n} UpTrend_{no.ticks_i} - \sum_{i=0}^{n} DownTrend_{no.ticks_i} \tag{2.18}$$

where:

- $UpTrend_{no.ticks_i}$: number of ticks in the uptrend of trend i
- $DownTrend_{no.ticks_i}$: number of ticks in the downtrend of trend i

- n: total number of trends considered

2.4 Summary

In this chapter the concepts of financial forecasting have been explored from the ground up, beginning with the efficient market hypothesis and the random walk theory, claiming that in order to gain an edge in the market a trader needs to have some form of informational edge that they can then use to predict future price movement. Following this, the approaches to sampling raw financial data were discussed, along with the differences between the more traditional approach of fixed interval sampling and the more modern event-based approaches with a focus on directional change sampling. Once the background information around the sampling algorithms had been established, details on how to apply technical analysis to both fixed interval sampling and DC sampling were given. In the next chapter the inner workings of the Machine learning and Reinforcement learning algorithms applied in this thesis are discussed and context is gathered for the research in this thesis by reviewing the relevant literature.

Chapter 3

Machine Learning Algorithms

Machine learning (ML) is a broad topic that refers to the use of mathematical algorithms to develop models that encode real world relationships between a set of inputs and outputs. Machine learning is typically split into three main subcategories: supervised learning, unsupervised learning and reinforcement learning. Supervised learning is the most common application of machine learning that aims to simply map inputs to outputs using a set of observed features and target values. Unsupervised learning identifies similarities in data without explicitly being given target values, enabling patterns to be recognised from the input data alone. Reinforcement learning involves the training of an agent that is capable of interacting with an environment in a way that achieves a maximum amount of reward for the agent. The information in the following Section 3.1 focuses on supervised learning methods. Reinforcement learning algorithms are then covered later in Section 3.2. Unsupervised learning methods are not covered in any more detail as they are outside the scope of this thesis.

Before examining specific machine learning techniques, it is useful to distinguish ML-based forecasting from traditional econometric approaches. Econometric models, such as ARIMA, GARCH, and VAR, rely on explicit statistical assumptions about the data-generating process, typically requiring stationarity, linearity, and pre-specified functional forms, which prioritise parameter interpretability and allow hypothesis testing through established statistical frameworks. ML algorithms

however, learn relationships directly from data in a largely non-parametric way, capturing complex, non-linear patterns without imposing predefined structures, though often at the cost of reduced interpretability. For the high-frequency FX trading problem examined in this thesis, ML approaches are more appropriate than econometric methods because the non-linear and non-stationary nature of tick level price dynamics violates many econometric assumptions, the large data volumes align well with deep learning requirements, and the sequential decision-making nature of trading is particularly suited to reinforcement learning frameworks that can learn adaptive policies responding to evolving market conditions.

3.1 Supervised Learning Techniques

Supervised learning is the most popular sub-category of machine learning, mainly due to its adaptability to different problems. Supervised learning itself can be split further into shallow and deep learning methods. The terminology of 'shallow' is used in this thesis to distinguish between the machine learning methods that do not use deep learning and those that do. This section is divided into two subsections, Subsection 3.1.1 first discusses the shallow supervised learning methods that are referred to in the literature review of the application of ML to DC sampling, this is then followed by Subsection 3.1.2 which discusses the deep supervised learning methods used in this thesis.

3.1.1 Shallow Supervised Learning Methods

The following content explains supervised learning from the outlook of a regression problem, where inputs are mapped to continuous outputs. If inputs are mapped to a discrete set of values then this is referred to as a classification problem. Both regression and classification approaches are common in the field of financial forecasting but for simplicity, the following subsection will look exclusively at regression applications of the algorithms covered. In many cases, the classification

approach is very similar to the regression approach, just with a few tweaks to the optimisation function to enable discrete outputs of the models. The following subsection is by no means a comprehensive overview of all shallow supervised learning methods, but aims to give an overview of the fundamentals of those referred to in the literature review in Section 3.3.

Linear Regression Linear regression [54], or more specifically the Ordinary Least Squares method (OLS) [55], is a fundamental statistical method that models the linear relationship between a dependent variable and independent variables by minimising squared differences using the least squares approach (see Equation 3.1). The model assumes that this relationship is linear, meaning that changes in the independent variables result in proportional changes in the dependent variable. Despite being a relatively simple method of modelling relationships in data, it has proven to be particularly effective as it is both explainable and unlikely to overfit the data.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon \tag{3.1}$$

where:

- Y: dependent variable.
- $X_1, X_2, \ldots, X_n$: independent variables.
- β_0: y-intercept.
- $\beta_1, \beta_2, \ldots, \beta_n$: slope of each independent variable.
- ϵ: error or unobserved factors influencing Y.

Bayesian Ridge Regression Bayesian Ridge Regression is an alternative linear regression approach where the coefficients of the model are assumed to be random variables, as opposed to fixed as in the previously described OLS method. Using

Bayesian principles to estimate model parameters [56], it allows for the incorporation of prior information. This makes it effective in situations where other machine learning methods might struggle due to a scarcity of data. The Bayesian Ridge Regression model assumes a Gaussian prior distribution on the regression coefficients, and it employs Bayes' theorem to update the prior distribution based on the observed data. The goal is to find the posterior distribution of the coefficients, taking into account both the prior information and the likelihood of the observed data given the model.

Elastic Net Regression Elastic Net Regression [57] is another variant of linear regression that uses a combination of L1 [58] and L2 [59] regularisation techniques to improve on each technique individually. Elastic Net regression is considered useful for high dimensional data due to the combined effect of both the L1 and L2 regularisation techniques employed.

Stochastic Gradient Descent Regression Stochastic Gradient Descent Regression [60] uses Stochastic Gradient Descent (SGD) [61] to solve linear regression by minimising the Ordinary Least Squares (OLS) objective function. Unlike traditional linear regression, which computes an exact solution by minimising the residual sum of squares using all data at once, SGD regression updates the model parameters iteratively based on small, randomly selected batches of data, making it highly efficient for larger datasets where processing all data at once is impractical. SGD regression also allows for regularisation to prevent overfitting, providing more flexibility compared to standard OLS.

Kernel Ridge Regression Kernel Ridge Regression [62] is a regression algorithm that combines linear regression with kernel methods. Kernel Ridge Regression uses the kernel trick [63], which allows the algorithm to operate in a higher-dimensional space without explicitly transforming the input features. Kernel Ridge Regression minimises the sum of squared differences between observed

and predicted values while penalising the magnitude of the coefficients using a regularisation term. The choice of kernel function, such as the Gaussian or polynomial kernel can also tweak the learning algorithm to make it more appropriate for different types of problems.

Support Vector Machine Support Vector Machines (SVMs) [64] are designed to find a hyperplane that best separates different classes in the feature space with the goal of maximising the space between them. SVMs can handle linear and non-linear relationships in the data and are considered very effective in high-dimensional spaces, as well as handling outliers efficiently. The algorithm assigns data points to different classes based on their position relative to the hyperplane. SVMs can be extended to regression tasks using Support Vector Regression (SVR) [65], which applies the principles of Support Vector Machines (SVMs) to predict continuous values rather than class labels.

Decision Trees Decision trees [66] are constructed through a recursive process where a series of decisions are made to split data into subsets with the intention of maximising information gain or reduce variance for classification and regression tasks respectively. This process can be modelled with a tree like structure, consisting of nodes and branches. The tree is developed until a stopping criterion is met, such as a certain depth or a minimum number of samples per leaf. Decision trees are particularly popular for their explainability in decision-making, as the paths from the root to the leaves explicitly show the exact criteria used for generating the output.

Random Forests Random Forests [67] are an ensemble learning technique built upon the foundation of decision trees. This algorithm constructs a number of decision trees during training, each on a random subset of the training data and using a random subset of the features. The predictions of the individual trees are then aggregated through voting for classification tasks, or averaging for regression

tasks. The randomness used in both data and feature selection helps mitigate overfitting and improves the model's robustness and generalisation performance. Random Forests are often used for their ability to produce reliable predictions when dealing with noisy or incomplete data.

Gradient Boosting Regression Gradient Boosting Regression [68] is an ensemble learning method that builds a predictive model by sequentially combining the outputs of multiple weak learners, most commonly decision trees. The algorithm begins by fitting a simple model to the data and subsequent models are constructed to correct the errors of the previous ones. The key idea is to optimise a loss function, such as mean squared error for regression tasks, by adjusting the weights of individual learners. Gradient Boosting focuses on minimising the residuals or errors in predictions, gradually improving the model's accuracy.

In recent years, deep learning has emerged as a powerful extension of traditional machine learning techniques, demonstrating the ability to learn complex representations of data. Unlike classical machine learning methods, which are often limited by their ability to capture intricate relationships, deep learning models can automatically learn these relationships from raw data through multiple layers of abstraction. This capacity has allowed deep learning to outperform older methods across a wide range of tasks, including computer vision, natural language processing and control systems. The key advantage of deep learning lies in its scalability and flexibility. With large datasets and increased computational power, these models have consistently achieved state-of-the-art results, surpassing conventional algorithms in both accuracy and generalisation. This paradigm shift has not only improved performance in well established domains but has also unlocked new possibilities in areas previously considered too complex for machine learning models.

3.1.2 Deep Supervised Learning Methods

Both regression and classification implementations of deep neural networks are components of deep reinforcement learning, further details on this will be provided in Section 3.2. The following subsection therefore takes a conceptual approach to explaining deep supervised learning to provide the relevant background information for when deep reinforcement learning is covered. The deep learning architecture used in this thesis is focussed on the Multi-Layer Perceptron (MLP) [69] neural network architecture due to its versatility. The fundamentals of deep learning architectures such as Recurrent Neural Netorks (RNNs) [70] and Long-Short Term Memory (LSTM) [71] are covered to provide the required background information for when these algorithms are mentioned in related works. Convolutional Neural Networks (CNNs) [72] and Transformers [73] are considered out of scope for this thesis, but could form the basis of future work.

MLPs Mulilayer Perceptrons (MLPs) are a type of artificial neural network that consists of a single input layer, one or more hidden layers and then a single output layer. Each layer is composed of nodes that are connected by weights. The input layer consists of as many nodes as there are features in the data. The hidden layers take information from the previous layer, whether this is an input layer or previous hidden layer, and sum up the values before passing this information through an activation function to scale the result. The scaled result is then passed to the next layer until the final layer, which will sum the input as before and produce a final logit value. In a regression setup the logits represent the predicted value, in a classification setup the logits are fed into a softmax function which calculates the confidence levels of each discrete output from which an argmax function can be applied to identify the recommended output class.

RNNs Recurrent neural networks (RNNs) [70] are an architecture of neural network built to handle sequential data. The recurrence relation in an RNN at time step t is mathematically expressed as shown in Equations 3.2 and 3.3.

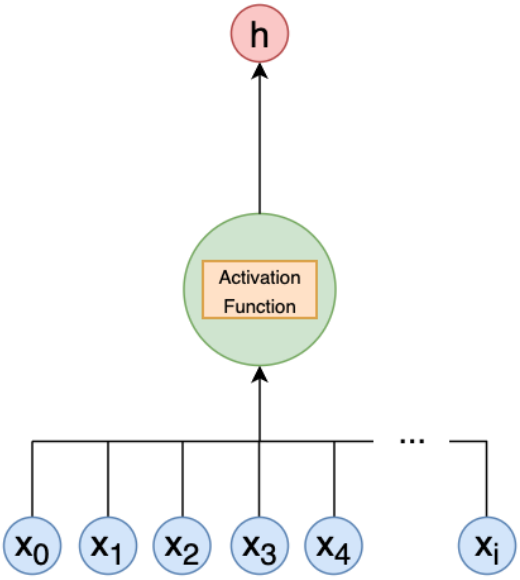

Figure 3.1: MLP Node

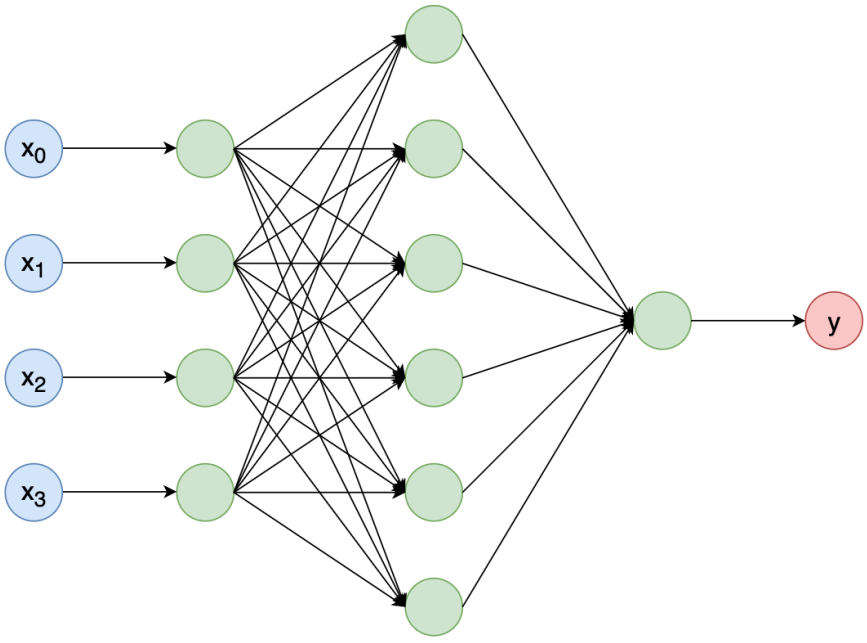

Figure 3.2: MLP Network

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{3.2}$$

$$y_t = W_{hy}h_t + b_y \tag{3.3}$$

where $x_t \in \mathbb{R}^{d_x}$ is the input vector at time t, $h_t \in \mathbb{R}^{d_h}$ is the hidden state at time t, $y_t \in \mathbb{R}^{d_y}$ is the output at time t, $W_{xh} \in \mathbb{R}^{d_h \times d_x}$ is the input-to-hidden weight matrix, $W_{hh} \in \mathbb{R}^{d_h \times d_h}$ is the hidden-to-hidden (recurrent) weight matrix, $W_{hy} \in \mathbb{R}^{d_y \times d_h}$ is the hidden-to-output weight matrix, b_h and b_y are bias vectors, and tanh is the hyperbolic tangent activation function. The critical feature of RNNs is the recurrent connection $W_{hh}h_{t-1}$, which allows information from previous time steps to influence the current computation.

Figure 3.3 shows a recurrent neural node unfolded through time, demonstrating how the hidden state h_t (Equation 3.2) propagates through sequential time steps. These recurrent nodes can be stacked horizontally and vertically to create a full network that is able to learn patterns in time series data when trained using the same techniques as the MLP. The nature of an RNN however gives rise to the exploding and vanishing gradient problem, this issue is a result of the recursive application of the chain rule during back-propagation (discussed in detail later in this section) which causes gradients to grow or diminish exponentially with each iteration [74]. The effects of this exploding or vanishing gradient problem can be tackled in a number of ways including limiting the size of the gradients with a threshold (gradient clipping) [75], penalising large weight values with regularisation [76], smaller weight initialisation [77] and adding residual connections [78]. An alternative to these approaches involve alterations to the internal network mechanism in itself in the form of LSTM networks.

LSTMs Long Short-Term Memory (LSTM) networks [71] are an improvement on the original recurrent neural network (RNN) architecture, designed to address the exploding/vanishing gradient problem. To address the gradient problems the LSTM introduces three gates: input, forget and output gate with the purpose of

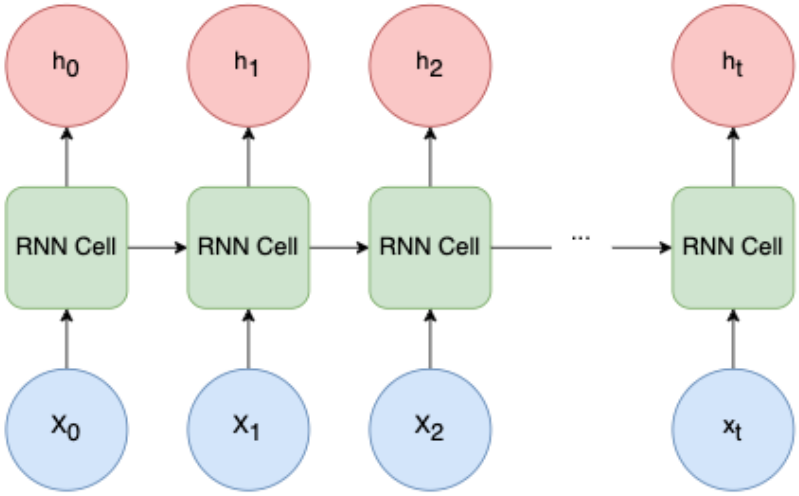

Figure 3.3: RNN Unfolded

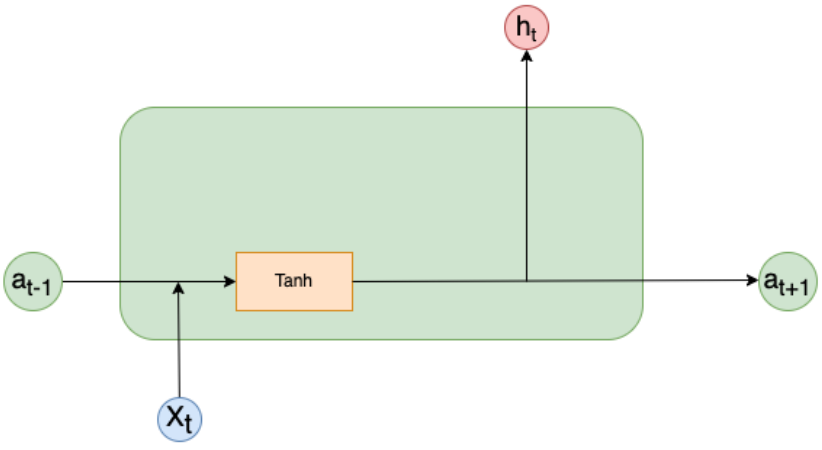

Figure 3.4: RNN Cell

controlling the flow of information.

The mathematical operations at each time step t are defined as follows.

The **input gate** determines which new information to store:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3.4}$$

The **forget gate** determines which information to discard from the cell state:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.5}$$

The **output gate** determines what information to output:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.6}$$

The **cell state update** is computed as:

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{3.7}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{3.8}$$

The **hidden state output** is:

$$h_t = o_t \odot \tanh(C_t) \tag{3.9}$$

where σ denotes the sigmoid activation function $\sigma(z) = \frac{1}{1+e^{-z}}$, tanh denotes the hyperbolic tangent activation function, $\odot$ represents element-wise multiplication, W_i, W_f, W_o, W_C are weight matrices for each gate, b_i, b_f, b_o, b_C are bias vectors for each gate, $[h_{t-1}, x_t]$ denotes the concatenation of the previous hidden state and current input, C_t is the cell state at time t, and h_t is the hidden state (output) at time t.

The forget gate f_t (Equation 3.5) multiplies the previous cell state C_{t-1}, allowing the network to selectively discard irrelevant information. The input gate

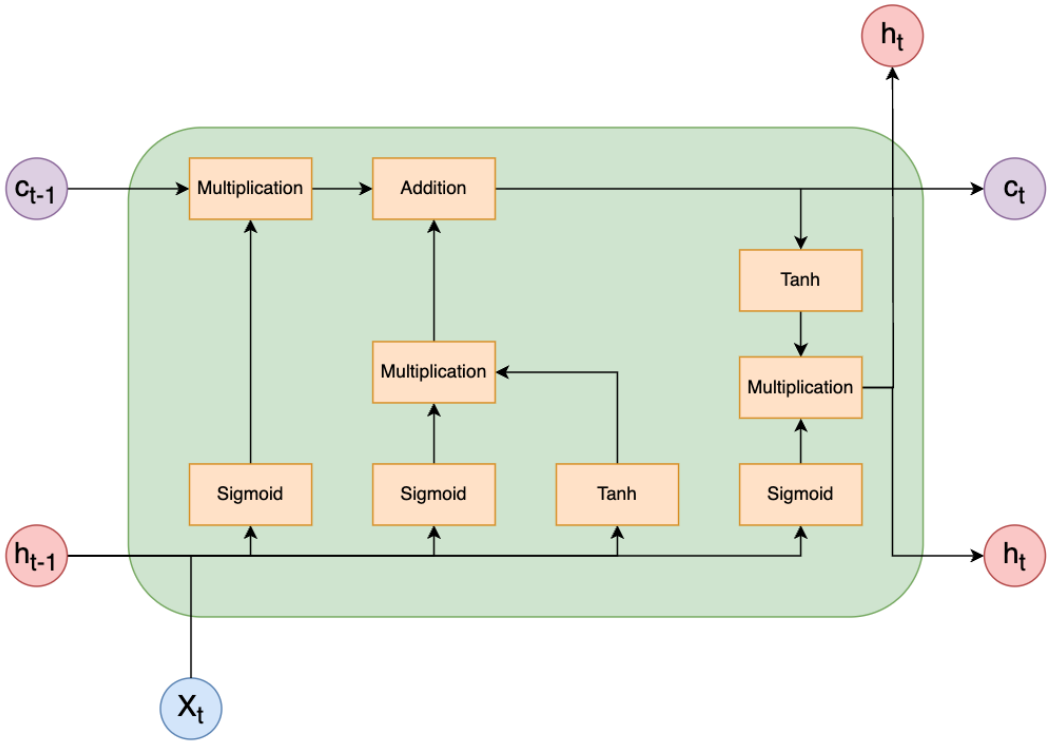

Figure 3.5: LSTM Cell

i_t (Equation 3.4) controls how much of the candidate cell state $\tilde{C}_t$ is added to the current cell state (Equation 3.8). Finally, the output gate o_t (Equation 3.6) determines which parts of the cell state are exposed in the hidden state output (Equation 3.9). This also allows LSTM networks to generate long-term relationships within the data, something RNNs are not well adapted to doing. The ability to identify these long-term relationships over time makes LSTMs particularly appropriate for time series data among other applications. This gating mechanism addresses the vanishing gradient problem by providing paths through which gradients can flow unchanged during backpropagation through time, as the additive cell state update in Equation 3.8 prevents gradient decay that occurs with repeated matrix multiplications in standard RNNs.

Back-propagation Now the architectures of deep neural networks have been established, the fundamental training algorithm of deep neural networks known as back-propagation can be reviewed. The algorithm was introduced in its modern form in 1970 in a masters thesis by Seppo Linnainmaa [79], however it was not

practically applied to deep learning until the work of Yann LeCun was published in 1989 [80].

Back-propagation works by calculating the gradient of a loss function with respect to the weights of a neural network. By finding the gradient of a loss function with respect to the weights, it can be identified whether individual weights need to be increased or decreased to minimise the error of the output given a target value. The mechanics of back-propagation are rooted in calculus, by using the chain rule to calculate the partial derivative of the loss function with respect to each weight, it can be deduced how much each weight contributes to the total error of the network.

In order to conduct back-propagation, the input values first need to be passed through the network to judge the current accuracy of the network, this is the 'forward pass' and is demonstrated in Equation 3.10.

$$a_i^l = f(z_i^l) = f\left(\sum_j w_{ij}^l a_j^{l-1} + b_i^l\right) \tag{3.10}$$

where:

- w_{ij}^l is the weight between neuron j in layer $l-1$ and neuron i in layer l.
- a_j^{l-1} is the activation of the neuron in the previous layer.
- b_i^l is the bias for neuron i in layer l.
- f is the activation function (e.g. ReLU function).
- z_i^l is the weighted sum before applying the activation function.

Once the forward pass is complete, an output prediction is provided for the given variables with the current state of the network. Using this predicted value and the target value, the error of the network can be calculated with the loss function. The loss function is interchangeable, but the most common approach is to use Mean Square Error (MSE) (see Equation 3.11) for regression problems and Cross-entropy Loss (see Equation 3.12) for classification problems.

$$L = \frac{1}{2}\sum_{k}(\hat{y}_k - y_k)^2 \tag{3.11}$$

where:

- $\hat{y}_k$ is the predicted output for the k-th output node.
- y_k is the target value for the k-th output node.
- The factor $\frac{1}{2}$ is included for convenience during differentiation.

$$L = -\sum_{k} y_k \log(\hat{y}_k) \tag{3.12}$$

where:

- y_k is a binary value denoting if class label k is the correct.
- $\hat{y}_k$ is the predicted probability for class k.

Once the error has been calculated the extent to which the network is uncalibrated is known and so, in order to adjust the weights as required to reduce this error, the gradient of the loss with respect to the weights is calculated as defined in Equation 3.13.

$$\frac{\partial L}{\partial w_{ij}^l} = \delta_i^l a_j^{l-1} \tag{3.13}$$

where:

- $\delta_i^l = \frac{\partial L}{\partial z_i^l}$ is the error term for neuron i in layer l.
- a_j^{l-1} is the activation of the neuron in the previous layer $l-1$.

Each layer applies Equation 3.13 but relies on the gradient calculated by the subsequent layer (first layer relies on calculated gradient of layer two etc.) and therefore after calculating the gradient of each node in the output layer using Equation 3.14, each successive layer back uses Equation 3.15.

$$\delta_i^L = (\hat{y}_i - y_i) f'(z_i^L) \tag{3.14}$$

$$\delta_i^l = \left(\sum_j \delta_j^{l+1} w_{ij}^{l+1} \right) f'(z_i^l) \tag{3.15}$$

Once the gradients have been calculated, the weights can be updated using gradient descent, as demonstrated in Equation 3.16.

$$w_{ij}^l \leftarrow w_{ij}^l - \eta \frac{\partial L}{\partial w_{ij}^l} \tag{3.16}$$

where:

- η is the learning rate, a small positive number that controls the size of the weight adjustments.
- $\frac{\partial L}{\partial w_{ij}^l}$ is the gradient of the loss function with respect to the weight w_{ij}^l.

The training behaviour of a neural network can be influenced by a number of different techniques. The activation function of the nodes can be altered to control the output signal of each node which in turn can control the flow of gradient values through the network. Activation functions can vary but the most popular include ReLU, sigmoid and tanh [81]. Different optimisation techniques like Stochastic Gradient Descent (SGD), Adam [82] and RMSProp [83] can be applied to alter how the weights of the network are updated during training to prioritise factors like speed and stability. Network architecture can also play a factor when training a model and by increasing or decreasing the number of layers and nodes, the ability of the model to generalise to new data can be improved.

3.2 Reinforcement Learning Techniques

Reinforcement learning [84] is a subset of machine learning that aims to train an agent to interact optimally with an environment to achieve the maximum amount

of reward. A reinforcement learning problem consists of the interaction between two entities, an agent and an environment. The agent executes an action at each time step based on the observed state of the environment, which in turn causes a change in the state of the environment. With each new state comes a reward and the goal of the agent is to maximise reward over a given number of steps within the environment, the full number of steps taken by the agent before resetting an environment is known as an episode. The concept of training a reinforcement learning agent is analogous to training a pet dog to sit, when the dog (agent) performs the correct action of sitting, you provide them with a dog treat (reward). From this treat the dog associates the decision to conduct the action of sitting with a dog treat and are therefore more likely to do this action in the future as it will likely result in the same reward, therefore reinforcing this behaviour. This process can then be expanded to include multiple time steps, forming a sequence of interactions between the agent and environment with rewards being provided at irregular intervals. A sequence of interactions between the agent and environment can be modelled as a Markov Decision Process (MDP) [85] and can be represented as shown in Figure 3.6.

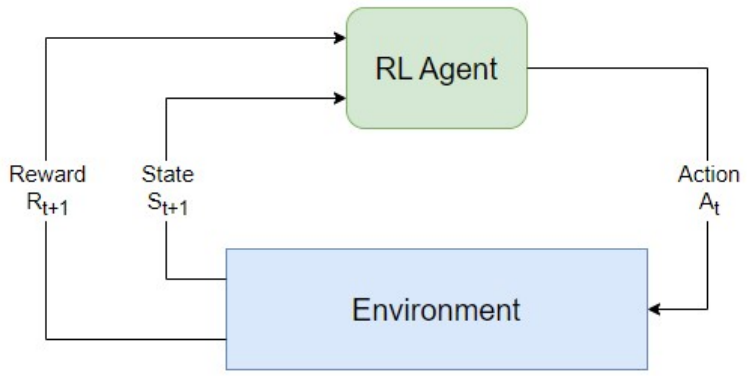

Figure 3.6: Markov Decision Process with RL Agent

A Markov Decision Process (MDP) is a mathematical framework used to describe agent decision making in environments where transitions between states is partially stochastic with an element of agent control. Agents navigate an environment by employing a policy, a policy refers to a function that maps a set of states

to a set of actions, therefore encoding the decision making process of the agent. The role of a reinforcement learning agent is to learn a policy that navigates an MDP to maximise cumulative reward over the length of an episode.

It is important to note that the application of the MDP framework to financial trading involves a departure from its classical assumptions. In a standard MDP, agent actions directly influence state transitions through the probability function $P(s' \mid s, a)$. In liquid foreign exchange markets however, individual trading agents operate at scales far smaller than the market as a whole, meaning their actions are extremely unlikely to alter the underlying market dynamics. The market state (prices, indicators and trends) evolves exogenously, driven by aggregate market forces beyond any single agent's control. Agent actions only affect the agent's own internal state, such as position and balance, rather than the market itself. The problem therefore cannot be strictly framed as a Markovian Control Process in the traditional sense. Despite this limitation, the MDP framework remains a valuable and widely used formalism for financial trading problems, as it provides a structured approach to sequential decision making and enables the application of reinforcement learning algorithms.

3.2.1 The Bellman Equation

The Bellman equation [86] forms the foundation of reinforcement learning, as it expresses the value of a current state based on the expected cumulative reward from that state. The objective of the reinforcement learning agent can be mathematically described as the optimisation of the Bellman equation. This Bellman optimality equation is presented in Equation 3.17.

$$V(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} P(s' \mid s, a) V(s') \right] \tag{3.17}$$

where:

- $V(s)$ is the value of state s, which represents the expected cumulative reward

that can be obtained from that state.

- $R(s, a)$ is the immediate reward obtained after taking action a in state s.
- γ is a discount factor that determines the importance of future rewards relative to immediate rewards.
- $P(s' \mid s, a)$ is the probability of transitioning to state s' from state s when taking action a.
- $\sum_{s'} P(s' \mid s, a)V(s')$ is the expected value of the future states that can be reached from state s when taking action a. The sum is taken over all possible future states s'.

3.2.2 Solution Methods for the Bellman Equation

Classical approaches to solving the Bellman optimality equation rely on dynamic programming techniques that guarantee convergence to optimal solutions in finite state and action spaces. Value iteration [87] is one such method that iteratively applies the Bellman optimality operator until the value function converges, updating the value of each state based on the maximum expected return over all possible actions. Policy iteration [88] is an alternative approach that alternates between two phases: policy evaluation, where the value function for the current policy is computed, and policy improvement, where a new policy is derived by selecting actions that maximise expected returns. While policy iteration often converges in fewer iterations than value iteration, each iteration is computationally more expensive as it requires solving a system of linear equations. Both methods require complete knowledge of the environment dynamics, specifically the state transition probabilities $P(s' \mid s, a)$ and reward function $R(s, a)$, making them model-based approaches.

Despite their optimality guarantees, these classical dynamic programming methods face significant practical limitations when applied to real-world problems. As the state and action spaces grow large or become continuous, the computational

requirements for storing and updating value functions across all states become prohibitive, a challenge known as the "curse of dimensionality" [89]. These limitations have driven the development of modern reinforcement learning algorithms, which are generally classified into two broad categories: model-based and model-free. Model-based algorithms incorporate a learned or approximated model of the environment, which enables the simulation of interactions during the learning process and facilitates greater sample efficiency, as the agent can generate trajectories starting from individual states without constant real-world interaction. Model-free algorithms however, rely on direct interaction with the environment over multiple time steps, enabling the agent to learn an optimal policy without explicitly modelling the environment itself [90]. In many practical applications, including financial trading, accurate models of environment dynamics are either unavailable or too complex to construct reliably. The methods employed in this research therefore focus on model-free techniques due to the inherent complexities of the foreign exchange market, where building a policy based on a learned environment model would carry a significantly higher risk of inaccuracies. Q-learning, discussed in the following subsection, represents one such model-free approach that approximates the solution to the Bellman equation without requiring knowledge of transition probabilities.

3.2.3 Q-Learning

Q-learning [91] is one of the original model-free reinforcement learning algorithms introduced in 1992 and relies on a value-based method. A value-based method means that the agent attempts to learn the inherent value of state-action pairs, known as the 'Q-value', and stores these in a table referred to as the 'Q-table'. The Q-table is updated after each step using the Q-learning update rule (see Equation 3.18), which uses principles from the Bellman Equation defined in Equation 3.17. The ultimate goal of this approach enables the agent to find the optimal policy to navigate an environment to maximise the resultant cumulative returns. One

limitation of Q-learning however is that it does not handle continuous or high-dimensional state spaces very well as it is designed to learn an explicit one-to-one mapping of state-action pairs to Q-values that can be stored in the Q-table. Taking this approach works well for discrete state spaces with a small number of possibilities, but as the number of potential states begins to grow, creating a mapping for every state-action pair becomes infeasible. The integration of the deep learning methods described in Section 3.1.2 offers a solution to this problem.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{3.18}$$

where:

- $Q(s_t, a_t)$: The Q-value for taking action a_t in state s_t.
- α: The learning rate, controlling how much new information overrides old information.
- r_t: The reward received after taking action a_t in state s_t.
- γ: The discount factor, determining the importance of future rewards (0 to 1).
- $\max_a Q(s_{t+1}, a)$: The maximum estimated Q-value for the next state s_{t+1}, across all possible actions.

3.2.4 Deep Reinforcement Learning

By extending standard Q-learning techniques, deep reinforcement learning (DRL) leverages deep learning models to overcome limitations in handling continuous or high-dimensional state spaces. The shift makes DRL more scalable and adaptable to various problems compared to the conventional use of Q-tables. A popular implementation arising from this adaptation is Deep Q-learning, where a neural network is employed to map states to Q-values for each possible action instead of a

Q-table, effectively learning the 'Q-function', in a similar way to how deep neural networks would learn input to output mappings in a supervised learning problem.

Deep Q Networks

The DQN algorithm was introduced by [6] in 2015 and was trained to play Atari games while receiving only the pixels as inputs. With this information, the agent was able to achieve a similar score to professional human players across all 49 games tested. As well as employing a deep neural network to learn the mapping of state and action pairs to a Q-value, a few other techniques are used in DQNs to perform well on a task. Experience replay [92] is one of these techniques, which stores agent experiences in a buffer and samples random mini-batches to break the correlations between consecutive experiences during training. This prevents the network from overfitting to recent trajectories. DQNs can also use a target network [93], which is a copy of the Q-network that is periodically updated to provide stable target values for the weight updates. The deep Q-network is trained to minimise the difference between the predicted Q-values and the target values generated using the target network. This is done by iteratively updating the Q-network's parameters via gradient descent. The pseudocode of the DQN algorithm is presented in Algorithm 2.

Algorithm 2 DQN Algorithm

Initialise experience replay memory $\mathcal{D}$ to capacity N

Initialise action-value function (Value network) Q with random parameters θ

Initialise target action-value function (Target network) $\hat{Q}$ with parameters $\theta^- = \theta$

Initialise learning rate $\alpha \in (0, 1)$, discount factor $\gamma \in [0, 1]$, batch size B, target update frequency C (typically 10^3 to 10^4)

for each episode **do**

  Initialise state s

  for each step in the episode **do**

    Select action a from s using ϵ-greedy policy based on $Q(s, a; \theta)$

    Take action a, observe reward r and next state s'

    Store transition (s, a, r, s') in memory $\mathcal{D}$

    Sample random mini-batch of transitions (s_j, a_j, r_j, s'_j) from $\mathcal{D}$

    For each mini-batch sample (s_j, a_j, r_j, s'_j), set target:

$$y_j = \begin{cases} r_j & \text{if episode terminates at } s'_j \\ r_j + \gamma \max_{a'} \hat{Q}(s'_j, a'; \theta^-) & \text{otherwise} \end{cases}$$

    Perform a gradient descent step on loss function:

$$L(\theta) = \frac{1}{B} \sum_{j=1}^{B} \left(y_j - Q(s_j, a_j; \theta)\right)^2$$

    Set $s \leftarrow s'$

    Every C steps, update target network: $\theta^- \leftarrow \theta$

  end for

end for

return Q

To balance exploration and exploitation during training, DQNs employ an

epsilon-greedy policy. With probability ϵ, the agent selects a random action to explore the environment, while with probability $1-\epsilon$, it exploits its current knowledge by selecting the action with the highest Q-value. ϵ typically starts at a high value (e.g. 1.0) and is gradually reduced to a lower value (e.g. 0.01) throughout training, encouraging exploration in early episodes and exploitation as the agent learns. This ϵ-greedy approach is a common exploration strategy used across many DRL algorithms, particularly in value-based methods, though policy-based methods often employ alternative exploration techniques.

Deep Q Networks are used to represent the value of taking an action in a state and are hence referred to as a value-based method. This is effectively a regression problem with a single output node in the DQN representing the Q-value. When using the trained model, a wrapper around this network is required to exploit the best strategy learned by the model which will then act as the agent's decision to take an action. The confidence levels of each action in a given state require an argmax function to be applied in order to get the associated index of the action to be taken. Not all DRL algorithms learn the value of a state and instead opt for learning the policy directly and are therefore referred to as policy-based methods. Policy-based methods learn in a similar way to value-based methods, by using experiences in an environment to update a deep neural network, however the setup of the neural network is slightly different. Policy-based methods have the flexibility to setup as a classification network if the action space itself is discrete. This means that the output layer of the neural network has a node per output action and a softmax function is applied to the final layer to provide confidence levels of each action. Value-based and policy-based methods can be combined in Actor-Critic methods to improve performance.

Actor-Critic Methods

Actor-Critic methods [94] represent a family of algorithms in reinforcement learning (RL) that combine the strengths of both policy-based and value-based ap-

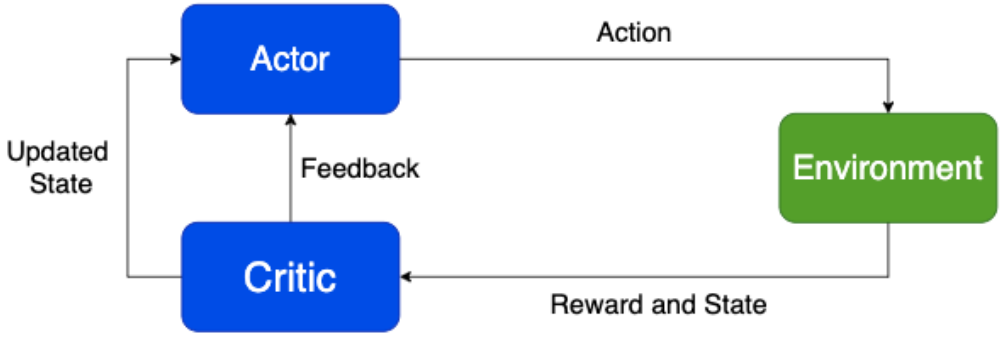

Figure 3.7: Actor Critic Diagram

proaches. In older DRL algorithms, value-based methods like Deep Q Networks focus on learning a value function that estimates the quality of any given state or state-action pair. Policy-based methods offer a slightly different approach by directly learning a policy that maps states to actions [95]. Actor-Critic methods unite these two paradigms by maintaining two separate components: an actor, which learns the policy, and a critic, which estimates the value function to criticise the actions chosen by the actor. The critic guides the actor by evaluating its actions and providing feedback in the form of a value estimate, which can be a state-value function or an action-value function as shown in Figure 3. This feedback mechanism allows for more efficient updates to the policy, resulting in smoother and more stable learning.

Actor-Critic algorithms use gradient ascent to adjust the weights of the actor network to maximise reward and gradient descent to adjust the weights of the critic network to minimise the error in the value estimation. The actor updates its policy parameters by following the policy gradient, using the feedback from the critic to determine how to improve. Meanwhile, the critic is trained to minimise the temporal difference (TD) error, which is the difference between the estimated value and the observed return. The pseudocode for a standard Actor-Critic algorithm is presented in Algorithm 3.

Algorithm 3 Actor-Critic Algorithm

Initialise actor network $\pi(s;\theta_\pi)$ with random parameters θ_π
Initialise critic network $V(s;\theta_v)$ with random parameters θ_v
Initialise learning rates α_π and α_v for actor and critic respectively
Initialise discount factor $\gamma \in [0,1]$
for each episode **do**
  Initialise state s
  for each step in the episode **do**
    Select action a from policy $\pi(s;\theta_\pi)$
    Take action a, observe reward r and next state s'
    Compute TD error:

$$\delta = r + \gamma V(s';\theta_v) - V(s;\theta_v)$$

    Update critic network by minimising:

$$\theta_v \leftarrow \theta_v + \alpha_v \delta \nabla_{\theta_v} V(s;\theta_v)$$

    Update actor network using policy gradient:

$$\theta_\pi \leftarrow \theta_\pi + \alpha_\pi \delta \nabla_{\theta_\pi} \log \pi(a|s;\theta_\pi)$$

    Set $s \leftarrow s'$
  end for
end for
return Actor network $\pi(s;\theta_\pi)$

A simple implementation of an actor-critic algorithm can suffer from instability due to large or overly frequent policy updates which leads to inefficient learning. The Trust Region Policy Optimisation (TRPO) algorithm was developed to address these issues by introducing the idea of limiting policy updates to ensure new

policies are a more gradual improvement of the previous. Without the introduction of the 'trust region' to constrain the policy updates, the set of weights used by the actor network can jump into a new policy space, often resulting in worse performance.

Trust Region Policy Optimisation

Trust Region Policy Optimisation (TRPO) [96] is a policy optimisation algorithm designed to ensure stable and reliable policy updates of the actor network in an actor-critic DRL architecture. The core idea behind TRPO is to maximise the expected reward while ensuring that the policy update does not deviate too far from the previous policy, as measured by the Kullback-Leibler (KL) divergence [97] between the old and new policies. The KL divergence is a statistical measure that quantifies how much one probability distribution differs from another, serving as a distance metric between policy distributions. Instead of directly optimising the objective function with respect to the policy parameters, TRPO solves a constrained optimisation problem. The algorithm updates the policy iteratively by maximising the surrogate objective subject to a constraint on the KL divergence, ensuring that the policy changes are both significant and safe. By using trust regions, the algorithm is able to balance exploration and exploitation effectively, avoiding the erratic behaviour that can affect other DRL techniques. The pseudocode for TRPO is provided in Algorithm 4.

Algorithm 4 Trust Region Policy Optimisation (TRPO)

1: Initialise policy network $\pi(s;\theta)$ with random parameters θ

2: Initialise value function network $V(s;\phi)$ with random parameters ϕ

3: Initialise learning rates α_π and α_v, and discount factor γ, and KL divergence constraint δ

4: **for** each iteration **do**

5: Collect a set of trajectories $\{\tau_i\}$ by running the current policy $\pi(s;\theta)$ in the environment

6: Estimate cumulative rewards $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ for each time step in each trajectory

7: Estimate the advantage function for each time step:

$$A_t = R_t - V(s_t;\phi)$$

8: Update value function by minimising:

$$\phi \leftarrow \phi - \alpha_v \nabla_\phi \sum_t \left(V(s_t;\phi) - R_t\right)^2$$

9: Compute the policy gradient:

$$\nabla_\theta L(\theta) = \mathbb{E}_t \left[\nabla_\theta \log \pi(a_t|s_t;\theta) A_t\right]$$

10: Use conjugate gradient algorithm to compute the step direction p that solves:

$$\nabla^2_\theta L(\theta) p = -\nabla_\theta L(\theta)$$

11: Compute step size α via a line search that ensures the KL-divergence constraint is satisfied:

$$\mathbb{E}_t \left[D_{\mathrm{KL}} \left[\pi_{\theta_{\mathrm{new}}}(s) \| \pi_\theta(s)\right]\right] \leq \delta$$

12: Update policy parameters:

$$\theta \leftarrow \theta + \alpha p$$

13: **end for**

14: **return** Policy network $\pi(s;\theta)$

TRPO addresses most of the issues faced by the DRL algorithms that came before it, but in doing so the computational complexity associated with training the agent increases significantly due to the the calculations involved with constraining the policy weight updates. Proximal Policy Optimisation (PPO) refined this approach by using a simpler method that clips policy updates to ensure they still remain within a safe range but also offering a balance between stability and computational efficiency.

Proximal Policy Optimisation

Proximal Policy Optimisation (PPO) [98] is a widely used policy optimisation algorithm in reinforcement learning (RL) that addresses some of the limitations of earlier methods, such as Trust Region Policy Optimisation (TRPO). PPO achieves this by introducing a clipped surrogate objective function, which allows for more flexible updates while still preventing excessively large policy changes. The key innovation in PPO is the clipping mechanism applied to the probability ratio between the new policy and the old policy. This clipping mechanism modifies the objective function so that, if the probability ratio deviates significantly from 1.0 (indicating a large policy update), the update is clipped. This prevents harmful updates that could destabilise learning, similar to how TRPO uses a trust region, but in a simpler and more computationally efficient way. The pseudocode of the PPO algorithm is presented in Algorithm 5.

Algorithm 5 Proximal Policy Optimisation (PPO)

Initialise policy network $\pi(s;\theta)$ and value function network $V(s;\phi)$ with random parameters

Initialise learning rates α_π and α_v, and discount factor γ

Set clipping parameter ϵ and number of epochs K

for each iteration **do**

Collect a set of trajectories $\{\tau_i\}$ by running the current policy $\pi(s;\theta)$

Estimate rewards-to-go $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ for each time step in each trajectory

Estimate the advantage function for each time step:

$$A_t = R_t - V(s_t;\phi)$$

Update value function by minimising:

$$\phi \leftarrow \phi - \alpha_v \nabla_\phi \sum_t \left(V(s_t;\phi) - R_t\right)^2$$

for each epoch $k = 1, \ldots, K$ **do**

Compute the probability ratio:

$$r_t(\theta) = \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_{\text{old}})}$$

Compute the clipped objective:

$$L(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right]$$

Update policy network by maximising the objective:

$$\theta \leftarrow \theta + \alpha_\pi \nabla_\theta L(\theta)$$

end for

end for

return Policy network $\pi(s;\theta)$

The deep reinforcement learning algorithms described in this section are all used in the following contribution chapters in varying quantities. PPO has demonstrated impressive levels of performance across a variety of tasks for a relatively inexpensive computational cost, so is the algorithm of choice for both FDRL and PADRL contributions. For the SADRL contribution, a more holistic approach is taken to investigate the difference in performance between Deep Q Networks, Actor-Critic Methods, Proximal Policy Optimisation and Trust Region Policy Optimisation.

3.3 Applications of Machine Learning to Trading

Machine learning has evolved significantly since its discovery as a powerful tool for identifying patterns in data. Its potential for predicting financial markets, including the Foreign Exchange market has piqued the interest of researchers, who are keen to uncover hidden relationships between various market variables and future price movements. The primary goal of such efforts is to enhance the profitability of trading strategies through the accurate forecasting of currency price trends. The following section is organised into four separate parts, with each part considering the application of a different style of machine learning to trading: shallow supervised learning, evolutionary algorithms, deep supervised learning and reinforcement learning. To gain a more complete understanding of the existing literature, each part will discuss the application of the machine learning algorithm to price data, sampled both with and without the DC sampling algorithm across a number of different asset classes. By including work using non-DC approaches and non-FX asset classes, existing successful techniques can be uncovered that could potentially experience enhanced success under the DC paradigm.

Shallow Supervised Learning Shallow supervised learning refers to the subset of machine learning methods that do not use deep neural networks to predict a single or set of target variables from historical training data. This includes but

is not limited to methods such as Linear Regression, Support Vector Machines (SVMs) and Decision Trees. These models, while less computationally complex than modern deep learning architectures, have been extensively studied for their predictive capabilities in financial markets.

Several studies have explored various linear regression models for predicting price movements of a number of different assets. [99] introduced a LASSO linear regression model to predict stock market behaviour, emphasising the model's ability to handle sparse solutions efficiently. Their experiments with Goldman Sachs stock demonstrated the LASSO model's superiority over ridge regression in forecasting stock prices. [100] developed a linear regression model for predicting commodity prices, specifically focusing on China's cotton market. They improved accuracy over the China Cotton Association's price table by using a multiple linear regression model but acknowledged challenges with quality-price mismatches in their predictions. [101] applied both linear regression and decision tree regression to stock market analysis. They evaluated the models' accuracy in forecasting stock prices using different dataset sizes, concluding that linear regression is the superior approach in this case. Together these works showcase the importance of linear regression techniques in financial prediction and decision-making but given the existence of other more advanced ML regression there is plenty of scope for improvement.

The integration of support vector regression (SVR) with novel optimisation techniques has shown promising results in the application of machine learning to trading. In [102], a correlation-aided support vector regression (cSVR) method is proposed to enhance the prediction of Forex exchange rates. By leveraging graphical channel correlation analysis along with parametrised Pearson's correlation to filter noise, the model successfully improves the accuracy of time series predictions for various currency pairs over the period of 2007-2008. This method demonstrates the benefits of capturing correlation patterns in financial data to improve SVR's predictive capabilities. [103] introduce the Krill Herd - Support

Vector Regression (KH-vSVR) model, which employs the Krill Herd algorithm to optimise SVR parameters by navigating between local and global optima. Applied to commodity Exchange Traded Funds (ETFs), the KH-vSVR demonstrates superior performance in terms of both statistical accuracy and trading efficiency when compared to conventional SVR models.

The application of decision tree based approaches to trading have focused on improving both short-term and long-term forecasting through various machine learning techniques. [104] explores the use of decision trees for stock trend prediction, highlighting the benefits of technical indicators such as MACD, RSI, Stochastic Oscillator and Bollinger Bands for short-term forecasts. The study reveals that short-term trends rely heavily on technical indicators, but long-term stock movement prediction requires a combination of technical factors and fundamental data. [105] examine the application of the Extreme Gradient Boosting (XGBoost) algorithm to forecast the exchange rate of the US dollar against the Indonesian rupiah. The authors developed a model that effectively reduces forecasting errors, claiming that with hyperparameter tuning the model achieved low RMSE and MAPE scores, demonstrating the effectiveness of XGBoost in time series forecasting.

[106] conducted a comparative analysis of classic regression methods, SVMs and neural networks to predict directional price movements. Their findings revealed that ridge regression, a type of linear regression, outperformed both SVMs and neural networks in certain market conditions, particularly when predicting upward or downward price movements. This aligns with earlier studies which suggested that while machine learning models can outperform traditional econometric models, their performance is highly dependent on the specific characteristics of the data being analysed.

Other work such as [107], [108] and [109] analysed how well machine learning can predict price changes in the FX market, focusing primarily on currency pairs involving the US Dollar and the Euro. All papers found that machine learning

methods provide significant market edge, citing support vector regression and ensemble methods as the most effective for the regression tasks undertaken. [110] came to a similar conclusion when analysing a similar prediction problem on US stock market data. [111] and [112] put more of a focus on feature engineering as they cite Moving Averages, Relative Strength Index and Bollinger Bands as particularly helpful technical indicators when predicting price movements on numerous asset classes.

[113] explores the use of ML to trade financial markets using dynamic threshold breakout labelling, a technique somewhat tangential to DC sampling due to it's ability to focus on significant price moves as a way of strengthening trading signals. The results show that machine learning has a positive impact on identifying price breakouts in the US stock market, commodities market and the cryptocurrency market. [114] tackles intraday trading using ML with a focus on India's Nifty 50 stocks. This work cites the shallow ML methods as playing a pivotal role in predicting the stock prices of the component companies of the Nifty 50, therefore helping to optimise trading decisions.

The application of shallow supervised learning methods to trading have demonstrated the ability of machine learning algorithms to recognise patterns from financial data and be used as a component of a trading algorithm. These methods have also been shown to be effective over numerous different asset classes. By identifying algorithms that work across multiple asset classes a hypothesis can begin to be formed around whether these algorithms are effective specifically for foreign exchange data, the asset class used in this thesis. As well as considering the asset class it is also possible to begin to speculate on the efficacy of the algorithms on DC sampled data. The remainder of this section reviews related works investigating more advanced machine learning algorithms, including evolutionary algorithms, deep learning algorithms and reinforcement learning algorithms.

Evolutionary Algorithms A large part of the existing literature that uses DC sampling to prepare the data upon which trades are executed consists of the

application of an evolutionary algorithm, hence this segment is solely focussed on the applications of evolutionary algorithms to DC sampled data. Although this thesis seeks to apply different machine learning methods to DC sampled FX data, by reviewing the application of the evolutionary methods, the intention is to uncover the techniques used to prepare the data for training that support related research using alternative learning algorithms.

Directional Changes (DC) is a powerful event-based sampling method that has gained attention for improving trading strategies by shifting away from traditional physical time scales. Some early works that combine DC sampling and genetic programming techniques include [115] which uses genetic programming to generate DC based trading strategies and [116] which uses genetic programming to develop a regression approach to predict the end of DC trends in the FX market, with both approaches demonstrating promising levels of success. [117] introduced an innovative approach that leverages DC, combined with a genetic algorithm, to evolve profitable trading strategies. This work focused on summarising market data through DC events, allowing for the identification of significant market activities. Experiments on foreign exchange market data demonstrated that their method outperformed traditional trading strategies such as technical analysis and buy-and-hold approaches.

[118] and [119] used genetic programming to predict trend reversals of DC trends, with the latter of the two using classification techniques to enhance profitability. [120] advanced this concept by integrating the classification and regression techniques into multiple DC-based strategies. This work tested the algorithm across 20 foreign exchange markets, showing a significant performance boost in terms of return and risk over several benchmarks, including both single DC-based strategies and traditional trading strategies. These studies both provide evidence for the effectiveness of DC as a basis for constructing robust trading strategies in algorithmic trading.

[121] proposed the optimisation of DC-based trading strategies using genetic

algorithms (GA). This approach, which introduced four novel DC-based strategies and optimised them using GA, outperformed individual DC strategies as well as traditional buy-and-hold benchmarks in experiments across 44 stocks. This approach was developed further in [122] by introducing a multi-threshold DC model, further refining trading strategies by using weighted thresholds. This method demonstrated superior performance against single-threshold strategies and traditional technical indicators such as MACD and RSI.

Works such as [123] and [124] also use genetic programming to develop effective strategies for trading under the DC sampling paradigm. These studies demonstrate the effectiveness of incorporating DC and evolutionary algorithms into trading strategies, consistently performing well across different market conditions. [125] and [126] expanded this work by integrating multi-objective optimisation with genetic programming (GP), combining both DC and traditional technical indicators. This research found that a multi-objective GP significantly improved cumulative returns compared to a single-objective GP, with a substantial increase in return performance, while also outperforming the buy-and-hold strategy.

Other work that applies evolutionary algorithms to trading but without the directional changes sampling algorithm includes [127] which takes a unique approach by ensembling ML methods with genetic algorithms, also producing an effective trading system on FX data. [128] also uses genetic algorithms along with technical analysis to effectively trade stock data.

The body of literature applying evolutionary algorithms to trading DC sampled data demonstrates that DC sampling is an effective sampling algorithm upon which to build automated trading strategies. This would suggest that the replacement of these evolutionary algorithms with a different machine learning algorithm could also yield favourable results, this theory forms the basis of Chapters 4, 5 and 6.

Deep Supervised Learning The recent advancements in the field of deep learning has revolutionised the field of financial forecasting, with many studies exploring the use of deep neural networks (DNNs), recurrent neural networks (RNNs) and

long short-term memory (LSTM) networks to capture the non-linear and temporal dynamics of financial markets. The purpose of this section is to build up an idea of the most useful deep learning methods for trading. Once an internal taxonomy of the most effective methods used in the literature has been developed, a deeper understanding of the specifics of each method can be gained in Chapter 3.

[129] addresses the complexities of intra-day market forecasting, proposing a decision-making framework that integrates deep neural networks with genetic algorithms. Their findings, which demonstrate a prediction accuracy of 72.5% and an annualised net return of 23.3%, underscore the effectiveness of machine learning techniques in enhancing HFT performance. [130] developed a deep learning framework called FLF-LSTM for predicting Forex price trends. Their model introduced a Forex Loss Function (FLF), specifically designed to optimise performance in the highly volatile currency market. By incorporating this loss function, the FLF-LSTM model achieved superior predictive accuracy compared to traditional loss functions. [131] made significant strides in applying deep learning to emerging market currencies. Their research demonstrated that deep neural networks outperformed traditional models in forecasting the highly volatile and less liquid currencies of emerging markets. Another critical study [132] introduced a hybrid model combining gated recurrent units (GRU) and LSTMs to tackle the challenges of forecasting complex and volatile time series data. Their hybrid network architecture used the strengths of both GRU and LSTM cell architectures, leading to more accurate predictions of exchange rate movements than models using either GRU or LSTM in isolation.

[133] conducted a comprehensive comparison of various RNN architectures, including LSTM networks, gated recurrent units (GRU) and traditional RNNs, to evaluate their performance in predicting FX rates. The study found that while deep networks can offer strong predictive capabilities, simpler neural network models can often perform equally well, especially in terms of trading profitability. [134] also explored the effectiveness of the LSTM architecture in FX trading, com-

paring it against ARIMA and Support Vector Regression models for USD/ZAR exchange rate forecasting. Their findings indicated that LSTM outperformed SVR and ARIMA in terms of Mean Squared Error (MSE), although ARIMA performed better on Mean Absolute Error (MAE). Expanding on LSTM applications, [135] introduced a two-layer stacked LSTM (TLS-LSTM) model to enhance forecasting accuracy, particularly for AUD/USD exchange rates. Their model showed superior performance over single-layer LSTM, multilayer perceptron (MLP) and other advanced techniques like CEEMDAN-IFALSTM. Their correlation analysis also revealed significant relationships between AUD/USD and other currency pairs, such as EUR/AUD and AUD/JPY, showing the connected nature of FX currency pairs.

[136], [137] and [138] investigate the use of deep supervised learning for high frequency trading on a variety of cryptocurrency and traditional currency markets. All three propose deep learning frameworks that produce high levels of accuracy when predicting changes in price of the respective asset classes, further supporting the use of deep learning on market data. [139] and [140] investigate the use of sequence based neural network architectures, proposing that there are benefits to using LSTM, GRU and transformer based architectures on high frequency trading data. [141] takes the unique approach of predicting bid-ask spread, using XGBoost for feature selection and deep learning methods for prediction. This work found that this collaboration between the two methods produces some favourable results when tested on high frequency data from the Taiwan stock exchange.

Deep learning methods are most commonly cited for their application to fields like natural language processing and image classification. By analysing the literature involving the application of these same deep learning methods to trading, an understanding can be built on how to apply these deep learning methods that have achieved success in other applications to the trading task addressed in this thesis. Deep learning offers significantly greater potential than the shallow machine learning methods discussed, making the supporting evidence of its effectiveness

particularly valuable.

Reinforcement Learning Reinforcement learning (RL) can be particularly effective in the FX market, as the agent's objective is to develop trading strategies that maximise rewards (or more specifically returns) over time. The field of RL however is a relatively underdeveloped subset of machine learning compared to other areas such as supervised and unsupervised learning, but stands to offer some of the most promising qualities for real world applications. A 2019 analysis [142] of the literature involving the application of RL in the financial markets and states that, although results look promising, there are still some holes within the literature that do not bridge the gap between theory and practicality. Lack of transaction costs and slippage assumptions are the root cause of this gap.

Early work in 1998 [143] laid the groundwork for RL in financial trading, establishing the importance of careful design of the environment, including the action space, state space and reward function. The design of the reward function, in particular, is critical in RL, as it directly influences the agent's trading decisions. This foundational approach has remained consistent in RL research, with more recent studies, such as [144] in 2019, continuing to emphasise the importance of environment design to achieve promising results in FX trading. The integration of deep reinforcement learning has further advanced the field, allowing for more complex and nuanced decision-making. [145] for example, developed an RL-based FX trading model that learned optimal strategies by interacting with the market environment over time. This model's ability to dynamically adapt its strategy in response to changing market conditions demonstrates the potential of RL in FX trading, particularly in environments where market conditions are constantly shifting. A 2019 study [146] incorporated deep LSTM networks into a reinforcement learning framework, specifically designed for high-frequency trading. This model enabled the RL agent to make real-time decisions based on the market's historical data, effectively learning to adjust its strategies on the fly. The combination of RL and deep learning allowed the system to handle the fast-paced, highly volatile

nature of foreign exchange markets with notable success.

[147] further explored the potential of RL by comparing Deep Q Networks (DQN) and Proximal Policy Optimisation (PPO). By encoding price data using Gramian Angular Fields (GAF), they demonstrated the feasibility of RL in handling complex currency pairs and achieving favourable returns. [148] extended RL's application to stock market trading, formulating the problem as a Markov Decision Process (MDP) and testing on-policy RL algorithms such as Vanilla Policy Gradient (VPG), TRPO and PPO on stocks like Apple and Nike. Their results demonstrated the ability of RL to maximise profits through adaptive learning based on rewards from trading actions. [149] proposed a Q-learning-based agent for dynamic equity trading, focusing on representing the environment's state and outperforming traditional Buy-and-Hold strategies in both Indian and American markets. [150] introduced an end-to-end trade execution framework using PPO, leveraging limit order book (LOB) data and LSTM networks to optimise trade execution without manually designed attributes. Their model surpassed industry-standard strategies like TWAP and VWAP, showcasing the strength of RL in trade execution.

Although limited, there exist some studies into the application of RL to DC sampled data. [151] proposed a dynamic algorithmic trading strategy known as the DCRL trading strategy, which leverages the DC sampling approach combined with RL. This approach dynamically adjusts to the price time-series patterns by using Q-learning to optimise trading decisions, resulting in robust performance when evaluated on stock market indices such as the S&P500, NASDAQ and Dow Jones. Their results indicate improved trading returns and Sharpe Ratios, particularly in volatile markets. In a related study [152] introduced an adaptive pattern recognition model also based on the DC event approach and RL. This model departs from traditional fixed-interval analytical methods by employing an event-based time interval to better capture market behaviours. Through reinforcement learning, the DCRL model effectively identifies directional price changes, with strong

performance demonstrated using Saudi stock market data. Both studies highlight the potential of RL combined with DC in enhancing both trading and pattern recognition in financial time-series data.

[153], [154] and [155] all employ Deep Q Networks for trading different asset classes in different ways, ranging from statistical arbitrage to commodity futures, with the first two works demonstrating effective strategies in high frequency environments and all three demonstrating positive results. [156] and [157] implement PPO with [156] combining this with a convolutional neural network and short-term memory networks to enable the processing of time-series data. [157] uses LSTM networks in a multi-agent system to produce a profitable strategy on the EUR/USD currency pair. [156] produces similarly positive results for it's approach on high frequency NASDAQ data.

[158] and [159] both use a system of hierarchical DRL agents that have been trained to trade in high frequency environments, with both methods yielding impressive results in the cryptocurrency market. [160] focusses more on the internal neural network structure as opposed to the architecture of the agents and opts for implementing a transformer based model. This approach also yields impressive results but on the US stock market as opposed to the cryptocurrency market.

[161] and [162] both employ actor critic methods that investigate the ability of these deep reinforcement learning algorithms to trade the stock market and cryptocurrencies respectively, with the latter focusing on high frequency data. These works both demonstrate how useful DRL can be for trading but highlight the sticking points of training stability when considering DRL methods, proposing that future work should be steered more towards this area. The development of the platform in [163] demonstrates the interest in growing the area of deep reinforcement learning. The developed platform standardises the approach taken to train the required agents and the necessity to define state and action spaces as well as reward functions when training these agents which will likely lead to much more future progress in this space.

Reinforcement learning has a significant body of work exposing its application to trading in general, but only two of these approaches seem to consider the application of RL to DC sampled data. These two works [151,152] do not consider deep reinforcement learning, as is used in many of the other successful approaches. By investigating the use of deep reinforcement learning with FX data sampled under the DC sampling paradigm, a significant gap in the literature would be filled as it is a logical continuation of the research space.

3.4 Summary

This chapter provided the background information and literature for the application of machine learning techniques to financial data used in this thesis. In Section 3.1 supervised learning techniques that form the basis of deep supervised learning were discussed. These supervised learning techniques also included the deep learning methods that are a key component of the deep reinforcement learning methods discussed in Section 3.2, which themselves detail the techniques used in Chapters 4, 5 and 6. Finally, Section 3.3 discuses how machine learning algorithms are used to train financial models in related works.

The literature review in Section 3.3 analysed how increasingly more complex machine learning methods have been used for trading financial assets, with a bias towards FX data as that is the focus of this thesis. It was found that shallow supervised learning methods are effective for predicting price movement but can be enhanced with deep learning methods. The literature also reveals that evolutionary algorithms are very effective when dealing with directional change sampled data, which would lead us to believe other more advanced machine learning algorithms could also perform well using DC sampled data. Finally, the application of reinforcement learning to trading was analysed and it was discovered that there are a number of studies that have used RL to trade, but often face issues with making the results realistic. A gap in the literature was also identified where deep reinforcement learning and DC sampled data intersect as there are currently only

two works outside of this thesis that apply RL to DC sampled data and they are both on stock data and do not use deep learning methods with RL. The following three chapters consist of the thesis contributions, over the course of these three chapters increasingly more complex and realistic trading algorithms using DC sampling and machine learning are developed with the intention of profitably trading in high frequency trading FX markets.

Chapter 4

Positionally-Naive Deep Reinforcement Learning Trading

4.1 Motivation

In this chapter, a novel trading agent framework FDRL (Filtered Deep Reinforcement Learning) is introduced, which uses the deep reinforcement learning (DRL) technique of PPO described in Section 3.2.4 to optimise the trading decisions of a set of trading agents in high-frequency FX environments. Unlike traditional ML-based methods that rely on adding trading rules to price move predictions made by supervised learning models, FDRL takes an agentic approach. This allows agents to learn and adapt their actions based on the evolving state of the market and receive rewards directly related to the outcomes of their actions.

The FDRL framework leverages the Proximal Policy Optimisation (PPO) algorithm, due to its inherent stability and training efficiency, to train the agents in a simulated market environment. This environment was designed to provide signals based on data sampled using the directional changes (DC) framework. By attempting to learn the optimal trading policy within this environment, the agents can make decisions directly related to trading performance by focusing on the steps required to maximise returns rather than predicting precise price points

which would indirectly influence trading behaviour through a rule-based wrapper.

By comparing FDRL results to technical analysis and directional change benchmarks, it is shown that the combination of DRL and DC sampling offers a more robust and adaptive trading solution at a fixed transaction cost of 0.025% of the position size. The ability of the resultant FDRL agents to extract valuable information from the DC framework allows it to outperform strategies that do not use deep reinforcement learning or DC-based sampling.

The rest of this chapter is organised as follows. Section 4.2 explains the methodology used to conduct the experiments, Section 4.3 then discusses the experimental setup. The results are then presented in Section 4.4 and interpreted in Section 4.5. Section 4.6 then summarises the findings of the study.

4.2 Methodology

An overview of the FDRL methodology is presented in Figure 4.1 that shows the flow of data through the methodology pipeline. The first step shows the splitting of FX tick data for a given currency pair into windows which was then sampled and enriched using the DC sampling algorithm, before each window was split further into the required training, validation and test sets. The training and validation sets of each window were then ready to be used to train and optimise the agents before being retrained using the concatenated training and validation sets, the output of which was assessed and reported using the test set.

In the first methodology section, the data preparation required for training a DRL agent per window is discussed. This involved manipulating the tick data to provide cleaner trading signals and designing the environment, action space, state space and reward function in such a way that the DRL agents were able to learn an effective trading policy. The training and hyperparameter optimisation process is then discussed, before outlining the trade filtering alteration made after discovering trading characteristics on the validation set that had to be made in order to limit losses on loss-making windows.

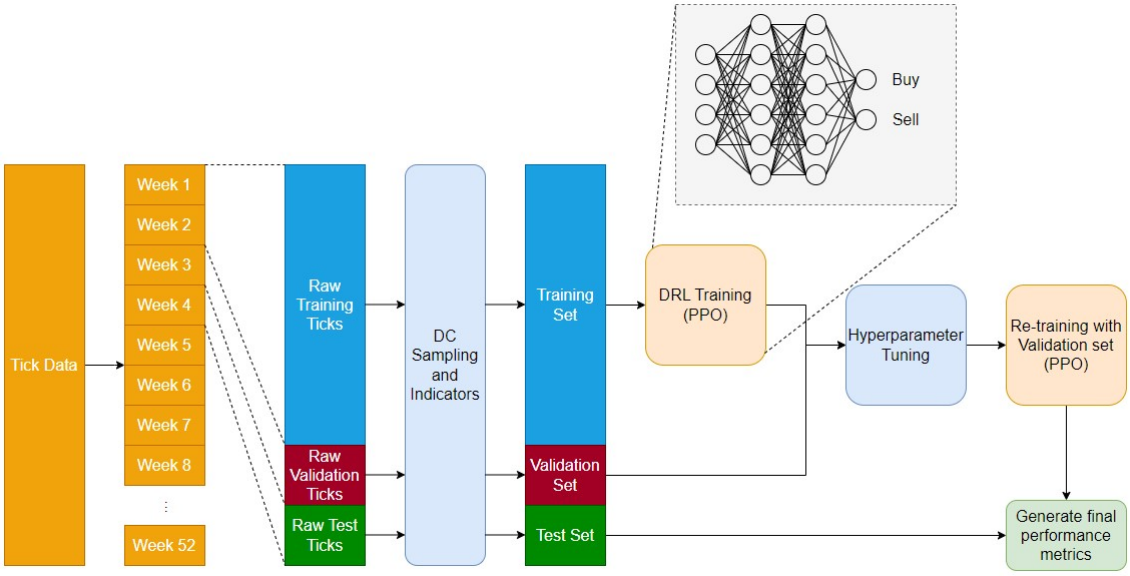

Figure 4.1: Experiment Methodology (see Section 4.3.2 for real network architecture)

4.2.1 Training Preparation

Data Preparation The raw tick data went through a number of data preparation steps before it was ready to be used for training. The first of these steps was to split the tick data into weekly segments to create a contiguous array of weeks over the duration of a year (see Section 4.3 for more details). A rolling window was then applied to the array of weeks to create 4-week windows that consist of a 2-week training period, a 1-week validation period and a 1-week testing period of raw ticks. Each window of raw ticks was then sampled using the DC sampling algorithm and the DC based indicators listed in Table 4.1 were calculated to generate the features for the training, validation and test sets. The prepared data was then used to simulate historical market behaviour and therefore created a training environment for the DRL agents to learn in.

Environment Development Once the data had been prepared, each individual agent underwent the same training process per window, involving the same environment definition, so the only difference in setup from one agent to another was the training data. The experiment was set up like this with the prior understanding that the market experiences different regimes that cause price behaviour

Indicator	Description	Equation	Period
$\boldsymbol{TMV}$	Ratio of whole price move to threshold	$\frac{\lvert\Delta price\rvert}{\theta}$	1
$\boldsymbol{OSV}$	Percentage change between current DCC and previous DCC normalised by threshold	$\frac{\left(\frac{DCC_t - DCC_{t-1}}{DCC_{t-1}}\right)}{\theta}$	(3, 5, 10)*
$\boldsymbol{R_{DC}}$	Number of ticks adjusted by the return of the event	$\frac{(TMV * \theta)}{\Delta Event_t}$	(3, 5, 10)*
$\boldsymbol{T_{DC}}$	Number of ticks over the course of the event	$\Delta Event_{no.ticks}$	(3, 5, 10)*
$\boldsymbol{N_{DC}}$	Number of ticks over a certain number of events	$\sum_{i=0}^{n} Event_{no.ticks_i}$	(1, 10, 20, 30, 40, 50)
$\boldsymbol{C_{DC}}$	Sum of $\lvert TMV\rvert$ over a certain number of events	$\sum_{i=0}^{n} \lvert TMV\rvert_i$	(1, 10, 20, 30, 40, 50)
$\boldsymbol{A_T}$	Difference between the number of ticks spent on an up and down trend over n events	$\sum_{i=0}^{n} UpEvent_{no.ticks_i} - \sum_{i=0}^{n} DownEvent_{no.ticks_i}$	(1, 10, 20, 30, 40, 50)

Table 4.1: DC Indicators
Where: θ is DC threshold and DCC is DC confirmation point (Periods marked with a * use a moving average of the indicator)

to change over a period of time [36]. By allowing different agents to learn based on the recent window of preceding data, the price dynamics (regime) of their training data was more likely to reflect the regime of the price data they were tested on. The environment was defined as a custom environment in the `Gym` [164] Python library so that it was compatible with the appropriate reinforcement learning and deep learning libraries.

Action Space The action space of the agent refers to the set of possible actions the agent was able to take at each time step. The action space chosen for this experiment was discrete, as the agent was only allowed to either buy or sell. This action space was selected after preliminary testing on a smaller subset of validation data, which consisted of providing the agent with three possible actions of buy, sell and hold. This resulted in the agent learning policies that consisted of little to no trading activity as the agent learned that buy or sell actions would result in losses due to the immediate transaction costs and therefore would fall into a policy space of just holding its first position and never executing any further trades. To counter this behaviour the option to hold a trade was removed, forcing the agent to make a decision between buy and sell. The agent was still able to hold its position by suggesting the same action as the current position, but reducing the option to two actions instead of three provoked much more active trading while still allowing for transaction costs and coaxed the agent into learning a more profitable strategy.

State Space The state space refers to the representation of the environment the agent observed at each time step. The state space was generated from the data that was prepared to represent the real FX market at a given point in time, therefore the agent received the DC data as a representation of market state and used this to make decisions. A fine balance between information rich data and lack of noise in the environment representation had to be struck as the agent needed to be provided with as much data as possible so it was able to learn an effective policy, but also to not have confused the agent with misleading signals as a result of

noise. The state at each time step was represented by a preceding window of price data and relevant features. The selection of the appropriate size for the previous window of time steps (also known as the state space lag) that was provided as input to the agent represents another hyperparameter that required tuning (see Section 4.3.2 for details).

More formally, the state space $\mathcal{S}$ represents the set of all possible observations the agent can receive from the environment. In the context of this trading system, each state $s_t \in \mathcal{S}$ is a tensor consisting of market features over a temporal window. Specifically, the state at time t is constructed as:

$$s_t = [f_{t-\ell}, f_{t-\ell+1}, \ldots, f_{t-1}, f_t] \in \mathbb{R}^{d \times \ell} \tag{4.1}$$

where $f_t \in \mathbb{R}^d$ represents the feature vector at time step t with dimensionality $d = 30$ (comprising the DC-based indicators defined in Table 4.1), and $\ell = 5$ is the state space lag (the lookback window size). The complete state representation is therefore a 150-dimensional vector that is flattened before being passed as input to the policy and value networks.

Each feature vector f_t contains 30 elements calculated from the DC-sampled data: 1 TMV value, 3 OSV values (one for each period: 3, 5, 10), 3 R_{DC} values, 3 T_{DC} values, 6 N_{DC} values (one for each period: 1, 10, 20, 30, 40, 50), 6 C_{DC} values, 6 A_T values, plus the DC event start and end prices (2 values). This results in $d = 1 + 3 + 3 + 3 + 6 + 6 + 6 + 2 = 30$ features per time step. When combined with a temporal lag of $\ell = 5$ time steps, the resulting state space dimensionality is $d \times \ell = 30 \times 5 = 150$.

The dimensionality of the state space is crucial for the agent's learning capability. A state space that is too small may lack sufficient information for the agent to distinguish between different market conditions, whilst an excessively large state space can suffer from the curse of dimensionality, making it difficult for the neural network to generalise effectively. The choice of features and temporal window length directly determines the state space dimensionality, which in turn affects

both the computational requirements and the agent's ability to learn an effective policy. The use of deep neural networks in this framework allows the agent to handle this relatively high-dimensional continuous state space, which would be intractable for traditional tabular RL methods like Q-learning.

Reward Function A reward function was used to provide feedback to the agent at each time step to help guide its training and was crafted to elicit the correct behaviour from the final trained agent. Each agent was provided with a fixed starting balance of 100 units of the base currency for each pair and was rewarded with the profit of a trade once the position was closed. If the trade was not closed then the agent received a reward of zero for each time step. This reward function then created a balance between the incentive to trade frequently in order to realise a larger number of positive returns but also to identify effective trading opportunities that did not result in losses.

4.2.2 Agent Training

Once the environment and all its components had been designed, the agent was ready to be trained. The training process involved iterating over each step of the environment to generate an action from the agent, the new environment state of this action and the reward was then fed back into the agent and the cycle continued until a terminal condition was met. The duration of this training process was defined by the number of time steps, a hyperparameter that was passed to the training algorithm. The strategy applied to the tuning of the hyperparameter is defined in Section 4.3.2. The number of time steps defined how many times the algorithm iterated through each time step of the data. The implementation of the deep reinforcement learning algorithm was largely abstracted by the `Stable Baselines 3` (SB3) library [165] with a `PyTorch` [1] backend. The SB3 library integrates well with the `Gym` library which was used to design the environment and was able to automatically step through this process and apply the required

[1] https://pytorch.org/

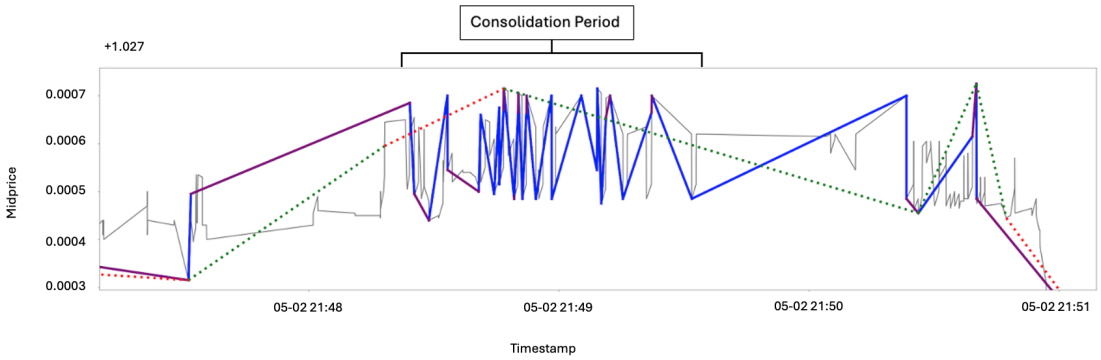

Figure 4.2: Consolidation Period when sampled at 0.015% but not when sampled at 0.029% showing how different DC thresholds provide different outlooks on the data.

updates to the weights of the deep neural networks representing the value and policy network of the DRL agent. This whole process returned a trained model per window, the trained model then used the validation set to identify the correct set of hyperparameters.

4.2.3 Trade Filtering

Analysis of the trading strategies learned by the DRL agents on validation sets across a subset of currency pairs and windows revealed the consistent pattern that agents predominantly learned to execute trades during periods of low volatility, termed 'consolidation periods' in this thesis. These consolidation periods are characterised by successive DC trends exhibiting minimal or absent overshoot movements, as illustrated by the solid lines representing 0.015% DC moves in Figure 4.2.

The identification of consolidation periods is threshold dependent. This can be observed in Figure 4.2, where the dotted 0.029% threshold lines do not capture a consolidation period for the same underlying tick data, demonstrating how different DC thresholds provide distinct perspectives on market behaviour. This threshold sensitivity arises because larger thresholds require greater price movements to trigger DC events, potentially filtering out the smaller oscillations that define consolidation periods at lower thresholds.

Preliminary testing indicated that whilst agents occasionally identified these periods autonomously, their performance was inconsistent. When successful identification occurred, agents exhibited distinctive trading behaviour. This involved rapid opening and closing of positions across successive DC trends, with this pattern repeating until the consolidation period concluded. However, the agents struggled to reliably distinguish consolidation periods from other market conditions without additional structure. This limitation meant that agents would sometimes attempt similar rapid trading strategies during non-consolidation periods, leading to poor performance.

To address this challenge, a filtering mechanism was introduced in the testing environment to explicitly identify and restrict trading to consolidation periods. The filter operates by monitoring DC trends for the absence of overshoot events, permitting trading only after a specified number of consecutive no-overshoot trends have been observed. This approach effectively acts as a gating function, enabling the agent to trade when market conditions match the consolidation pattern it has learned to exploit, whilst preventing trading during unsuitable market regimes.

The threshold parameter for this filter (specifically the number of preceding consecutive no-overshoot trends required to activate trading) was determined through systematic hyperparameter optimisation. A grid search was conducted on the validation sets, testing values of 2, 5, 10, and 20 successive no-overshoot trends (as detailed in Section 4.3.2). This search identified 5 consecutive no-overshoot trends as the optimal trigger value, yielding the highest total return across the subset of validation sets evaluated. This value represents a balance where fewer required trends (e.g. 2) resulted in premature trading activation during brief periods resembling consolidation, whilst more conservative thresholds (e.g. 20) delayed trading activation and reduced the number of exploitable consolidation periods. The resultant trading behaviour emerging from this filtered approach is examined in greater detail in Section 4.5, where the final results and their implications are interpreted.

4.2.4 Performance Metrics

Once the agents were trained on the concatenated training and validation sets, their performance was tested using a number of metrics. Marginal return (see Equation 4.2) was used to measure each individual trade made by the agent in a specific window. The successive accumulation of the marginal returns results in the total return (see Equation 4.3) which was used to measure the overall success of the system of agents. Maximum Drawdown (see Equation 4.4) was used as a measure of risk as this represents the largest loss that had to be incurred in order to produce the final total return result. Total return and maximum drawdown give a breakdown of the realised upside and downside associated with this performance, the Calmar ratio was then used to represent both these performance metrics in a single aggregated value and therefore provided a measure of risk adjusted return.

$$MR = \pm\%\Delta p * P_{size} \tag{4.2}$$

where: MR is marginal return, p is price, sign is changed to reflect profit and loss, and P_{size} is the position size, this represents the quantity of the currency pair being traded for a specific position.

$$R = \sum_{i=0}^{no.trades} MR_i \tag{4.3}$$

where: R is total return, and MR_i is the marginal return defined in Equation 4.2 for trade i.

$$MDD = \frac{\rho - \tau}{\rho} \tag{4.4}$$

where: MDD is maximum drawdown, ρ is the peak balance before largest drop, this refers to the largest balance observed at the peak of the largest drop in the balance, and τ is next lowest balance before a new high, this refers to the lowest balance observed after the peak balance before the balance rises above the peak value.

$$CalmarRatio = \frac{R}{MDD} \tag{4.5}$$

4.3 Experimental Setup

4.3.1 Data

The tick data for fourteen currency pairs was downloaded from TrueFX.com[2]. Data for currency pairs AUD/JPY, CAD/JPY, CHF/JPY, EUR/CHF, EUR/GBP, EUR/JPY, GBP/JPY were taken from the period 01/05/2022 to 30/04/2023, the remaining seven currency pairs all include USD (AUD/USD, EUR/USD, GBP/USD, NZD/USD, USD/CAD, USD/CHF and USD/JPY) and were sampled from the period of 01/01/2022 to 31/12/2022[3]. The size of each rolling window was 4 weeks of physical time with a shift of 1 week from one window to the next. These size and shift values were selected based on the number of resultant DC trends that could be obtained based on the threshold since the larger the threshold, the fewer DC trends there are. It was ensured that a number of DC trends were present in each set of the data to not only account for the leading values of the indicators but also to provide the DRL agent with enough data to learn from and generalise correctly. The DC sampling method was applied to the midpoint of the bid and ask prices obtained from the raw data. This sampling method was applied independently to the training, validation and testing set per window, resulting in three sets of events under the DC framework per window. This process was repeated for the set of windows for each pair using an array of eight DC thresholds (θ), ranging from 0.015% to 0.029% with gaps of 0.002% between each threshold totalling 5488 (14 pairs $\times$ 8 thresholds $\times$ 49 windows) datasets. Each threshold produced a different summary of the data, therefore requiring the FDRL agent to be tested on multiple thresholds to identify how effective it is at trading under the DC sampling

[2]https://www.truefx.com/truefx-historical-downloads/

[3]Limitations inherent to the data source meant these periods for the two sets of currency pairs were collected separately, hence the difference in date ranges.

framework.

4.3.2 Hyperparameter Optimisation

Training models using Deep Reinforcement Learning algorithms is a computationally expensive process so a grid search was run over a subset of validation sets to identify the most effective array of different model architectures and activation functions. From a subset of the total number of validation sets, it was determined that both the policy and the value networks produced the best results with two hidden layers of 64 units each and the ReLU activation function. This means that the full architecture of the policy network was (150, 64, 64, 2). The number of input neurons was defined by the 30 input values per time step as shown in Table 4.1 (no. indicators $\times$ no. periods + DC start and DC end price = 30 features per time step) multiplied by the lag of 5 time steps. The final two neurons of the policy network were then for buy and sell actions.

The hyperparameters `batch_size` and `n_epochs` were also identified through a grid search on a subset of validation sets. It was observed that a `batch_size` of 65,536 and an `n_epochs` value of 10 produced the most favourable results without introducing impractical computational overhead.

State space lag and training time steps were also optimised using a heuristic approach. By performing a grid search over the same subset of validation sets on lag values of 5, 10 and 20, it was clear that no lag value offered any significant benefit over another but the computational expense increase associated with larger lag values made 5 trends the most appropriate value. The same optimisation process was used for identifying the number of training time steps. Training the subset of models for 1 million time steps showed that after 200,000 time steps the models experienced no significant performance increases in the validation sets and in some cases began to over fit the training data.

From preliminary testing it was discovered that the agents cannot trade profitably outside of consolidation periods, so in order to avoid making trades outside

of the consolidation periods, the filtering approach was developed to identify when the market was entering these periods. This approach involved identifying, under the DC framework, when the market had a certain number of successive trends with no overshoot event. The number of preceding successive trends that would trigger trading was a hyperparameter that was determined by testing the trading algorithm with values of 2, 5, 10 and 20 successive trends. Setting this hyperparameter at 5 produced the highest total return on the subset of validation sets and was determined the optimal value, so was used to produce the final results.

4.3.3 Benchmarks

The FDRL trading benchmarks consist of Buy and Hold (B&H), BLV (Blind Low Volatility), RSI (Relative Strength Index) and MAC (Moving Average Crossover) with each benchmark strategy defined below.

Buy and Hold (B&H) The B&H benchmark strategy enters a long position on the first DC event and then exits that position on the final DC event, making a single trade over the duration of the data. The B&H strategy is a common financial benchmark, as it is a passive strategy (not active trading) so is a useful comparison to strategies that perform active trading.

Algorithm 6 Buy and Hold Trading Strategy

Require: Initial balance

1: Initialise agent with initial trading balance
2: Buy at the opening price of the first trading period
3: Hold for duration of data
4: Sell at the closing price of the last trading period

Ensure: Output final return

Moving Average Crossover (MAC) The Moving Average Crossover (MAC) strategy is technical analysis strategy that uses data sampled at a fixed interval and two moving averages to identify trading opportunities. This strategy calculates

two moving averages (MAs), namely a fast and a slow MA, derived from the fixed interval sampling of the tick mid-prices for two separate periods. The period for the fast MA is based on a smaller number of periods than the slow MA and as a result is more sensitive to price changes. When the fast MA crosses above the slow MA, a buy signal is given and conversely, when it crosses below, a sell signal is given.

In order to identify a best case scenario metric for this strategy, a grid search was run over four different fixed sample interval (1 second, 15 minute, 1 hour and 4 hour intervals), three different short moving average periods (5, 7 and 10) and three different long moving average periods (10, 14 and 20) and the best result from these is reported in the results in Section 4.4. This strategy represents the more classic form of technical analysis without the use of the DC framework and is used to compare how the performance of an ML and DC based strategy differs from that of a traditional technical analysis strategy.

Relative Strength Index (RSI) The Relative Strength Index (RSI) strategy is based on the RSI technical analysis indicator and uses data sampled at a fixed interval to identify trading opportunities. The RSI strategy calculates the RSI indicator over a given period and defines an oversold and overbought level at 30 and 70 respectively. Whenever the RSI value crosses below the oversold level, a buy trade is executed and the opposite when the RSI value crosses above the overbought level. The intention being that the price will reverse as the market moves back to an equilibrium. When this corrective move is made and reaches the overbought or oversold level, the trade is exited and any profits or losses are taken.

As with MAC, to identify a best case scenario metric for this strategy, the RSI strategy was run over four different fixed sample interval (1 second, 15 minute, 1 hour and 4 hour intervals) and three different RSI periods (10, 14 and 20), the best result from these is reported in the results in Section 4.4. This strategy represents an alternative technical analysis strategy to the MAC strategy described

Algorithm 7 Three Moving Averages Trading Strategy

Input: Define *short_period*, *medium_period*, *long_period*, and *data*
Calculate:
    $short\_MA \leftarrow$ Moving average over *short_period*
    $medium\_MA \leftarrow$ Moving average over *medium_period*
    $long\_MA \leftarrow$ Moving average over *long_period* (filter)
Initialise:
    $position \leftarrow None$
    $entry\_price \leftarrow None$
for each time step t in *data* **do**
  if $short\_MA[t]$ crosses above $medium\_MA[t]$ and both are $> long\_MA[t]$ **then**
    if $position = None$ **then**
      Enter Buy Trade:
      $position \leftarrow buy$
      $entry\_price \leftarrow price[t]$
    end if
  end if
  if $medium\_MA[t]$ crosses below $short\_MA[t]$ and both are $< long\_MA[t]$ **then**
    if $position = buy$ **then**
      Exit Trade:
      $position \leftarrow None$
      Record profit or loss ($exit\_price - entry\_price$)
    end if
  end if
end for

previously and again is used to compare how the performance of an ML and DC based strategy differs from that of a traditional technical analysis strategy.

Algorithm 8 RSI Trading Strategy

Input: Define $RSI\_period = 140$, $overbought = 75$, $oversold = 25$, and $data$
Calculate:
    $RSI \leftarrow$ Relative Strength Index over RSI_period
Initialise:
    $position \leftarrow None$
    $entry\_price \leftarrow None$
for each time step t in $data$ **do**
  if $RSI[t]$ crosses below $oversold$ level **then**
    if $position = None$ **then**
      Enter Buy Trade:
      $position \leftarrow buy$
      $entry\_price \leftarrow price[t]$
    end if
  end if
  if $RSI[t]$ crosses above $overbought$ level **then**
    if $position = buy$ **then**
      Exit Trade:
      $position \leftarrow None$
      Record profit or loss $(exit\_price - entry\_price)$
    end if
  end if
end for
Optimisation (Grid Search):
Define range of possible values for $overbought$ and $oversold$ levels
for each pair of $overbought$ and $oversold$ levels **do**
  Apply the strategy
  Evaluate performance (e.g., profit, drawdown)
end for
Output: Best pair of $overbought$ and $oversold$ levels and trade performance

Blind Low Volatility (BLV) The filter used in the FDRL trading strategy ensures that the agent can only trade once there has been a consolidation period, a set of events characterised by a lack of overshoot move as described in section 4.2.3. In order to compare the marginal benefit of employing DRL, a simple DC-based strategy that tests the trading results of just the filter can be used. This strategy is referred to as the Blind Low Volatility strategy and tells the agent to buy at the DCC of a downtrend and sell at the DCC of an uptrend when

the filter allows trading. When the filter no longer allows trading, the rule-based system will exit the market and not trade until permission from the filter is granted again. Given the agent's propensity to trade in consolidation periods, by testing a strategy that is not selective of the type of consolidation period it trades, it can be determined if the agent is inferring both the significance of the oscillating trends as well as the behaviour of the future trends.

Algorithm 9 Blind Low Volatility Trading Strategy

Require: Historical price data, Threshold for trend detection, Number of no-overshoot trends (N)

Initialise agent state and position status (flat, long, short) as flat
Set trade permission to false
for each trading period t **do**
    Calculate DC move for period t
    Update no-overshoot trend counter based on DC move and overshoot
    Set trade permission to true if counter $\geq N$, otherwise false
    if trade permission is false and holding a position **then**
        Exit market
    end if
    if trade permission is true **then**
        if position status is flat **then**
            Buy at DCC of downtrend or sell at DCC of uptrend, update position status
        else if long position and uptrend or short position and downtrend **then**
            Exit market, set position status to flat
        end if
    end if
end for

Ensure: Output trading strategy based on trades made during low volatility periods

4.4 Results

This section presents the results of total return, maximum drawdown and Calmar ratio between the FDRL trading strategy and the benchmarks outlined in Section 4.3.3. Total return, maximum drawdown, and Calmar ratio metrics are evaluated across all strategies under a fixed transaction cost of 0.025%. The FDRL and Blind Low Volatility (BLV) strategies were tested across DC thresholds (0.015%–0.029%), all other benchmarks used fixed-interval sampling and therefore produce a single result per currency pair.

The FDRL agents demonstrate exceptional returns that exceed conventional expectations for FX strategies operating under comparable cost constraints. These returns exhibit patterns suggesting learned strategic exploitation by the agent of the fixed transaction cost structure and other common assumptions made when backtesting such as a lack of slippage constraints, resulting in the agent making frequent micro-transactions that capitalise on the fixed transaction cost and lack of slippage (both of which are addressed in Chapter 6). Section 4.5 investigates the learned behaviour of the agents to understand how they managed to produce such extreme returns and suggests ways in which this information could be used to make future efforts more realistic while still maintaining profitability.

Total Return The total returns for each strategy, as presented in Table 4.2a, reveal differences in performance across the 14 currency pairs. The extreme returns achieved by the EUR/CHF, EUR/GBP, and EUR/JPY currency pairs stand out, with peaks reaching 27971.37%, 11881.4%, and 154275.3% respectively. These three pairs significantly outperformed the other 11 currency pairs, demonstrating that they learned a unique trading strategy under the FDRL framework (see Section 4.5 for further investigation into trading behaviour). The results show substantial returns across most currency pairs during the testing period. Exceptions to this trend include 7 pair-threshold combinations that resulted in negative returns. These exceptions include AUD/JPY at DC thresholds of 0.015%, 0.017%,

Table 4.2: FDRL Total Return results and Friedman significance test results.

(a) Total Return (%) by DC threshold for DRL and BLV. MAC, RSI and B&H strategies are fixed time interval based strategies, so only a single value is presented per currency pair. Best value per currency pair is denoted in boldface and best value per DC threshold is underlined.

Pair/θ	0.015%		0.017%		0.019%		0.021%		0.023%		0.025%		0.027%		0.029%		B&H	MAC	RSI
	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV			
AUD/JPY	-6.34	-39.34	-3.96	-33.29	-2.77	-28.75	6.24	-24.90	7.84	-22.39	16.86	-20.00	9.55	-17.73	**22.06**	-15.43	0.26	-1.90	1.38
AUD/USD	43.08	-27.48	13.06	-25.09	72.95	-22.53	40.56	-20.69	102.87	-19.14	44.76	-17.94	**153.5**	-16.61	56.88	-15.48	-5.44	-4.66	0.84
CAD/JPY	-4.25	-24.90	-1.29	-20.53	27.15	-17.12	4.64	-14.61	6.40	-12.89	18.93	-11.64	**76.36**	-9.85	53.97	-8.89	1.02	-2.33	3.40
CHF/JPY	-5.25	-15.34	1.89	-12.72	-1.75	-10.53	8.33	-8.65	9.02	-7.08	8.79	-6.03	15.07	-5.11	**28.99**	-4.67	13.93	7.52	-0.31
EUR/CHF	159.45	-91.98	2268.72	-86.14	11899.52	-80.25	7124.94	-75.04	20897.29	-70.69	21192.54	-67.14	**27971.37**	-63.97	8909.44	-61.15	-4.79	-4.82	0.30
EUR/GBP	209.49	-85.54	2199.69	-78.89	1668.07	-73.15	1966.08	-67.58	7624.45	-63.25	5255.38	-58.61	7347.38	-55.00	**11881.4**	-51.80	3.51	1.83	-3.65
EUR/JPY	3181.26	-97.31	2631.49	-95.51	45464.81	-93.42	16089.53	-91.20	13070.63	-88.86	116953.36	-86.47	52592.02	-84.21	**154275.3**	-82.17	10.04	2.47	1.87
EUR/USD	38.33	-21.95	1.34	-19.04	65.27	-16.79	29.08	-15.03	64.28	-13.32	**85.03**	-11.70	67.4	-10.57	79.14	-9.63	-5.98	5.58	0.23
GBP/JPY	4.99	-55.01	1.02	-48.58	21.21	-43.42	25.17	-38.82	28.97	-35.05	20.63	-31.47	**72.1**	-28.47	69.99	-25.33	6.69	7.67	1.08
GBP/USD	85.86	-52.18	156.70	-46.61	128.83	-41.70	79.28	-37.05	125.09	-33.33	126.41	-30.66	**266.03**	-28.19	54.95	-26.28	-12.01	4.36	-0.42
NZD/USD	16.46	-41.49	26.71	-38.10	57.88	-33.64	105.28	-30.73	73.91	-27.80	186.41	-25.56	**303.08**	-23.80	150.11	-22.96	-5.86	-7.62	4.09
USD/CAD	45.51	-17.95	20.63	-15.48	34.05	-13.82	20.49	-12.54	31.10	-11.57	30.15	-10.76	37.2	-10.09	**56.39**	-9.35	7.14	-3.01	-1.04
USD/CHF	72.33	-51.92	109.00	-46.89	317.46	-42.83	105.33	-38.95	409.96	-35.21	226.04	-32.22	308.17	-29.81	**420.71**	-27.86	1.54	7.08	-3.82
USD/JPY	20.13	-27.76	14.26	-22.74	58.03	-19.57	20.54	-17.01	43.21	-14.69	**68.46**	-13.33	55.13	-12.11	59.65	-11.12	13.31	14.06	-5.80

(b) Significance test results using Friedman and Conover post-hoc test.

Friedman test p-value		**9.527e-67**
	Ave. Rank	p_{Con}
FDRL (c)	1.28	-
B&H	2.81	**1.407e-44**
MAC	2.91	**6.546e-40**
RSI	3.00	**6.181e-54**
BLV	5.00	**3.955e-140**

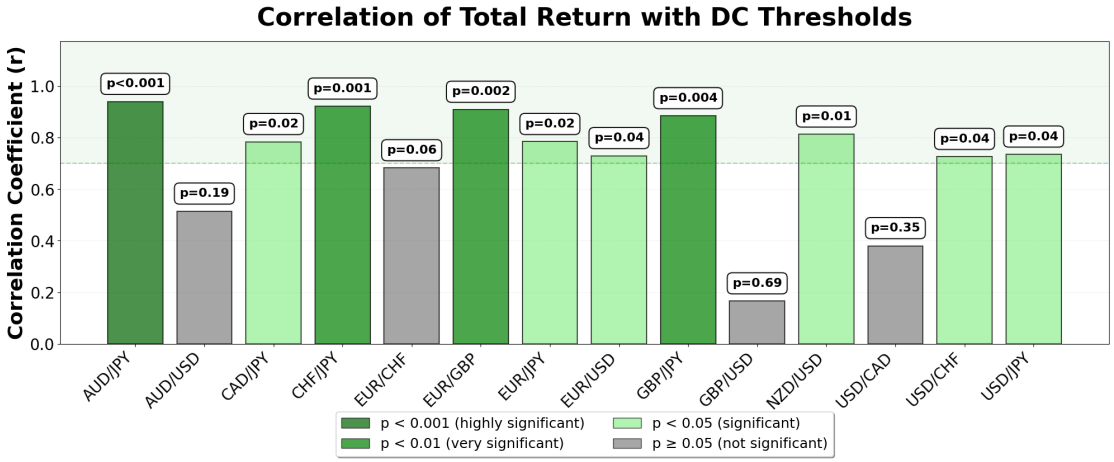

Figure 4.3: FDRL Total Return Correlation Coefficients

and 0.019%, CAD/JPY at thresholds of 0.015% and 0.017% and CHF/JPY at 0.015% and 0.019%. This outcome suggests that while the strategy was effective overall, its performance varied depending on the currency pair and DC threshold under which the data was sampled.

When the variation in FDRL performance across different DC thresholds is analysed, it can be observed that higher DC thresholds are generally associated with increased total returns. Figure 4.3 presents the correlation coefficients between the total returns of each currency pair and the 8 DC thresholds ranging from 0.015% to 0.29%. Most currency pairs show a correlation coefficient exceeding 0.6, with pairs such as AUD/JPY, CHF/JPY, EUR/GBP, and GBP/JPY showing correlations above 0.8. Notably, the majority of these correlations are statistically significant ($p < 0.05$), with several pairs including AUD/JPY ($p < 0.001$), CHF/JPY ($p = 0.001$), EUR/GBP ($p = 0.002$), and GBP/JPY ($p = 0.004$) demonstrating highly significant relationships. Exceptions to this trend include AUD/USD ($p = 0.19$), GBP/USD ($p = 0.69$), and USD/CAD ($p = 0.35$), which display correlation coefficients below 0.6 and non-significant p-values greater than 0.05. These findings align with the hypothesis derived from Table 4.2a, supporting the conclusion that performance improves with higher DC sampling thresholds.

When comparing FDRL to the BLV trading strategy that blindly implements the no OS filter, it can be seen that FDRL is by far the superior trading strategy.

The BLV strategy consistently produces negative returns across all currency pairs and DC thresholds demonstrating how ineffective it is relative to FDRL. This clear difference in performance is down the FDRL's ability to adapt to the magnitude of the range of the consolidation period which can exceed the DC threshold provided when successive ticks change by a percentage greater than the provided threshold. This behaviour emphasises the efficacy of the reinforcement learning component in the FDRL framework, which adapts and learns an optimal policy on top of the filtering mechanism that is applied. Without this dynamic learning provided by the DRL component of the trading framework, the strategy's performance would be similar to that of the BLV approach. These findings demonstrate the critical role of reinforcement learning in enhancing trading strategies and achieving consistently high returns.

The buy and hold strategy generated positive returns for 9 out of the 14 currency pairs, demonstrating that over half of the pairs generated positive returns without requiring active trading. The scale of the passive buy and hold returns however, is significantly overshadowed by the results of FDRL. This difference in these two results shows clearly that FDRL is the more effective trading strategy and active trading is required in order to generate more returns than the buy and hold strategy over this period in time. These results support the argument that, while the buy and hold approach may generate some positive results, it does not capitalise on the dynamic market opportunities that active strategies like FDRL can exploit.

The fixed interval technical analysis benchmarks, represented by the Moving Average Crossover (MAC) and Relative Strength Index (RSI) strategies, outperform the buy and hold strategy for the majority of currency pairs (10 of 14). Even among the pairs where active technical analysis trading does not exceed buy and hold returns, positive returns can still be achieved in all cases except USD/CAD. FDRL consistently outperforms both the passive buy and hold approach and the active technical analysis benchmarks, demonstrating that active DC and DRL-

based trading can generate considerably more returns than both the passive buy and hold strategy and the fixed interval, technical analysis trading strategies.

The null hypothesis, stating that there is no statistically significant difference in performance across the trading strategies, was tested using the non-parametric Friedman test and Conover's post hoc test. Table 4.2b reports the results of these tests, including the average rankings for each strategy.

The results demonstrate that the FDRL strategy, which uses both deep reinforcement learning (DRL) with directional change (DC) sampling, significantly outperforms all benchmark strategies. It achieves the highest average ranking (1.28), outperforming the passive buy-and-hold strategy (1.53), fixed-interval technical analysis methods (e.g. MAC and RSI, which perform similarly to buy-and-hold), and the Blind Low Volatility (BLV) strategy. While BLV employs DC sampling, its reliance on rigid rule-based trading as opposed to more dynamic DRL decision making, leads to consistently poor performance across all datasets. These results highlight the advantage of combining reinforcement learning with DC sampling to optimise trading outcomes, emphasising the limitations of traditional time-based sampling and rule-based trading approaches.

Maximum Drawdown The potential of the FDRL strategy is undoubtedly promising, yet assessing its associated risks is essential to gauge the true value of the strategy. One critical measure for evaluating risk is the maximum drawdown metric, which indicates the largest peak-to-trough decline in the strategy's value. Table 4.3a provides the maximum drawdowns for each strategy, revealing that while FDRL demonstrates strong performance for some currency pairs, it also encounters significant drawdowns for others. For instance, in the cases of AUD/USD and NZD/USD, the strategy experiences substantial drawdowns for most directional change (DC) thresholds, reaching peaks of 77.33% and 69.98% respectively at a DC threshold of 0.015%. Such severe drawdowns, particularly at the start of trading, can adversely impact the total return results by reducing the available capital for subsequent trades, potentially diminishing profitability.

Table 4.3: FDRL Maximum Drawdown results and Friedman significance test results.

(a) Maximum Drawdown (%) by DC threshold for DRL and BLV. MAC and RSI strategies are fixed time interval based strategies, so only a single value is presented per currency pair. Best value per currency pair is denoted in boldface and best value per DC threshold is underlined.

Pair/θ	0.015%		0.017%		0.019%		0.021%		0.023%		0.025%		0.027%		0.029%		MAC	RSI
	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV		
AUD/JPY	6.34	39.34	3.96	33.29	2.77	28.75	**0.18**	24.90	0.26	22.39	0.34	20.00	0.47	17.73	0.27	15.43	10.65	8.39
AUD/USD	77.33	27.48	60.47	25.09	72.98	22.53	66.12	20.70	60.62	19.14	30.4	17.94	38.96	16.61	51.03	15.48	14.14	**2.35**
CAD/JPY	4.25	24.90	1.29	20.53	0.24	17.12	0.4	14.61	0.49	12.89	0.21	11.64	0.36	9.85	**0.15**	8.89	13.96	5.04
CHF/JPY	5.25	15.34	0.16	12.72	1.75	10.53	0.14	8.66	0.10	7.09	**0.09**	6.04	0.13	5.12	**0.09**	4.68	7.85	3.51
EUR/CHF	38.59	91.98	4.63	86.14	8.3	80.25	5.94	75.04	10.31	70.69	8.34	67.14	11.47	63.97	6.35	61.15	6.83	**1.38**
EUR/GBP	22.73	85.54	35.84	78.89	6.76	73.15	20.1	67.58	10.05	63.25	8.75	58.61	6.24	55.00	6.52	51.80	**4.65**	5.77
EUR/JPY	0.29	97.31	0.73	95.51	0.16	93.42	0.18	91.20	0.15	88.86	0.2	86.47	**0.07**	84.21	0.15	82.17	12.17	3.12
EUR/USD	28.14	21.95	27.67	19.04	22.01	16.79	24.74	15.03	40.34	13.32	16.14	11.70	28.69	10.57	24.65	9.63	5.54	**3.7**
GBP/JPY	0.25	55.01	0.13	48.58	**0.03**	43.42	0.16	38.82	0.17	35.05	0.13	31.48	0.1	28.47	0.18	25.33	6.65	4.58
GBP/USD	40.13	52.18	59.57	46.61	32.88	41.70	27.19	37.05	16.97	33.33	11.02	30.67	22.17	28.19	36.75	26.28	9.77	**6.66**
NZD/USD	69.98	41.49	28.02	38.10	41.09	33.64	52.27	30.73	34.72	27.80	24.78	25.56	22.19	23.80	20.09	22.96	12.57	**1.59**
USD/CAD	16.76	17.96	13.13	15.49	17.36	13.83	16.25	12.55	8.02	11.58	14.23	10.76	11.47	10.09	11.33	9.35	5.23	**3.2**
USD/CHF	23.33	51.92	18.67	46.89	17.16	42.83	9.87	38.95	8.20	35.21	12.52	32.22	8.7	29.81	10.43	27.86	6.16	**5.14**
USD/JPY	0.39	27.76	0.27	22.75	0.32	19.57	0.49	17.02	0.19	14.69	0.26	13.33	0.23	12.11	**0.16**	11.12	8.04	9.7

(b) Significance test results using Friedman and Conover post-hoc test.

Friedman test p-value		**1.381e-33**
	Ave. Rank	p_{Con}
RSI	1.56	3.552e-09
FDRL (c)	2.37	-
MAC	2.38	1.221e-01
BLV	3.69	**1.758e-28**

Despite the poor performance of FDRL on AUD/USD and NZD/USD, certain currency pairs exhibit minimal drawdowns under the FDRL framework. Pairs such as AUD/JPY, CAD/JPY, CHF/JPY, EUR/JPY, GBP/JPY, and USD/JPY demonstrate very low maximum drawdowns, with values as low as 0.18%, 0.15%, 0.09%, 0.07%, 0.03%, and 0.16% respectively. This pattern suggests that the FDRL strategy is particularly effective at mitigating drawdown risks when applied to JPY currency pairs. The consistent performance across these pairs demonstrates the strategy's robustness in managing risk within this specific subset of currency pairs, suggesting potential opportunities for strategic application in JPY-focused trading scenarios.

When comparing FDRL to the technical analysis benchmarks, it becomes clear that while technical analysis strategies may show relatively weak performance in terms of total return, they provide stronger competition when considering maximum drawdown and therefore trading risk. FDRL achieved the most favourable maximum drawdown in six of the fourteen currency pairs, all of which involve JPY, while the RSI benchmark strategy outperforms FDRL in seven pairs, and MAC outperforms FDRL in one pair (EUR/GBP). For the six currency pairs where FDRL achieves the lowest drawdown, it tends to provide the best maximum drawdown by a substantial amount. FDRL's best maximum drawdown values for these pairs range between 0.03% and 0.18%, significantly outperforming the benchmarks, whose best maximum drawdown values on the remaining eight pairs range from 1.38% to 6.66%. The BLV strategy demonstrates particularly poor performance regarding maximum drawdown. In many cases, its maximum drawdown equals its total return, indicating a steady decline in returns from the start of trading.

An analysis of the correlation between DC threshold and maximum drawdown reveals a pattern similar to that observed with total return where maximum drawdown performance generally improves as DC thresholds increase. Since lower maximum drawdowns are preferable, supporting evidence of this pattern would be shown by a negative correlation coefficient between maximum drawdown and

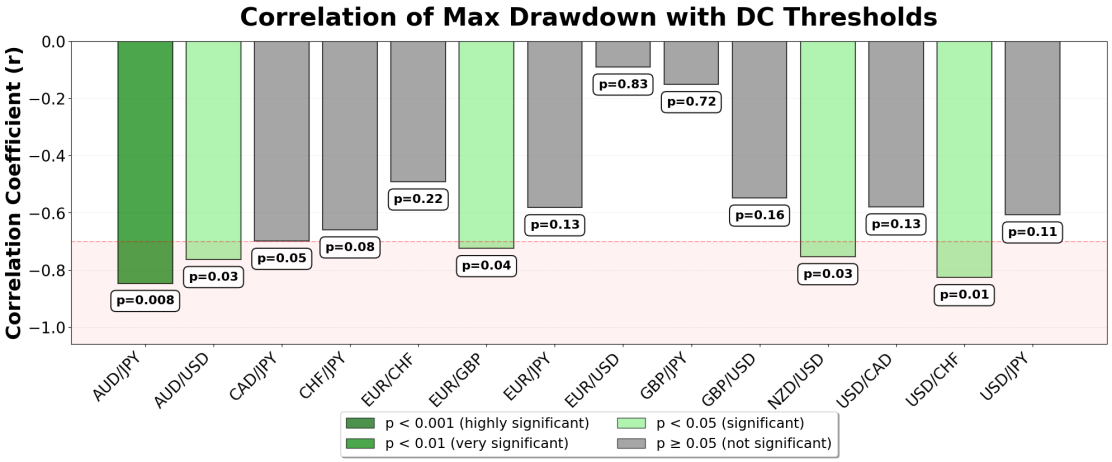

Figure 4.4: FDRL Maximum Drawdown Correlation Coefficients

the 8 DC thresholds. Figure 4.4 illustrates this relationship, with correlation coefficients below -0.7 observed for currency pairs AUD/JPY, AUD/USD, EUR/GBP, NZD/USD, and USD/CHF. Among these, AUD/JPY ($p = 0.008$), AUD/USD ($p = 0.03$), EUR/GBP ($p = 0.04$), NZD/USD ($p = 0.03$), and USD/CHF ($p = 0.01$) demonstrate statistically significant negative correlations, suggesting that higher DC thresholds significantly reduce maximum drawdowns for these pairs, enhancing the risk profile of the strategy. For the remaining currency pairs, the relationship is slightly weaker but still notable. Most show correlation coefficients of -0.5 or less, indicating moderate improvements in drawdown performance with increasing DC thresholds, though these relationships are generally not statistically significant ($p \geq 0.05$). Exceptions such as EUR/USD ($p = 0.83$) and GBP/JPY ($p = 0.72$) display only very weak negative correlations with non-significant p-values, suggesting that DC threshold adjustments have minimal impact on drawdown performance for these pairs.

The results of the non-parametric Friedman test and the post hoc Conover test for maximum drawdown, along with the rankings of each trading strategy, are presented in Table 4.3b. The Friedman test indicates statistically significant differences among the four trading strategies, as shown by a p-value of 1.381e-33 that does not exceed the significance threshold of 0.05 set prior to testing. From the rankings, the RSI strategy is the best performing algorithm for maximum

drawdown, achieving an average rank of 1.56. RSI outperforms the FDRL strategy which has a similar average ranking to MAC, with values of 2.37 and 2.38 respectively. The statistical significance of RSI's performance is further supported by the Conover post hoc test, which reveals a p-value of 3.552e-09 when comparing RSI with FDRL. The BLV strategy ranks as the worst performing strategy, with an average ranking of 3.69. FDRL significantly outperforms BLV, demonstrating BLV's limitations in managing maximum drawdown effectively.

Calmar Ratio To assess the risk-return trade-off of a trading strategy, the Calmar ratio is used as a risk-adjusted return metric. This ratio measures trading performance by comparing the total return to the maximum drawdown of the strategy. Table 4.4a reports the Calmar ratios for the FDRL, BLV, RSI and MAC strategies. As with the total return results presented in Table 4.2a, the FDRL strategy shows significantly higher Calmar ratios compared to the other three strategies. For the EUR/JPY currency pair, the combination of extreme total returns and very low maximum drawdown levels results in exceptionally high Calmar ratios, with the largest value reaching 999,582.79 at a DC threshold of 0.029%. The underlying reasons for these extreme returns are explored in Section 4.5.

While all EUR/JPY values generate substantial Calmar ratios, it is clear that the nature of the Calmar ratio calculation will result in these values when such large total returns are observed. These findings suggest that, when simulating trading on the observed data for EUR/JPY across all thresholds under the defined constraints, FDRL is a particularly effective approach to trading due to outperforming other currency pairs by a considerable margin. The specific characteristics of this currency pair and their implications for the FDRL strategy are further analysed in Section 4.5.

The negative Calmar ratios observed in the FDRL strategy align with combinations of currency pairs and DC thresholds where negative total returns were reported, as shown in Table 4.2a. All negative Calmar ratios are observed for currency pairs AUD/JPY, CAD/JPY, and CHF/JPY, each producing a Calmar

Table 4.4: FDRL Calmar Ratio results and Friedman significance test results.

(a) Calmar Ratio by DC threshold for DRL and BLV. MAC and RSI strategies are fixed time interval based strategies, so only a single value is presented per currency pair. Best value per currency pair is denoted in boldface and best value per DC threshold is underlined.

Pair/θ	0.015%		0.017%		0.019%		0.021%		0.023%		0.025%		0.027%		0.029%		MAC	RSI
	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV	FDRL	BLV		
AUD/JPY	-1.00	-1.0	-1.00	-1.0	-1.00	-1.0	34.36	-1.0	30.54	-1.0	49.73	-1.0	20.31	-1.0	**81.8**	-1.0	-0.18	0.16
AUD/USD	0.56	-1.0	0.22	-1.0	1.00	-1.0	0.61	-1.0	1.7	-1.0	1.47	-1.0	**3.94**	-1.0	1.11	-1.0	-0.33	0.36
CAD/JPY	-1.00	-1.0	-1.00	-1.0	110.86	-1.0	11.61	-1.0	13.16	-1.0	89.78	-1.0	209.9	-1.0	**365.51**	-1.0	-0.17	0.67
CHF/JPY	-1.00	-1.0	12.07	-1.0	-1.00	-1.0	58.95	-1.0	87.36	-1.0	92.93	-1.0	115.17	-1.0	**309.44**	-1.0	0.96	-0.09
EUR/CHF	4.13	-1.0	490.18	-1.0	1433.85	-1.0	1200.21	-1.0	2027.37	-1.0	**2539.78**	-1.0	2439.36	-1.0	1403.68	-1.0	-0.71	0.22
EUR/GBP	9.22	-1.0	61.37	-1.0	246.67	-1.0	97.81	-1.0	758.58	-1.0	600.8	-1.0	1177.05	-1.0	**1823.5**	-1.0	0.39	-0.63
EUR/JPY	11111.85	-1.0	3593.02	-1.0	291249.23	-1.0	90520.35	-1.0	87994.02	-1.0	588490.56	-1.0	785908.07	-1.0	**999582.79**	-1.0	0.20	0.60
EUR/USD	1.36	-1.0	0.05	-1.0	2.97	-1.0	1.18	-1.0	1.59	-1.0	**5.27**	-1.0	2.35	-1.0	3.21	-1.0	1.01	0.06
GBP/JPY	20.28	-1.0	7.71	-1.0	645.96	-1.0	160.67	-1.0	172.82	-1.0	158.19	-1.0	**689.78**	-1.0	395.84	-1.0	1.15	0.24
GBP/USD	2.14	-1.0	2.63	-1.0	3.92	-1.0	2.92	-1.0	7.37	-1.0	11.47	-1.0	**12.0**	-1.0	1.5	-1.0	0.45	-0.06
NZD/USD	0.24	-1.0	0.95	-1.0	1.41	-1.0	2.01	-1.0	2.13	-1.0	7.52	-1.0	**13.66**	-1.0	7.47	-1.0	-0.61	2.57
USD/CAD	2.72	-1.0	1.57	-1.0	1.96	-1.0	1.26	-1.0	3.88	-1.0	2.12	-1.0	3.24	-1.0	**4.98**	-1.0	-0.58	-0.32
USD/CHF	3.10	-1.0	5.84	-1.0	18.50	-1.0	10.68	-1.0	**49.98**	-1.0	18.06	-1.0	35.43	-1.0	40.32	-1.0	1.15	-0.74
USD/JPY	51.95	-1.0	53.28	-1.0	182.04	-1.0	41.86	-1.0	224.35	-1.0	264.34	-1.0	240.19	-1.0	**369.16**	-1.0	1.75	-0.60

(b) Significance test results using Friedman and Conover post-hoc test.

Friedman test p-value		**2.273e-55**
	Ave. Rank	p_{Con}
FDRL (c)	1.23	-
RSI	2.38	**1.878e-43**
MAC	2.43	**1.159e-35**
BLV	3.97	**2.255e-124**

ratio of -1.0 at a DC threshold of 0.015%. While JPY pairs are characterised by lower maximum drawdowns, the Calmar ratio results reveal that these reduced drawdowns are associated with negative returns. This indicates that the strategy learned for these pairs results in small losses, which translate into small negative returns and consequently a Calmar ratio of -1.0. AUD/JPY and CAD/JPY also produce a -1.0 Calmar ratio at a DC threshold of 0.017%, while AUD/JPY and CHF/JPY display the same Calmar ratio at 0.019%. These findings suggest that for these currency pairs, the FDRL strategy fails to make use of favourable market conditions identified at large DC thresholds, instead resulting in consistent small losses.

Analysis of how Calmar ratios change with the DC threshold indicates that a DC threshold of 0.029% produces the highest Calmar ratio for seven of the fourteen currency pairs. DC thresholds of 0.027%, 0.025%, and 0.023% produce the best Calmar ratios for four, two, and one currency pair respectively. Figure 4.5 displays the correlation coefficients of the Calmar ratio with the DC threshold. Seven of the fourteen currency pairs demonstrate strong positive correlations with correlation coefficients exceeding 0.8, with CHF/JPY ($p = 0.006$), EUR/GBP ($p = 0.002$), EUR/JPY ($p = 0.003$), and USD/JPY ($p = 0.004$) showing highly significant relationships ($p < 0.01$), while AUD/JPY ($p = 0.01$), CAD/JPY ($p = 0.02$), and EUR/CHF ($p = 0.02$) also demonstrate statistically significant correlations ($p < 0.05$). The remaining seven pairs show moderate positive correlations with coefficients near or above 0.5, though only NZD/USD ($p = 0.01$) and USD/CHF ($p = 0.03$) achieve statistical significance among this group, while AUD/USD ($p = 0.10$), EUR/USD ($p = 0.13$), GBP/JPY ($p = 0.21$), GBP/USD ($p = 0.23$), and USD/CAD ($p = 0.09$) show non-significant relationships. This overall trend indicates that the Calmar ratios for all currency pairs positively correlate with the DC sampling threshold, with ten of fourteen pairs showing statistical significance (p ¡ 0.05), thereby providing robust statistical support for the hypothesis that higher thresholds enhance the effectiveness of the FDRL trading strategy in terms

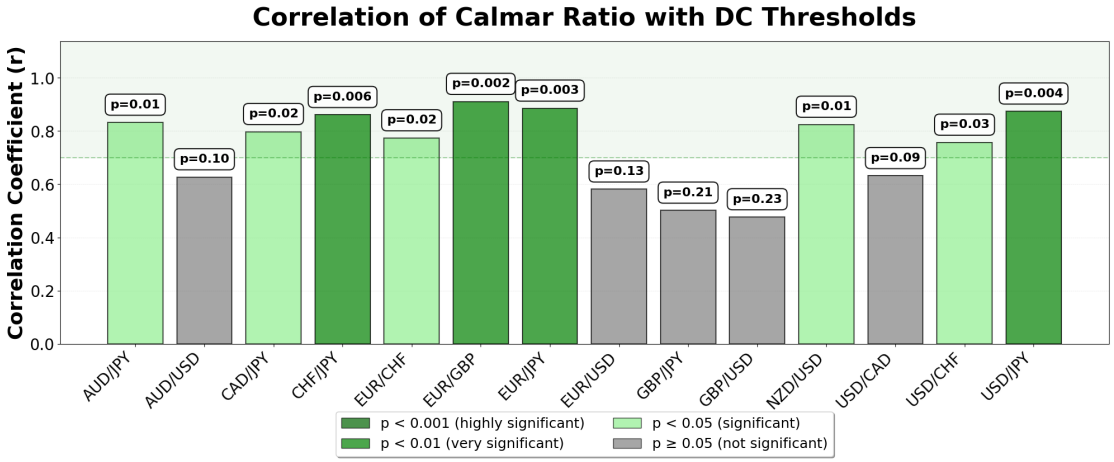

Figure 4.5: FDRL Calmar Ratio Correlation Coefficients

of risk-adjusted performance.

Only four of the fourteen currency pairs achieve Calmar ratios above 1.0[4] under the MAC benchmark, while the RSI benchmark produces a Calmar ratio above 1.0 only for NZD/USD. Overall the benchmarks perform worse than FDRL, this is even more evident when noting that the best performing pairs under the FDRL strategy (EUR/JPY, EUR/GBP, and EUR/CHF) report only mediocre returns when evaluated using the MAC and RSI benchmarks. This contrast shows significant differences in the trading behaviour employed by FDRL compared to traditional technical analysis approaches. These results highlight the efficacy of the FDRL strategy in leveraging the data of specific currency pairs along with the use of the DC sampling algorithm and deep reinforcement learning. These results also show that the market patterns that have been used to generate these extreme Calmar ratios that are not effectively captured by fixed time interval sampled technical analysis strategies such as MAC and RSI.

The null hypothesis states that there is no statistically significant difference in performance across the trading strategies, this is tested using the non-parametric Friedman test and Conover's post hoc test. The results of the non-parametric Friedman test for the Calmar ratios, presented in Table 4.4b, reveal a p-value of 2.273e-55, indicating that the four strategies are significantly different and the

[4]A Calmar Ratio above 1.0 is considered favourable as total return is greater than maximum drawdown.

null hypothesis can be rejected. Further analysis of the average rankings of each strategy shows that FDRL achieves an average Calmar ratio rank of 1.23, outperforming RSI, MAC, and BLV, which have ranks of 2.38, 2.43, and 3.97, respectively. These findings demonstrate the superior performance of FDRL under the trading conditions outlined in this experiment compared to both technical analysis and BLV benchmarks. To further asses the performance of FDRL, the Conover post hoc test was applied to test the significance of each strategy relative to FDRL. The results confirm that FDRL significantly outperforms all benchmark strategies. This outcome suggests that the combination of deep reinforcement learning and DC sampling offers a substantial advantage over fixed interval sampling and traditional technical analysis approaches.

Results Summary The analysis of results presented in Tables 4.2a, 4.3a and 4.4a, along with the corresponding significance Tables 4.2b, 4.3b, and 4.4b, indicates that the proposed FDRL algorithm demonstrates superior performance compared to fixed-interval and technical analysis-based trading strategies. This superiority is evident in both total returns and risk-adjusted returns, as measured by the Calmar ratio. When assessing risk, FDRL shows comparable performance to the MAC strategy but is outperformed by RSI. FDRL however, achieves significantly higher returns for a similar level of risk in currency pairs where RSI also performs well, suggesting that FDRL is the overall best-performing strategy. Table 4.4a reveals that instances where FDRL was outperformed by RSI or MAC are generally marginal, whereas its outperformance over these strategies is more evident. This difference showcases an important aspect not captured by significance testing alone. Given the extreme returns and therefore extreme Calmar ratios observed for FDRL, the next section analyses the trading behaviour of FDRL with the goal of identifying the root cause of these unconventionally extreme results.

4.5 Interpretation

In this section the trading behaviour of the FDRL algorithm is analysed to understand what is causing the large returns that are observed. From the results in Section 4.4 it is clear that the FDRL trading agents were consistently implementing a strategy that can generate considerable returns. To investigate this the profiles of the most extreme currency pairs and the behaviours of the FDRL agents during these highly profitable periods are analysed.

OS Proportion The EUR/CHF, EUR/GBP, EUR/JPY currency pairs tend to produce the most striking results across all eight DC thresholds so the analysis begins with an examination of the profile of these currency pairs. Figure 4.6 shows the mean OS proportion of the total move for each pair at each DC threshold. This is therefore telling us which pairs tend to have total DC sampled moves that are closer to the thresholds used to sample each pair as opposed to moves with much larger OS portions. From this graph it is clear that EUR/CHF, EUR/GBP and EUR/JPY have OS moves that range between 5-10% of total move, however all other pairs tend to have OS moves that range between 20-30% with the exception of USD/CHF which sits between the smaller group of the three and the remaining pairs. The graph also shows that increasing DC thresholds tends to result in a smaller proportion of OS move, an attribute that seems to be a determinant of successful trading using FDRL and is aligned with what would be expected from the improved results for larger thresholds in Tables 4.2a and 4.4a.

To further strengthen the understanding of why certain currency pairs generate substantially higher returns, Figure 4.7 presents the density distribution of OS proportions across all currency pairs. This distribution analysis reveals the fundamental mechanism underlying the superior performance of EUR/CHF, EUR/GBP, and EUR/JPY. The density curves show that these three pairs exhibit different overshoot behaviour compared to other pairs, with their distributions heavily concentrated near zero OS proportion values. This concentration indicates a high

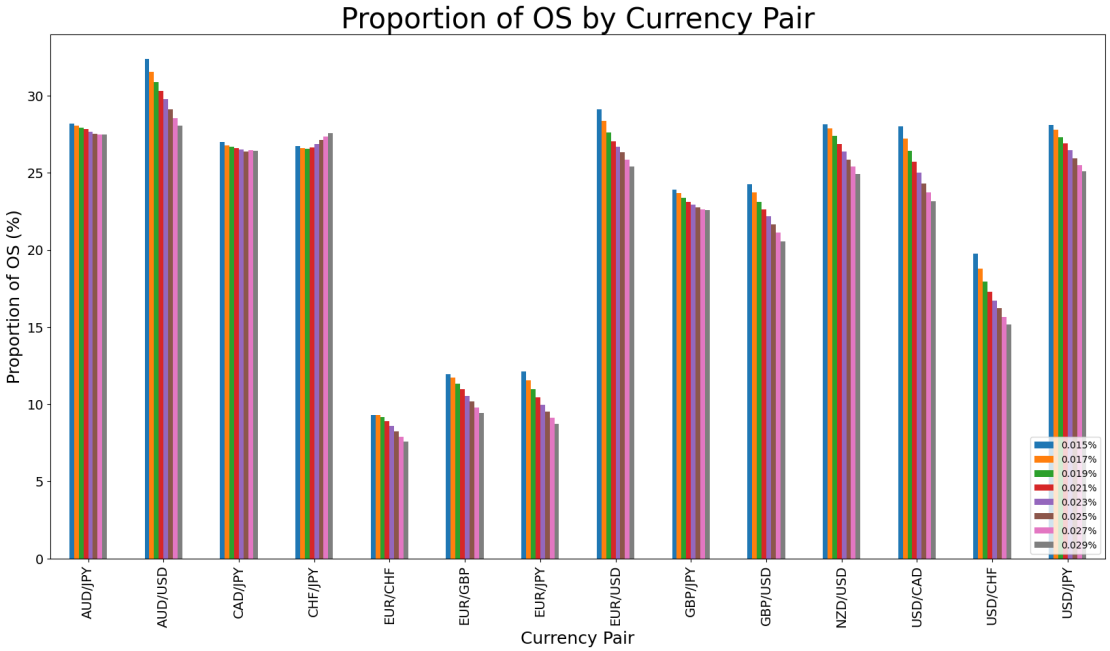

Figure 4.6: Mean OS Proportion of Total Move

frequency of events where price reversals occur immediately or shortly after reaching the DC threshold, creating optimal conditions for the FDRL trading strategy. This is in contrast with pairs such as USD/JPY, AUD/JPY, and CHF/JPY which display much flatter distributions with substantial density extending toward higher OS proportions, indicating that their price movements more frequently overshoot considerably beyond the initial DC threshold before reversing. This overshoot behaviour reduces trading opportunities and profitability as prices move further away from optimal entry points before reversal confirmation.

Figure 4.8 complements Figure 4.6 by presenting the number of events per currency pair with no overshoot. In this figure it is clear that EUR/CHF, EUR/GBP and EUR/JPY all have much larger numbers of events that have no overshoot move in the (i.e. as soon as the DC threshold is achieved by a change in price, the price then reverses). If a currency were to experience prolonged periods of no overshoot then it is clear that with the filter that has been applied to FDRL that the DRL agents are much more likely to trade these periods and in turn generate substantially more profit.

Having identified the key difference in the distribution of overshoot proportions

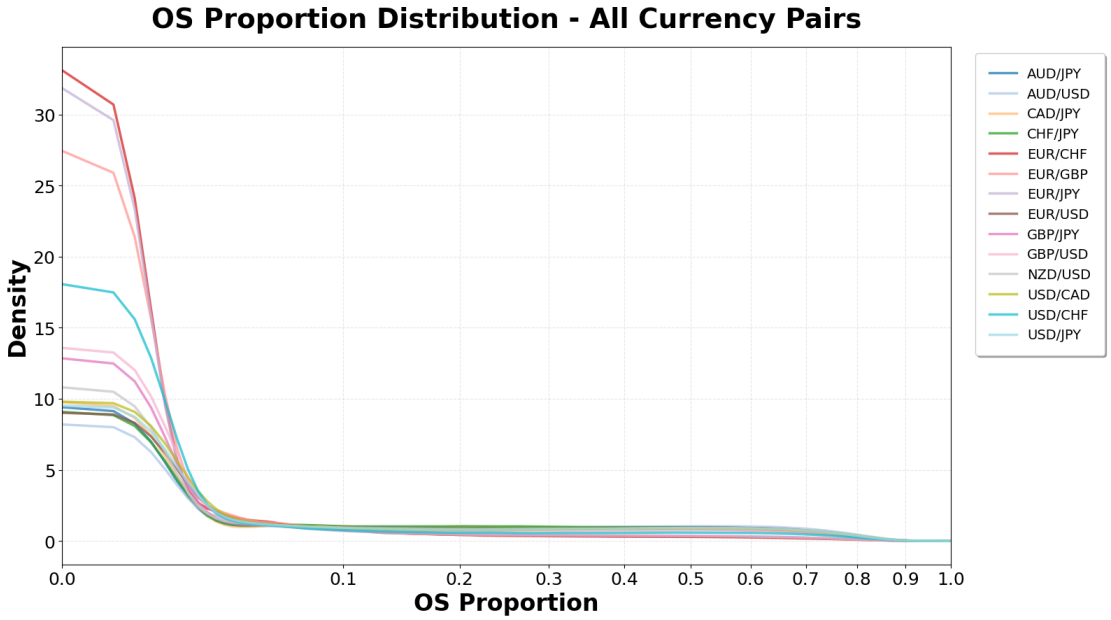

Figure 4.7: OS Proportion Distribution for all Currency Pairs

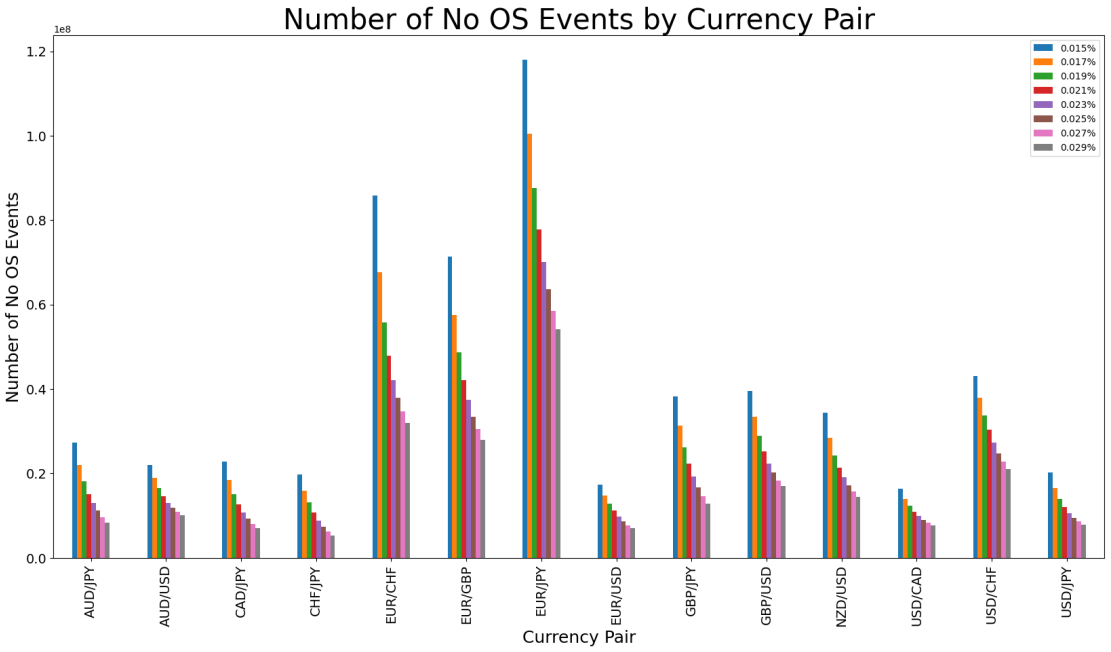

Figure 4.8: Number of No Overshoot Events per Currency-Theta Combination

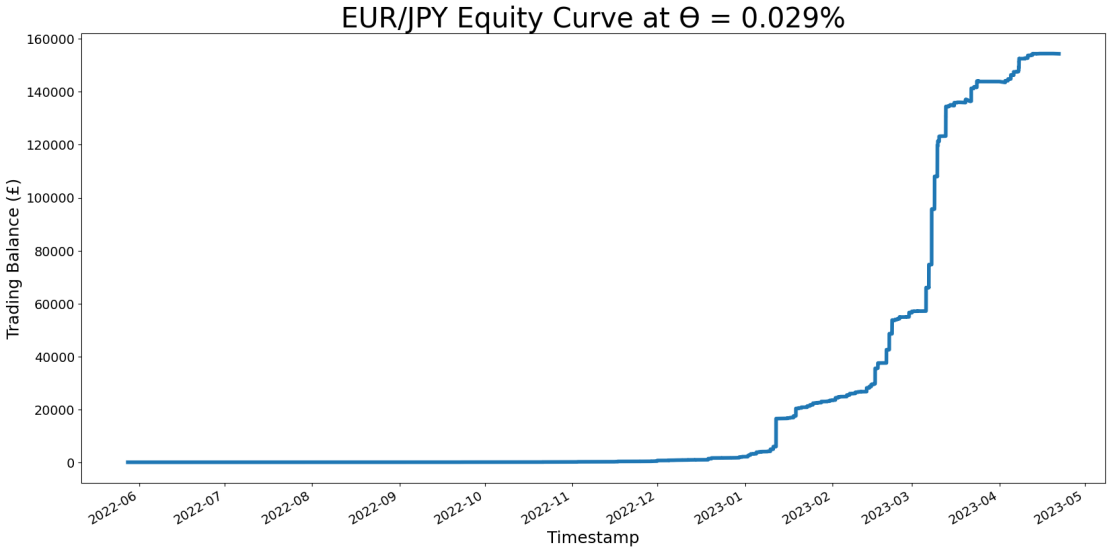

Figure 4.9: FDRL EUR/JPY ($\theta = 0.029\%$) Equity Curve

and the resulting number of no-overshoot events between the three currency pairs that generate the highest total returns and those with lower returns, the next step is to focus on the periods within the dataset that contribute to these larger returns. By isolating these periods, it becomes possible to analyse the specific trading behaviour that occurs.

Equity Curves The equity curves are used to identify the periods where the DRL agent is making the most return. When analysing the spikes in the equity curve in Figure 4.9 it can be seen that there is virtually no deviation from the initial 100 currency unit balance until there are sharp spikes in balance. The fact sharp spikes are observed as opposed to more gradual increases in equity over time suggests that there are periods that the FDRL agent is making significant profits over short periods of time. Spikes in equity at numerous other points in the year were observed, all of which are situated towards the later end of trading. These spikes suggest that this is when the data fits the optimal conditions for the FDRL agents to thrive.

Trading Behaviour From Figure 4.9 the period between 22:30 and 23:00 on 8th January 2023 is identified as the period over which the first initial spike in

the equity curve occurs. Figure 4.10 shows the mid-price over this period. The most notable characteristic of this graph is the densely packed periods of price change around the middle and end sections of the time range. These dense periods represent rapid oscillations in price that tend to be quite uniform. These uniform oscillations are common phenomena observed in high frequency FX data and can occur due to a number of reasons but are often attributed to the small and frequent changes in the limit order book.

Figure 4.11 overlays the DC Sampled data on the raw tick data. From this figure it can be identified where these rapid price oscillations were equal to or greater than the 0.029% threshold at which they are sampled. The dense period in the middle of this time period does not trigger any new DC events as the range over which the price oscillates is not large enough. The dense periods towards the end of this range however triggers the sampling of numerous events, a characteristic that would be missed by fixed interval sampling.

The DC sampled data in Figure 4.11 show us that there are plenty of trading opportunities for the FDRL agent to take action on, these trading actions are shown in Figure 4.12. The dense periods of price changes are identified by FDRL and short positions are executed on alternate DC moves as shown by the red downward arrows. Every trade is exited on the following DC confirmation point and then the next trade entry is shown by the red arrow on the following DC event. The high number of DC events in a short space of time is what leads to these large returns in Table 4.2a. Figure 4.13 shows a close up of the 10 second period between 22:49:00 and 22:49:10 on 2023-01-08 for EUR/JPY, where the alternate trading activity from the agent can be seen in more detail. Not every tick is represented by a DC event over this period and as in some cases there are intermediate ticks between events.

This trading pattern is the main source of profit for all pairs, EUR/JPY and EUR/CHF and EUR/GBP have the highest number of no OS events across all data sets. This therefore means the DRL models have more exposure to this type

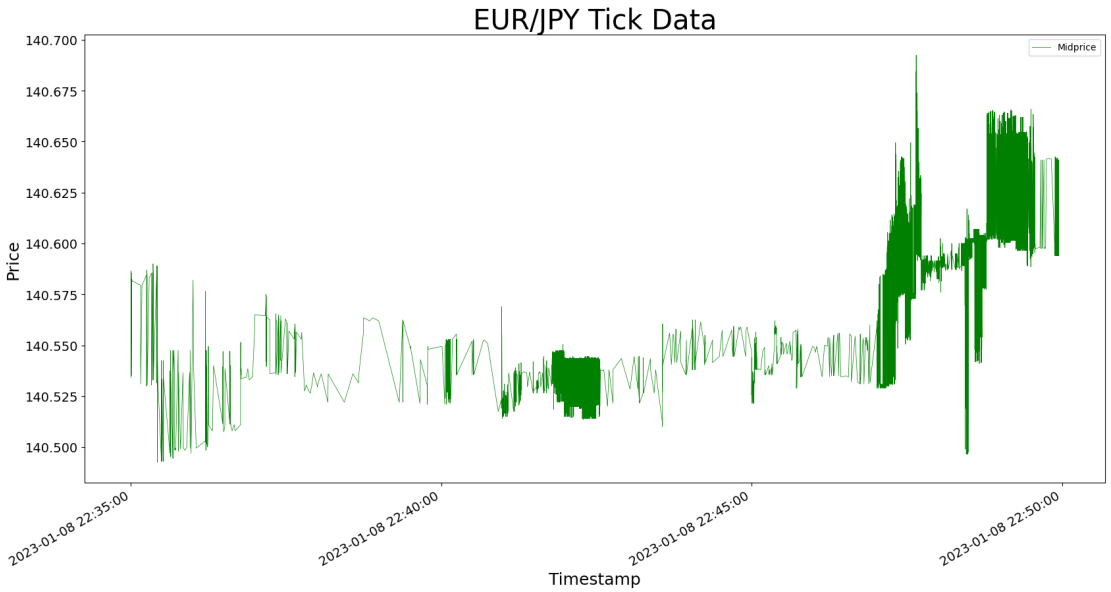

Figure 4.10: EUR/JPY Tick Raw Tick Data between 22:35 and 22:50 on 2023-01-08

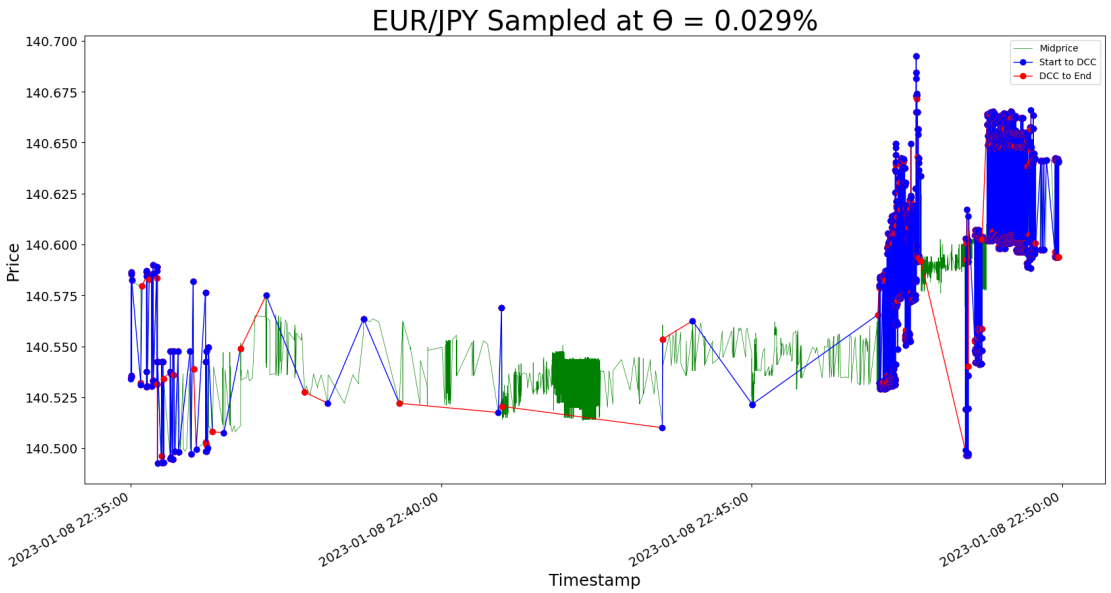

Figure 4.11: EUR/JPY DC Data Sampled at $\theta = 0.029\%$ between 22:35 and 22:50 on 2023-01-08

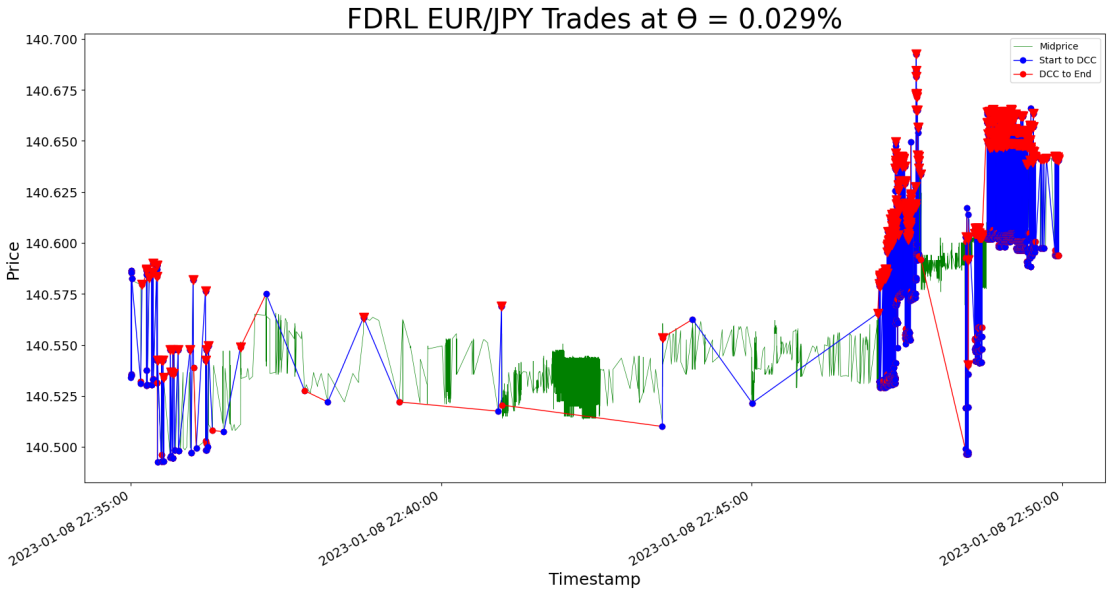

Figure 4.12: FDRL EUR/JPY Trades at $\theta = 0.029\%$ between 22:35 and 22:50 on 2023-01-08

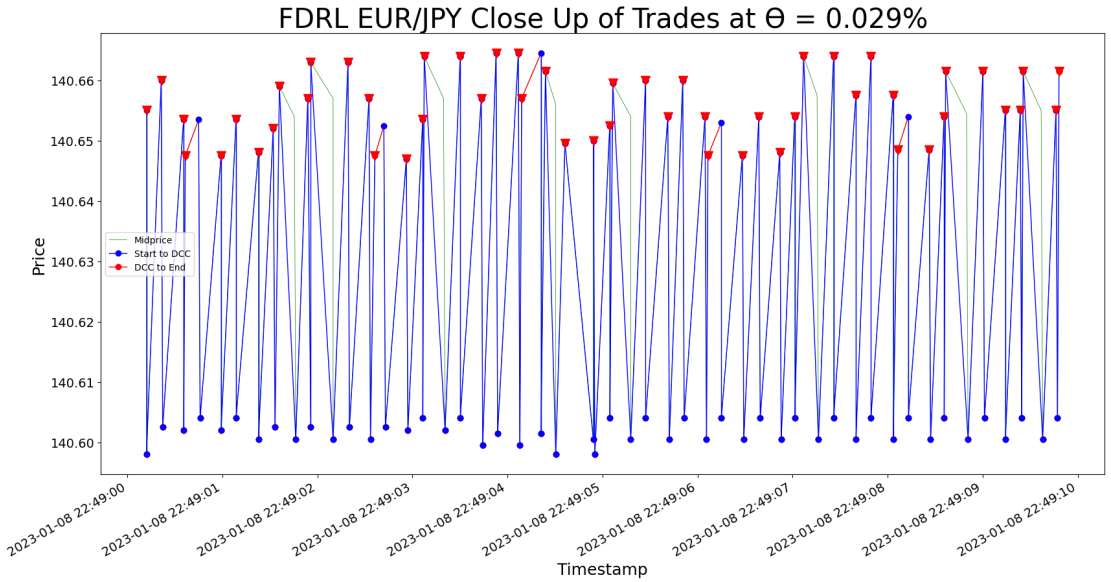

Figure 4.13: FDRL EUR/JPY Trades at $\theta = 0.029\%$ between 22:49:00 and 22:49:10 on 2023-01-08

of data to learn from in the training data, as well as an increased number of these events in the test sets meaning the FDRL agents for these currency pairs are not only better at trading these environments, but also has more opportunity to trade them and therefore more potential profit, as shown by the results.

4.6 Summary

This chapter introduces the Filtered Deep Reinforcement (FDRL) trading algorithm, a deep reinforcement learning based methodology for training agents to trade FX assets using data sampled with the DC sampling algorithm. Each agent is trained per window across 14 different currency pairs and suggests either a buy or sell action at each DC event which is then executed in a simulated market environment. Transaction costs are modelled as a fixed cost of 0.025% of the position size, which encompasses both commission fees and the bid-ask spread typically encountered in foreign exchange markets. Slippage, which refers to the difference between the expected execution price and the actual execution price due to market movements during order processing, is assumed to be negligible in this simulation due to the high liquidity of the major currency pairs traded and is therefore not explicitly modelled.

The FDRL strategy shows extremely positive returns across almost all 14 currency pairs. These high returns easily outperform the passive Buy and Hold and active technical analysis benchmarks that are subject to the same fixed transaction costs as FDRL, suggesting that the FDRL strategy is a much more efficient algorithm for trading while carrying similar levels of risk, therefore producing favourable Calmar ratio thresholds across all DC thresholds and all currency pairs.

Analysis of the trading behaviour of FDRL exposes the strategy that is employed across all pairs and all thresholds, demonstrating that the agents have learnt to identify small oscillations in the price that have been made more apparent by the DC sampling algorithm. The agents can then use these periods of small oscillations to quickly enter and exit trades in one direction, alternating buy and sell

actions to yield profits equal to the DC threshold. The agent then learns to maintain its position when these periods of small oscillations come to an end and only employ the same strategy once the market re-enters this same period. The agent however relies on a filter to execute the appropriate trades as without this filter it applies a similar trading strategy to a market regime within which the price is not oscillating the way it requires for successful trading, resulting in significant losses.

The application of a filter makes FDRL only partially autonomous, meaning that the strategy employed by the model is not fully learnt. The next chapter aims to move towards a fully autonomous agent by the introduction of positional awareness. Positional awareness, along with more advanced training methods, are used to provide the agent with a greater understanding of its position within the market. One limitation of FDRL is its lack of positional awareness, effectively making the strategy stateless as it would not have access to the current position and would only learn to take actions from the current point in the market without consideration for if this is the best decision based on the future profit potential of the current position.

Chapter 5

Positionally-Aware Deep Reinforcement Learning Trading

5.1 Motivation

The previous chapter presented evidence supporting the hypothesis that a trading strategy that has been developed using DC sampling and deep reinforcement learning can outperform passive strategies and strategies developed using fixed interval sampling. The resultant FDRL strategy showed that the trained FDRL agents implement a strategy that exceed conventional expectations of FX strategies. The exceptional returns observed for FDRL can be attributed to the fixed transaction cost level of 0.025% and the rule-based filter, which helped identify periods where these fixed costs worked in the agent's favour.

In the broader pursuit of achieving more realistic trading performance, this chapter first focuses on maximising returns under the existing fixed transaction costs approach, using the Positionally Aware Deep Reinforcement Learning (PADRL) framework. The research undertaken in this chapter aims to develop a system that is as profitable as possible, albeit not entirely realistic. This chapter builds on the findings of the previous chapter by re-evaluating the approach that was taken for FDRL to make the algorithm more autonomous with the introduction of a posi-

tionally aware agent and higher transaction costs of 0.035%. The main limitation of the FDRL strategy in the previous chapter was the filter that was used in order to suppress trading actions that would lead to unsuccessful trading. This chapter highlights the benefits of training a model with positional awareness to afford the agents a level of autonomy when trading, with the intention of removing the requirement for a hard-coded trading filter. To investigate this claim, this chapter explores an enhanced approach to model preparation by adding positional variables to the feature set and extending the time-frame upon which the models are trained.

The rest of this chapter is organised as follows: Section 5.2 explains the methodology used to conduct the experiments, Section 5.3 discusses the experimental setup and the results are then presented in Section 5.4. The results are interpreted in Section 5.5 and finally Section 5.6 summarises the conclusions of the study.

5.2 Methodology

The following methodology takes the same fundamental approach to that of FDRL, so the following section focuses on the notable differences. First, in Section 5.2.1 the changes to the data preparation phase are discussed. Next, in Section 5.2.2 the changes to the state representation are discussed, followed by the changes to the action space and reward functions in Section 5.2.3. Finally in Section 5.2.4, the performance metrics used to evaluate the framework are discussed.

5.2.1 Data Preparation

The same sliding window mechanism was used for PADRL with an adaption to the size of the windows as shown by the diagram of the first two windows in Figure 5.1. Each window consists of a training, validation and test set of duration 16, 4 and 4 weeks respectively, further details will be provided in Section 5.3. As with

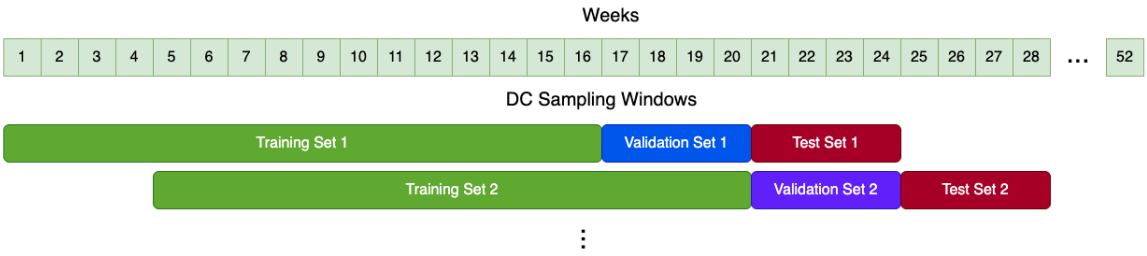

Figure 5.1: Diagram of first 2 sampling windows

FDRL, each window is used to train a separate agent which is optimised on the validation set and tested on the test set.

Prior to being fed into the neural network within the agent, the data itself undergoes normalisation to enhance the convergence and stability of the training process. Z-score normalisation (standardisation) was applied to guarantee that each feature possesses a mean value of 0 and a standard deviation of 1. Equation 5.1 was employed to compute the number of standard deviations by which a value (x) deviates from the mean (μ). The μ and σ values used in Equation 5.1 were derived from the training set of each window.

$$z = \frac{x - \mu}{\sigma} \tag{5.1}$$

where:

- z: number of standard deviations from the mean.
- x: the real value of the variable to be normalised.
- μ: mean.
- σ: standard deviation.

5.2.2 State Representation

As with the FDRL system, the state represents the market environment observed by the agent at each time step, represented as an array of numeric values. The market features employed in this study are represented using the same DC-based

indicators in FDRL. Alongside information on market behaviour, and as one of the key differences from the FDRL trading strategy, the agent was given awareness of its current position in the market. To address the lack of positional awareness, the agent was supplied with a set of four extra positional values to enhance the insights into the direct implications of its actions on market position.

The state features contain the following elements. Firstly a position value to signify whether the agent is in no position, a long position, or a short position (denoted by the discrete values 0, 1, and -1, respectively). Secondly, the initial balance of 100 base currency units minus the current balance to represent the current profit of the agent. Thirdly, the potential return if the agent were to exit the current trade at the current time step. Finally, the spread was also added to the input feature space to provide a dynamic measure of market volatility, a factor that often correlates with the low volatility periods that FDRL made use of. For PADRL spread was just used as an indicator and not a constraint, all trades for PADRL in this chapter are still subject to the 0.035% fixed transaction cost.

To comply with the requirements of the reinforcement learning algorithm, the feature set was bounded between the two values of -2 and 2. This constraint aided in the convergence and stability of training the neural network within the agent as all values were kept within a manageable scale to ensure faster and more stable network convergence. Since Z-score normalisation was applied to normalise market features, this ensured that a significant proportion of the data falls within the ± 2 bounds, minimising information loss for values outside these bounds while maintaining the crucial tight bounds for network training. The state features, including current position, balance, potential return and spread, were similarly bound by these values. While current position and potential return were minimally affected by these bounds, the balance, due to its subtraction for profit representation, experiences compression into -2 or 2, resulting in some information loss.

The formal structure of the state space $\mathcal{S}$ in PADRL differs from FDRL through

the explicit inclusion of positional information and lack of lag value. Each state $s_t \in \mathcal{S}$ is constructed as:

$$s_t = [f_t, p_t] \in \mathbb{R}^{d+k} \tag{5.2}$$

where $f_t \in \mathbb{R}^d$ represents the market feature vector at the current time step t with dimensionality $d = 30$ (comprising the same DC-based indicators as FDRL), and $p_t \in \mathbb{R}^k$ represents the positional feature vector with dimensionality $k = 4$. The positional vector is defined as $p_t = [pos_t, bal_t, ret_t, spread_t]$, where $pos_t \in \{-1, 0, 1\}$ encodes the current position, $bal_t \in [-2, 2]$ represents the balance deviation from initial capital, $ret_t \in [-2, 2]$ represents the potential return from the current trade, and $spread_t \in [-2, 2]$ captures the current spread value. The complete state representation therefore has dimensionality $d+k = 30+4 = 34$. Note that this does not have the same state space lag as FDRL which resulted in the multiplication by 5. It is assumed that the information required to trade successfully is held within the inherently period-based DC indicators.

The inclusion of positional features fundamentally alters the nature of the state space by making it explicitly dependent on the agent's trading history, not just market conditions. In FDRL, the agent lacked direct awareness of its current position and had to infer this information implicitly from the sequence of its past actions and their effects on subsequent states. By contrast, PADRL provides this information directly as part of the state observation, enabling the agent to make position-aware decisions. For example, the agent can now explicitly condition its policy on whether it is currently long, short, or in no position, and can directly observe the profitability of its current position through ret_t. This addresses a key limitation of FDRL where the positional unawareness sometimes led to suboptimal repeated actions. The bounded nature of the state features (constrained to $[-2, 2]$) ensures numerical stability during neural network training, though this introduces some information compression, particularly for balance values that exceed these bounds during periods of substantial profit or loss.

5.2.3 Action and Reward Definition

A discrete action space was employed, limiting the agent's choices to either buying or selling (represented by 1 and 0, respectively). This choice was made to prevent the agent from holding no position throughout an episode, a behaviour observed in preliminary testing of FDRL. Holding no position for the entire episode resulted in inactive systems that refrained from trading to avoid immediate negative positions due to transaction costs. To address this, if the agent wishes to maintain its current position, it returns the action corresponding to that position (e.g. returning 1 to hold a buy position). This resulted in a slightly different dynamic from FDRL as the PADRL system was aware of its current position, so it had the tools required to hold positions by producing the same action consecutively if it determined this was required by the current state information.

The reward function adopted in this study was the profit realised by the agent upon exiting a position. This specific reward function was chosen to incentivise both active and profitable trading, as in FDRL. By tying the reward to realised profit, the system encouraged the agent to engage in trades more actively, reinforcing a learning strategy that aligns with the goal of achieving profitable outcomes. Other reward functions were tested on the validation sets, such as Calmar ratio and final balance, but these were unable to produce the results observed for the profit reward function. This was likely due to the increased sparsity of the other reward signals during training, which made it more difficult for the agent to learn consistent behaviour.

5.2.4 Trading Performance Metrics

In order to fairly compare PADRL with FDRL and the technical analysis benchmarks, the same performance metrics were used as in FDRL. The Total Return (see Equation 4.3) served as a measure to quantify the ultimate return generated by the strategies. The evaluation of risk was measured through the examination of the Maximum Drawdown (see Equation 4.4). The Calmar ratio (see Equation 4.5)

was employed to aggregate both total return and maximum drawdown, providing a risk-adjusted return metric.

5.3 Experimental Setup

5.3.1 Data

The raw tick data for the fourteen currency pairs used in the results section is the same as the data used for FDRL in order to keep the comparison of the two methodologies fair. To reiterate the details, the data was sourced from TrueFX.com[1] and data for currency pairs involving USD spans from 1st January 2022, to 31st December 2022 with all other pairs ranging from 1st May 2022, to 30th April 2023.

The raw tick data underwent sampling across eight DC sampling thresholds, ranging from 0.015% to 0.029%. The duration of each window was extended by 20 weeks from the prior FDRL approach to 24 weeks with 16 training weeks, 4 validation weeks and 4 testing weeks. The window size was increased to increase the number of samples per window and therefore the amount of experience the agents could gain before trading the test set, resulting in the creation of 7 windows per threshold per currency pair. This new approach created a different view on the same tick data used in FDRL and therefore a total of 784 datasets are generated from 14 currency pairs, 8 DC thresholds and 7 windows.

5.3.2 Hyperparameter Tuning

Each model was trained for a duration of 3,000,000 time steps, 15 times as many as in FDRL. Performance tracking of training models was implemented using the validation set to assess trading performance every 50,000 timestamps. The choice of training over a larger number of time steps was motivated by the objective to identify the best policy by spending more time searching the policy space. The value of 3,000,000 time steps emerged from testing on a reduced sample size with

[1]https://www.truefx.com/truefx-historical-downloads/

the goal of identifying the optimal maximum number of timestamps while still maintaining a realistic training time (see Section 5.3.3 for more details).

The hyperparameters `batch_size` and `n_epochs` were also identified through a grid search on a subset of validation sets. It was observed that a `batch_size` of 65,536 and an `n_epochs` value of 10 produced the most favourable results without introducing impractical computational overhead.

Position size was another variable that was optimised and 10% of the total balance was found to produce the most effective trading behaviour on the validation set without incurring significant losses. This recognises that a larger balance could lead to a compounding effect detrimental to the negatively performing strategies.

5.3.3 Model Optimisation

Given the inherent instability associated with training using deep reinforcement learning, where performance may fluctuate while the algorithm explores different policy spaces, a strategy was implemented to address this behaviour. To mitigate the impact of this training volatility, a second validation environment was integrated into the training process, using an evaluation callback[2] within the `Stable Baselines 3` (SB3) reinforcement learning library [165] in Python.

This callback mechanism was activated every 50,000 time steps and used the model at that stage of the training process to simulate trading within the validation set. The final profit gained over the simulation of the validation set was recorded. When the validation simulation outperformed the highest validation balance achieved in prior runs, the new model was noted as the current best model. The model noted as the best performing model at the end of the training process was subsequently employed on the test set. This approach helps to navigate the inherent instability of DRL training and ensured the best model was used for testing

[2]A callback refers to a function that enables the interruption of the standard training process to conduct custom calculations.

5.3.4 Benchmarks

To keep the benchmarks consistent, PADRL is benchmarked against the same Buy and Hold (B&H), Moving Average Crossover (MAC), and Relative Strength Index (RSI) strategies as in FDRL, with the only difference being that it is now tested using a 0.035% fixed transaction cost level to ensure transaction costs match those applied to PADRL.

The FDRL system described in the previous chapter is also used to benchmark the PADRL strategy and involves training a DRL agent using data sampled under the DC framework as described in 4.1. FDRL is the earlier iteration of PADRL and therefore does not include the positional awareness variables that have been described for PADRL. By employing the FDRL system as a benchmark, an evaluation of the performance of a successful DC-based DRL system without the specific elements introduced in the PADRL system is facilitated, thereby illustrating the advantages of these novel elements. In order to make the comparison between FDRL and PADRL a fair one, FDRL was trained and tested using the 0.035% fixed transaction cost level. This is the same fixed transaction cost level used by PADRL during testing.

5.4 Results

The following results compare the performance of the novel PADRL trading system to the benchmarks detailed in Section 5.3.4. These benchmarks include the passive buy and hold strategy, fixed-interval technical analysis approaches such as MAC and RSI, and the FDRL, the first DC-based deep reinforcement learning strategy introduced in this thesis. Performance metrics, including total return, maximum drawdown, and the Calmar ratio, are used to evaluate and compare the efficacy of the PADRL strategy and benchmarks across fourteen currency pairs sampled at eight different DC thresholds. A key aspect of the comparison is the robustness of each strategy, as indicated by the maximum transaction costs that can be sustained

while maintaining positive returns. The higher the transactions costs that can be handled while still producing positive results, the more effective the trading strategy is as it is considered more robust. The transaction costs have therefore been increased by 0.01% to 0.035% from the previous FDRL results. The 0.035% transaction cost level has been applied across all benchmarks and PADRL to ensure consistency in testing conditions.

The testing period has an extended lead time of 16 weeks for the initial training set to accommodate changes in window sizes from the previous chapter, ensuring a fair and consistent comparison among PADRL and the benchmarks. The significance of each trading strategy was tested using the Friedman test and the Conover post hoc test. The null hypothesis states that there is no statistically significant difference in performance across the trading strategies and results of this significance test are provided in Tables 5.1a, 5.2a and 5.3a.

Total Return Table 5.1 presents the total return achieved for each currency pair across various DC thresholds for both PADRL (denoted as PA) and FDRL (denoted as F). The table also includes comparative results for the buy and hold strategy, MAC, and RSI benchmarks. While PADRL delivers considerable returns, the magnitude of these returns is generally lower than those observed in the previous FDRL results displayed in Table 4.2a. The highest return among all strategies is 3291.23%, achieved by PADRL on the EUR/JPY currency pair at a DC threshold of 0.027%. Among the currency pairs, EUR/JPY, EUR/GBP, and EUR/CHF demonstrate the largest returns for PADRL. This behaviour is consistent with the trends observed for FDRL at both the 0.025% and 0.035% transaction cost levels. PADRL reports only one negative total return across all pair-threshold combinations for EUR/USD at 0.015%. All remaining currency pairs produce positive returns across all other DC thresholds with all pairs apart from CHF/JPY, EUR/USD, USD/CAD and USD/JPY generating total returns exceeding 20% on every DC threshold.

Table 5.1 reveals that PADRL and FDRL show similar performance patterns,

Table 5.1: Total Return (%) by DC threshold (θ) for PADRL (PA) and FDRL (F). B&H, MAC, and RSI strategies are fixed interval based strategies, so only a single value is presented per currency pair. The best value per threshold is denoted in boldface and best value per threshold is underlined.

Pair / θ	0.015%		0.017%		0.019%		0.021%		0.023%		0.025%		0.027%		0.029%		B&H	MAC	RSI
	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F			
AUD/JPY	20.76	-0.58	26.28	2.40	22.75	1.52	23.98	2.74	29.79	3.32	25.7	6.85	**36.4**	8.72	33.58	7.74	-6.80	-2.71	-0.01
AUD/USD	23.14	10.33	56.25	18.46	72.53	26.64	**73.23**	30.43	27.97	27.91	57.69	35.38	43.83	33.61	59.42	37.64	-4.73	-5.83	0.42
CAD/JPY	22.88	-1.34	47.87	-0.56	34.68	-0.58	27.3	0.46	37.11	0.19	**53.62**	1.77	49.15	2.70	31.8	3.85	-8.17	-3.66	2.81
CHF/JPY	8.27	-0.18	3.59	-0.95	1.14	0.15	6.61	0.13	9.03	0.13	6.36	-0.73	9.64	2.01	**19.75**	3.33	1.71	4.12	0.55
EUR/CHF	272.62	340.83	135.00	440.89	127.02	486.81	107.5	916.20	1160.77	652.43	**1222.29**	741.49	755.38	612.47	400.13	1167.62	1.76	-5.99	1.37
EUR/GBP	81.55	158.85	110.30	140.81	167.55	299.59	326.36	284.40	615.3	368.17	**773.79**	394.80	233.02	467.90	129.17	463.36	0.96	1.00	-0.73
EUR/JPY	199.53	377.66	385.09	579.84	548.50	801.69	409.69	1320.83	440.28	1754.29	331.57	981.15	**3291.23**	1918.55	369.41	1583.91	3.04	1.28	1.36
EUR/USD	-1.20	3.79	13.02	10.03	31.35	12.21	20.88	10.63	40.57	12.06	42.62	12.06	**42.95**	12.00	28.84	14.29	0.56	4.16	-0.53
GBP/JPY	36.77	5.37	37.57	1.50	45.81	7.95	50.45	11.53	53.02	11.58	48.75	16.88	**53.96**	16.98	29.67	24.16	2.30	6.69	6.19
GBP/USD	33.78	25.01	47.00	17.11	53.79	27.63	**68.88**	21.52	66.04	52.78	66.05	46.66	44.4	54.02	61.04	58.27	-3.18	3.48	-0.97
NZD/USD	113.79	36.59	70.39	17.10	145.68	46.74	211.38	60.85	**278.16**	81.53	53.63	67.12	117.0	75.39	226.66	94.79	-1.59	-8.73	2.56
USD/CAD	12.75	10.27	13.48	9.33	10.56	11.68	11.66	12.22	23.31	13.26	23.09	11.77	12.79	13.16	**27.58**	14.16	5.95	-3.96	-0.73
USD/CHF	72.42	49.84	133.32	86.07	93.71	90.16	139.8	89.60	86.02	131.48	127.1	142.57	**293.2**	138.51	268.9	134.51	-3.99	6.22	-0.94
USD/JPY	33.66	0.53	13.62	6.94	29.35	7.70	**41.17**	8.09	25.22	11.29	32.54	15.51	28.59	13.53	30.76	16.06	4.06	13.17	-8.33

(a) Statistical test results for Total Return, according to the non-parametric Friedman test with the Conover's post hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

Friedman test p-value		**3.573e-57**
	Ave. Rank	p_{Con}
PADRL (c)	1.28	-
FDRL	2.13	**1.034e-04**
MAC	3.69	**5.217e-60**
RSI	3.85	**5.605e-67**
B&H	4.05	**1.834e-69**

with the EUR/JPY, EUR/GBP, and EUR/CHF currency pairs generating the largest total returns among all pairs. Among the 24 combinations of these three currency pairs across eight DC thresholds, FDRL outperforms PADRL in 17 instances. None of these instances represent the highest return for the respective currency pair, as shown by the bold values in the table. These three currency pairs are characterised by low-volatility oscillations, which is a market environment FDRL is known to thrive in. The fact that PADRL performs competitively on these three currency pairs indicates that it is capable of learning a similar approach to trading as FDRL. For the remaining 11 currency pairs, FDRL outperforms PADRL in only six out of 88 pair-threshold combinations and does not produce the highest return for any currency pair across all eight DC thresholds. These findings demonstrate PADRL's greater consistency compared to FDRL. This advantage suggests that the inclusion of positional variables in PADRL enables it to identify profitable trading opportunities outside the periods filtered by FDRL's rule-based approach.

As exposed by Figure 5.2, PADRL shows a weaker correlation between total return and DC threshold compared to FDRL. While the maximum total return for each currency pair occur within the DC threshold range of 0.021% to 0.029%, the overall correlation of total return with DC threshold is generally weak. On average, only three currency pairs (AUD/JPY, EUR/USD and USD/CHF) achieve a correlation coefficient of 0.7 or higher for PADRL with statistical significance observed for AUD/JPY ($p = 0.01$), EUR/USD ($p = 0.03$), and USD/CHF ($p = 0.02$). However, others such as AUD/USD ($p = 0.71$), GBP/JPY ($p = 0.73$), and USD/JPY ($p = 0.67$) demonstrate minimal correlation with coefficients below 0.2 and non-significant p-values. This contrasts with the FDRL results, where most currency pairs exhibit strong correlations with DC thresholds. For FDRL, all fourteen pairs show significant relationships ($p < 0.05$), typically achieving coefficients of 0.7 or higher. These differences suggest that PADRL's performance is less sensitive to variations in DC thresholds and is therefore a more robust trading framework.

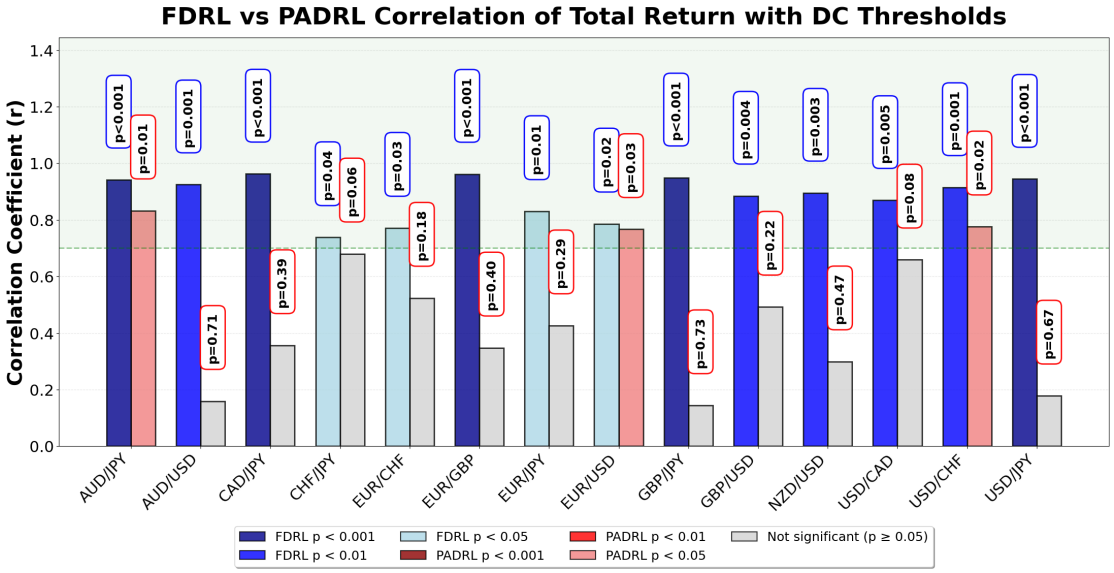

Figure 5.2: FDRL vs PADRL Total Return Correlation Coefficients

A comparison of PADRL and FDRL with the benchmark strategies shows the better performance of the DC and DRL-based approaches. Both PADRL and FDRL consistently outperform the passive buy-and-hold strategy as well as the active technical analysis strategies that rely on fixed-interval sampling. Across all 14 currency pairs and 8 DC thresholds, the MAC strategy outperforms PADRL on only three occasions, all involving CHF/JPY. RSI and the buy and hold strategy both outperform PADRL just once, on EUR/USD at the DC threshold of 0.015%. Out of 112 pair-threshold combinations, instances where PADRL is outperformed by a benchmark strategy are rare. These results further demonstrate the robustness of PADRL when compared to both active and passive benchmark strategies.

The significance of total returns and the ranking of each trading strategy are detailed in Table 5.1a. This analysis demonstrates that PADRL significantly outperforms all benchmarks, including its predecessor FDRL. PADRL achieves an average ranking of 1.28, significantly better than the average rankings of 2.13 for FDRL, 3.69 for MAC, 3.85 for RSI and 4.05 for the buy and hold strategy. These results show the efficacy of the PADRL framework, demonstrating that the integration of deep reinforcement learning (DRL) with directional change (DC) sampling

outperforms traditional technical analysis and the passive buy-and-hold strategy. The results also show that the introduction of positional variables enhances the autonomous DRL system, leading to more favourable returns. The non-parametric Friedman test produces a value of 1.222e-67, indicating highly significant results. All p-values from the post hoc Conover test comparing PADRL to the benchmarks are also below the 0.05 significance threshold, confirming PADRL's superior performance across all benchmarks.

Maximum Drawdown The potential benefits of PADRL, much like those of FDRL, are highly promising, however it is essential to evaluate these benefits in conjunction with the associated risks. Table 5.2 illustrates the risk profiles of various benchmark strategies tested in comparison to PADRL. The maximum drawdown results for PADRL, calculated across all currency pairs and DC thresholds, reveal that the largest observed drawdown was 23.04%. Only 21 out of 112 pair-threshold combinations report a maximum drawdown exceeding 5%, and just 11 are greater than 10%. The distribution of maximum drawdown values exhibits no clear pattern across currency pairs. Many currency pairs display sporadic increases in maximum drawdown at specific DC thresholds, but these deviations often revert back to the mean for the majority of thresholds.

PADRL demonstrates comparable maximum drawdown levels to FDRL but with greater stability. As discussed in the previous chapter, FDRL often generates either very small or significantly large maximum drawdowns. This variability means FDRL is less robust to different currency pairs and DC thresholds. This behaviour was observed when FDRL was trained and tested under a fixed transaction cost threshold of 0.035%. While FDRL outperforms PADRL in 50 out of the 112 pair-threshold combinations, these instances are predominantly characterised by very small maximum drawdowns, suggesting that FDRL is efficiently trading the small oscillations in price it tends to generate most of its profit from. This behaviour allows FDRL to perform well in those scenarios while avoiding substantial drawdowns. FDRL's performance is inconsistent, as it also experiences significant

Table 5.2: Maximum Drawdown (%) by DC threshold (θ) for PADRL (PA) and FDRL (F). MAC, and RSI strategies are fixed interval based strategies, so only a single value is presented per currency pair. B&H cannot be calculated, as it only performs a single trade. The best value per threshold is denoted in boldface and best value per threshold is underlined.

Pair / θ	0.015%		0.017%		0.019%		0.021%		0.023%		0.025%		0.027%		0.029%		MAC	RSI
	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F		
AUD/JPY	1.83	0.58	0.37	0.11	1.70	0.25	3.66	0.08	1.39	0.07	2.98	0.05	1.04	0.08	0.56	**0.03**	11.11	4.01
AUD/USD	3.25	13.94	1.56	23.90	4.15	15.35	1.58	8.59	0.59	7.11	1.26	8.88	0.76	5.39	**0.52**	6.75	14.72	2.46
CAD/JPY	7.93	1.34	0.58	0.56	12.99	0.58	11.38	0.13	9.22	0.22	0.65	0.19	0.6	**0.04**	17.31	**0.04**	14.93	5.07
CHF/JPY	1.19	0.18	5.25	0.95	8.48	0.07	4.92	0.07	4.71	0.04	5.69	0.73	5.44	**0.03**	1.73	0.04	9.48	4.22
EUR/CHF	3.13	41.83	13.59	20.63	0.80	37.59	**0.35**	8.88	7.45	7.60	1.35	5.33	1.44	3.33	0.73	2.13	7.26	1.54
EUR/GBP	15.90	76.48	13.61	42.82	1.03	15.64	6.29	9.26	1.36	8.58	0.54	3.78	1.57	2.87	**0.35**	3.09	4.73	2.74
EUR/JPY	1.84	0.61	3.11	0.33	3.34	0.16	2.17	0.30	11.23	0.06	0.3	0.06	2.42	0.04	2.52	**0.03**	12.66	3.12
EUR/USD	13.51	9.03	5.05	6.51	5.04	6.58	2.41	8.65	**0.8**	4.40	2.32	7.92	1.58	2.9	1.35	2.68	6.16	3.94
GBP/JPY	1.09	0.06	2.07	0.08	3.01	0.05	0.96	0.06	3.85	0.03	3.79	0.04	3.98	0.03	11.59	**0.02**	6.95	3.97
GBP/USD	23.04	6.03	3.76	15.11	2.89	16.20	17.56	6.27	1.85	5.14	0.91	4.58	**0.81**	2.38	2.59	4.31	10.30	6.85
NZD/USD	1.57	25.80	7.54	15.61	2.65	15.46	4.58	9.73	2.21	15.59	**0.7**	14.64	0.9	20.93	2.07	11.09	13.30	1.72
USD/CAD	0.64	3.97	0.75	2.06	0.60	2.71	1.05	3.45	0.91	5.87	**0.36**	4.15	0.68	5.5	0.93	1.08	5.69	2.78
USD/CHF	1.52	31.50	1.71	13.91	1.51	4.26	1.89	13.67	1.23	7.84	0.98	3.36	0.6	2.37	**0.34**	3.29	6.39	3.93
USD/JPY	4.34	0.08	3.54	0.14	4.07	0.09	3.47	0.06	3.7	0.06	3.51	0.03	3.3	0.04	3.06	**0.02**	8.17	11.09

(a) Statistical test results for Maximum Drawdown, according to the non-parametric Friedman test with the Conover's post hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface. B&H is not included as it only performs a single complete trade (buy on the first day and sell on the last), and as a result maximum drawdown cannot be defined.

Friedman test p-value		**3.479e-24**
	Ave. Rank	p_{Con}
PADRL (c)	1.84	-
FDRL	2.26	0.1767
RSI	2.33	**1.510e-02**
MAC	3.57	**2.755e-30**

drawdowns in certain cases. For example, with EUR/GBP at a 0.015% DC threshold, FDRL records a maximum drawdown of 79.48%. Large drawdowns are also observed with EUR/CHF at 0.015% (41.83%), EUR/GBP at 0.017% (42.82%), and EUR/CHF at 0.019% (37.59%). These instances highlight the susceptibility of FDRL to extreme drawdowns under specific conditions, demonstrating the value of adding positional awareness to the trading framework as in PADRL.

An analysis of the correlation between maximum drawdown and the DC threshold for PADRL and FDRL, as shown in Figure 5.3, reveals significant variability in correlation coefficients for PADRL across different currency pairs. For pairs such as AUD/USD ($p = 0.04$), EUR/GBP ($p = 0.01$), EUR/USD ($p = 0.02$), USD/CHF ($p = 0.006$), and USD/JPY ($p = 0.01$), statistically significant strong negative correlations were observed. This indicates that increasing the DC threshold for these pairs tends to reduce maximum drawdowns, which in turn lowers trading risk. In contrast, certain currency pairs such as AUD/JPY ($p = 0.81$), CAD/JPY ($p = 0.78$), CHF/JPY ($p = 0.91$), EUR/JPY ($p = 0.98$), and USD/CAD ($p = 0.81$), show negligible correlation in either direction with non-significant p-values ($p \geq 0.05$). This suggests that the application of PADRL to these pairs is not particularly sensitive to changes in DC sampling thresholds. GBP/JPY demonstrates a strong positive correlation, implying that higher DC thresholds lead to worse maximum drawdowns for this currency pair. The results for FDRL mostly align with patterns observed in the previous chapter under the 0.025% transaction cost level. For most currency pairs an increase in DC threshold results in a decrease in maximum drawdown. Exceptions to this rule exist however, with currency pairs such as CHF/JPY and USD/CAD deviating from this trend, reflecting inconsistencies in FDRL's sensitivity to DC sampling thresholds across different pairs.

A comparison of PADRL with technical analysis benchmarks reveals that PADRL outperforms both MAC and RSI in the majority of cases. RSI demonstrates significantly better performance than MAC, with a mean drawdown of 4.10% across all

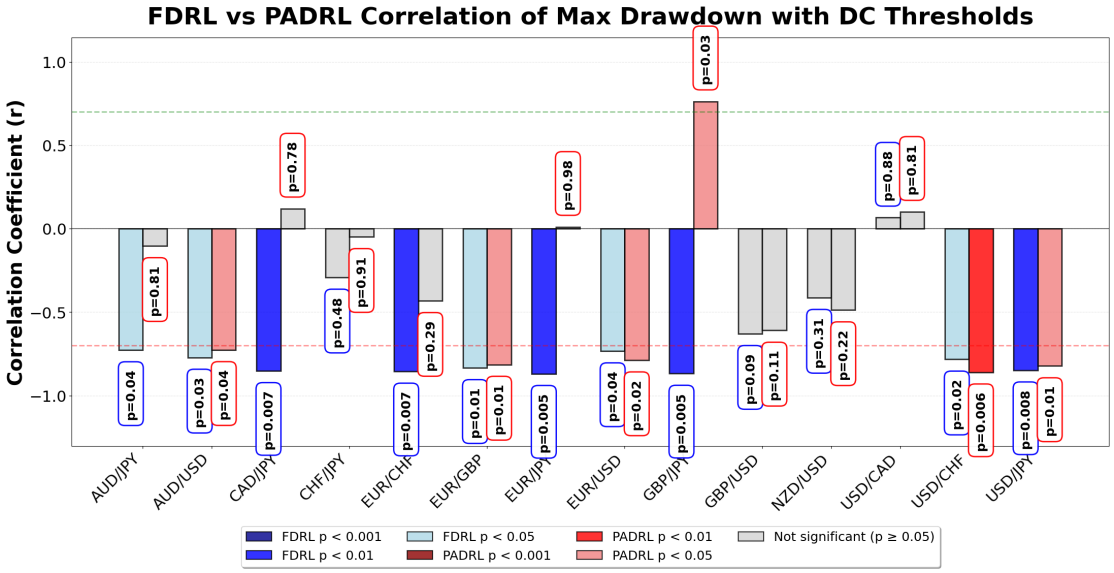

Figure 5.3: FDRL vs PADRL Maximum Drawdown Correlation Coefficients

currency pairs, compared to 9.42% for MAC. While RSI's relatively low maximum drawdowns position it as a competitive strategy, particularly against PADRL and FDRL, its performance is not enough to outperform them. As shown in Table 5.2a, RSI achieves an average rank of 2.33, this is a worse performance than both PADRL and FDRL which record average ranks of 1.84 and 2.26 respectively. MAC is the poorest performer in terms of maximum drawdown, with an average rank of 3.57. The results suggest that both PADRL and FDRL are more effective for managing risk than the technical analysis strategies.

The significance of the maximum drawdown results are summarised in Table 5.1a. The average rankings show that the inclusion of positional variables enhances the performance of the DRL algorithm on the testing data. The non-parametric Friedman test confirms the significance of these results, producing a p-value of 3.479e-24. By applying the post hoc Conover test with PADRL as the control algorithm, it was observed that PADRL significantly outperforms both RSI and MAC strategies at the 5% significance level, demonstrating the superior effectiveness of PADRL in managing drawdowns compared to these benchmarks. Despite the fact that PADRL achieves a better average ranking than FDRL, the Conover post hoc test resulted in a p-value of 0.1767, indicating that PADRL does not

significantly outperform FDRL at the predetermined 5% significance level. This suggests that while PADRL's performance is less risky, its advantage over FDRL in terms of maximum drawdown is not statistically conclusive.

Calmar Ratio The Calmar ratio is used to evaluate the risk-return trade-off of each trading strategy. Table 5.3 presents the Calmar ratio for each of the four strategies. EUR/JPY, EUR/CHF, and EUR/GBP are shown as the best-performing pairs when applying the PADRL strategy. This performance is largely due to the high returns demonstrated by these pairs, as shown in Table 5.1, combined with the moderate maximum drawdown levels reported in Table 5.2. EUR/JPY achieves the highest Calmar ratio values for PADRL across DC thresholds of 0.015%, 0.017%, 0.019%, and 0.027%, with corresponding values of 108.71, 123.79, 164.12, and 1359.31. EUR/CHF records the highest ratios at thresholds of 0.021% and 0.029%, with values of 311.39 and 550.54 respectively, while EUR/GBP performs best at thresholds of 0.023% and 0.025% with values of 451.84 and 1430.57. Among these thresholds, 0.025% stands out as producing the best Calmar ratio for PADRL across four separate currency pairs. Thresholds of 0.023% and 0.027% generate the highest ratios for one pair each, and 0.029% does so for two.

When comparing PADRL to FDRL, it becomes evident that FDRL can produce highly variable outcomes, as mentioned in the previous chapter. One striking example is an extreme Calmar ratio of 58,529.73 observed at the 0.029% DC threshold where the agent implemented a similar unrealistic strategy to that interpreted for FDRL, but with extreme success, demonstrating that the high returns generated coupled with very small drawdowns can result in enormous Calmar ratios. These exceptionally high Calmar ratios are not consistent across all currency pairs and thresholds when FDRL is trained and evaluated at the 0.035% transaction cost level. Negative Calmar ratios are reported for AUD/JPY, CAD/JPY, and CHF/JPY at thresholds of 0.015%, 0.017%, and 0.019% respectively. PADRL however, demonstrates more stability, generating only a single negative Calmar

Table 5.3: Calmar Ratio by DC threshold (θ) for PADRL (PA) and FDRL (F). MAC, and RSI strategies are fixed interval based strategies, so only a single value is presented per currency pair. B&H cannot be calculated, as it only performs a single trade. The best value per threshold is denoted in boldface and best value per threshold is underlined.

Pair / θ	0.015%		0.017%		0.019%		0.021%		0.023%		0.025%		0.027%		0.029%		MAC	RSI
	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F		
AUD/JPY	11.32	-1.00	71.03	21.73	13.38	6.18	6.55	34.93	21.5	45.13	8.62	150.97	35.01	107.44	60.29	**226.93**	-0.24	-0.00
AUD/USD	7.11	0.74	36.10	0.77	17.46	1.74	46.25	3.54	47.45	3.92	45.96	3.98	57.98	6.24	**114.13**	5.58	-0.40	0.17
CAD/JPY	2.88	-1.00	82.45	-1.00	2.67	-1.00	2.40	3.56	4.02	0.85	83.02	9.55	82.08	68.13	1.84	**109.91**	-0.25	0.55
CHF/JPY	6.94	-1.00	0.68	-1.00	0.13	2.16	1.34	1.81	1.92	3.52	1.12	-1.00	1.77	58.53	11.42	**78.68**	0.43	0.13
EUR/CHF	87.19	8.15	9.94	21.37	159.49	12.95	311.39	103.18	155.89	85.80	**906.89**	139.08	525.73	184.11	550.54	547.72	-0.83	0.89
EUR/GBP	5.13	2.08	8.10	3.29	162.77	19.16	51.88	30.70	451.84	42.90	**1430.57**	104.44	148.32	163.01	372.73	149.75	0.21	-0.27
EUR/JPY	108.71	620.33	123.79	1767.82	164.12	5036.20	188.57	4366.39	39.22	27656.83	1092.77	16215.10	1359.31	54476.93	146.66	**58529.73**	0.10	0.44
EUR/USD	-0.09	0.42	2.58	1.54	6.21	1.86	8.65	1.23	**50.47**	2.74	18.34	1.52	27.26	4.13	21.36	5.33	0.68	-0.13
GBP/JPY	33.81	94.05	18.12	19.68	15.21	167.03	52.41	190.56	13.77	345.03	12.85	458.78	13.56	632.72	2.56	**1159.53**	0.96	1.56
GBP/USD	1.47	4.14	12.50	1.13	18.60	1.71	3.92	3.43	35.66	10.26	**72.89**	10.20	55.1	22.70	23.57	13.52	0.34	-0.14
NZD/USD	72.45	1.42	9.33	1.10	54.90	3.02	46.12	6.25	125.62	5.23	76.69	4.59	**130.12**	3.60	109.44	8.55	-0.66	1.49
USD/CAD	19.88	2.59	17.91	4.52	17.61	4.31	11.09	3.54	25.58	2.26	**64.59**	2.83	18.81	2.39	29.77	13.06	-0.70	-0.26
USD/CHF	47.73	1.58	77.88	6.19	62.04	21.19	74.16	6.55	69.93	16.76	129.13	42.43	488.46	58.47	**780.21**	40.87	0.97	-0.24
USD/JPY	7.76	6.55	3.85	51.27	7.22	88.22	11.86	142.43	6.82	189.26	9.27	443.83	8.66	322.17	10.05	**959.49**	1.61	-0.75

(a) Statistical test results for Calmar ratio, according to the non-parametric Friedman test with the Conover's post hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface. B&H is not included as it only performs a single complete trade (buy on the first day and sell on the last), and as a result maximum drawdown and consequently Calmar ratio cannot be defined.

Friedman test p-value		**4.700e-50**
	Ave. Rank	p_{Con}
PADRL (c)	1.37	-
FDRL	1.80	**4.139e-03**
RSI	3.35	**2.038e-65**
MAC	3.48	**3.234e-68**

ratio of -0.09 for EUR/USD at the 0.015% DC threshold across all 14 currency pairs and 8 DC thresholds.

The negative Calmar ratios for FDRL are all observed to be -1.0, indicating that the total return of FDRL on these pair-threshold combinations equals the maximum drawdown. This scenario typically arises when the agent engages in minimal trading activity, resulting in marginal losses due to its reluctance to execute trades. When assessing Table 5.3 in the context of Table 5.1, it is evident that in all instances where the Calmar ratio is -1.0, the corresponding total return remains below 2%. This pattern supports the limited effectiveness of FDRL in these specific cases.

An analysis of the correlation between currency pairs and DC thresholds for the Calmar ratio, as shown in Figure 5.4, reveals different patterns for PADRL and FDRL. For PADRL, there is a weak to moderate correlation between the Calmar ratio and DC thresholds across all currency pairs except GBP/JPY. Statistically significant positive correlations are observed for AUD/USD ($p = 0.006$), EUR/CHF ($p = 0.03$), NZD/USD ($p = 0.05$) and USD/CHF ($p = 0.02$), indicating that for some pairs, an increase in the DC threshold is generally associated with an improvement in the Calmar ratio. However, several pairs including AUD/JPY ($p = 0.65$), CAD/JPY ($p = 0.70$), CHF/JPY ($p = 0.47$), and EUR/JPY ($p = 0.19$) show non-significant relationships. GBP/JPY demonstrates a relatively strong negative correlation ($p = 0.17$), though this relationship does not reach statistical significance, suggesting that higher DC thresholds tend to negatively impact the Calmar ratio for this pair. FDRL demonstrates a different pattern, with most currency pairs demonstrating a strong correlation (correlation coefficient of 0.8 or higher) between the Calmar ratio and DC thresholds. Specifically, ten of fourteen pairs show highly significant relationships (p ¡ 0.01), including AUD/JPY ($p = 0.003$), AUD/USD (p ¡ 0.001), EUR/GBP (p ¡ 0.001), EUR/JPY ($p = 0.003$), EUR/USD ($p = 0.007$), GBP/JPY ($p = 0.002$), GBP/USD ($p = 0.01$), USD/CHF ($p = 0.006$), EUR/CHF ($p = 0.01$), and USD/JPY ($p = 0.007$), while CAD/JPY

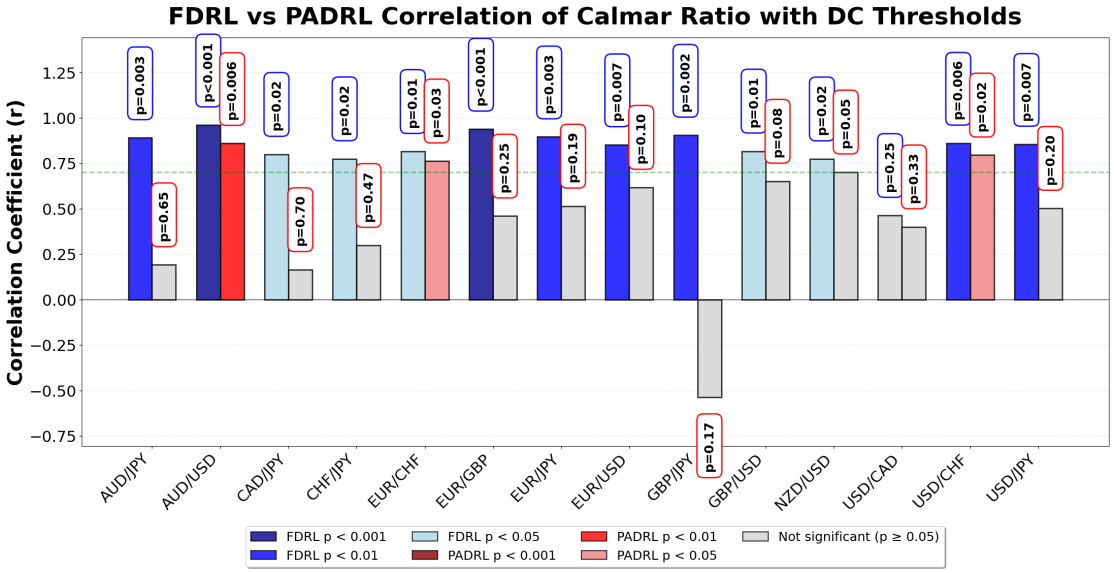

Figure 5.4: FDRL vs PADRL Calmar Ratio Correlation Coefficients

(p = 0.02), CHF/JPY (p = 0.02), and NZD/USD (p = 0.02) also demonstrate significant correlations. The only exception is USD/CAD (p = 0.25), which shows a moderate correlation. These findings suggest that PADRL shows variability and weaker correlations in its relationship between the Calmar ratio and DC thresholds but FDRL Calmar ratios have a consistently stronger correlation with the DC threshold across all currency pairs.

A comparison of PADRL results to the technical analysis benchmarks MAC and RSI reveals that the benchmarks produce a Calmar ratio greater than 1.0 in only three instances, MAC on USD/JPY and RSI on GBP/JPY and NZD/USD. In all other cases, PADRL consistently outperforms these technical analysis benchmarks, except for the single instance where PADRL produces a negative Calmar ratio (EUR/USD at 0.015%). Table 5.3a reports average ranks of 1.37 for PADRL and 1.80 for FDRL, compared to the significantly lower average ranks of 3.35 for RSI and 3.48 for MAC. This ranking supports the superior performance of PADRL over traditional technical analysis strategies across the tested scenarios.

Table 5.3a presents the results of the non-parametric Friedman test which results in a p-value of 4.700e-50, confirming the statistical significance of the findings. When comparing the performance of each strategy to PADRL, the Conover

post hoc test demonstrates that PADRL significantly outperforms all three benchmarks. The corresponding p-values are 4.139e-3 for FDRL, 2.038e-65 for RSI, and 3.234e-68 for MAC. These results clearly show the effectiveness of using positional variables in the development of the PADRL trading system. PADRL is a system which is fully autonomous and can significantly outperform both technical analysis benchmarks as well as its predecessor FDRL when comparing risk adjusted return performance using Calmar ratio.

Results Summary The analysis of results presented in Tables 5.1, 5.2, and 5.3, along with the corresponding significance tables 5.1a, 5.2a, and 5.3a, indicates that the proposed PADRL algorithm demonstrates superior performance compared to fixed-interval and technical analysis-based trading strategies. PADRL demonstrates that it is the best performing trading strategy on all three metrics by significantly outperforming B&H, FDRL, MAC and RSI on every metric apart from FDRL for maximum drawdown. In this one exception however PADRL still achieves a better average rank than FDRL, demonstrating that the introduction of positional variables and more advanced training methods to the FDRL result in the more effective trading framework of PADRL.

There are still some clear patterns in the PADRL results that were observed in FDRL, suggesting that PADRL learns a similar strategy to FDRL. Unlike FDRL, which depends on a rule-based filter, PADRL appears to have internalised this behaviour. This suggests it has developed an implicit representation that enables it to identify profitable trading opportunities independently. The next section investigates the trading strategies implemented by the PADRL across the 14 currency pairs and 8 DC thresholds.

5.5 Interpretation

One drawback of DRL methods is the fact that the resultant model is a blackbox, meaning the decision making process is not interpretable. To combat this issue

the decision made by the agent can be analysed to gain a form of intuition on the decision making process to deduce characteristics of the model's behaviour and hypothesise about why certain trades are entered.

Trade Win Rate Figure 5.5 visualises the win percentages across all the 14 currency pairs under all 8 DC thresholds as a heatmap. A notable trend is the strong and consistent performance of EUR-based pairs, such as EUR/CHF, EUR/GBP, and EUR/JPY, which maintain high win percentages across most thresholds. These pairs exhibit resilience to varying thresholds, suggesting that euro-based pairs may be more predictable or stable, making them strong candidates for approaches that prioritise more consistent returns.

On the other hand, certain pairs like CAD/JPY and CHF/JPY show weaker performance, with win percentages frequently below 50%, particularly at lower thresholds ($\theta = 0.019\%$). This indicates that these pairs may struggle in more volatile or less stable market conditions. USD/CHF and AUD/JPY perform better at higher thresholds ($\theta = 0.027\%$ and $\theta = 0.029\%$), which could imply that these pairs thrive in more stable conditions, where larger directional changes are more predictable. The change in performance as thresholds increase has also been analysed, where it can be seen that GBP/JPY demonstrates strong performance at $\theta = 0.021\%$ but sees a sharp decline at higher thresholds, indicating that it may be highly reactive to specific market shifts. In contrast, pairs like AUD/USD and AUD/JPY maintain relatively stable performance across different thresholds, reflecting a smoother trading profile and lower sensitivity to volatility.

Average Return Figure 5.6 compares the mean return per trade for the 14 different currency pairs across varying directional change thresholds. The EUR/JPY and EUR/GBP currency pairs consistently deliver relatively high average returns across most thresholds, particularly at the lower and mid-threshold ranges (0.015% to 0.025%). This suggests that these pairs tend to respond more favourably to smaller directional shifts, which could indicate greater sensitivity to short-term

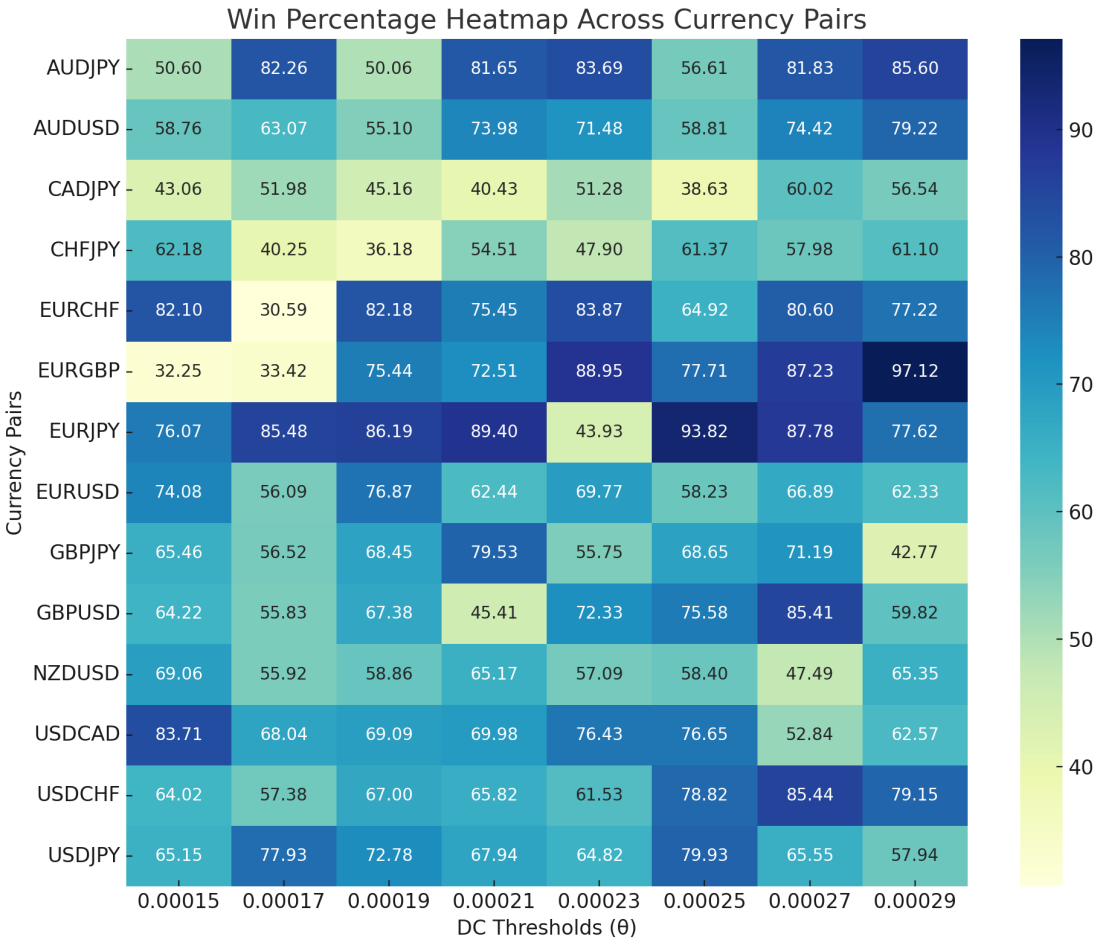

Figure 5.5: Trade Win Rate (%)

market movements. In contrast, pairs like EUR/USD and GBP/JPY show lower and more variable returns, suggesting they might be more stable or less volatile, requiring larger directional changes to generate significant returns.

Sensitivity to the DC threshold is more evident in pairs such as NZD/USD and USD/CHF, which exhibit moderate to high returns, particularly at higher thresholds (0.023% and 0.025%). This suggests that these currency pairs may require more pronounced market movements to yield higher returns. On the other hand, CHF/JPY shows a unique behaviour where returns hover near zero or even dip into negative values at 0.017% and 0.019% thresholds, indicating a potential change in market regime from the training data to the test set. This could point to underlying structural factors in the CHF/JPY pairing that make it less responsive to certain market changes or possibly more influenced by external macroeconomic factors.

A key takeaway from the chart is the wide variability in returns across currency pairs at different thresholds, implying that a one-size-fits-all trading strategy may not be optimal. Currency pairs such as EUR/CHF and AUD/JPY demonstrate sharp fluctuations in returns across thresholds, making them potentially profitable under specific market conditions but risky when the market is less volatile. In contrast, USD/JPY and CAD/JPY show more stable, consistent returns, which might appeal to more conservative trading strategies focused on minimising risk.

Trading Pattern PADRL agents follow the same trading pattern as the FDRL agents, meaning they rely on the short oscillations observed in the data to generate numerous trading opportunities in quick succession and then capitalise on these by quickly entering and exiting the positions. The introduction of positional variables and removal of the trading filter applied to FDRL means that the PADRL agents have much more trading freedom. This trading freedom however comes with drawbacks, as the PADRL agent must now identify these high profit potential periods itself, rather than assuming it is in one as with FDRL. The result of this setup mean that PADRL learns a similar strategy as FDRL for the periods of high profit

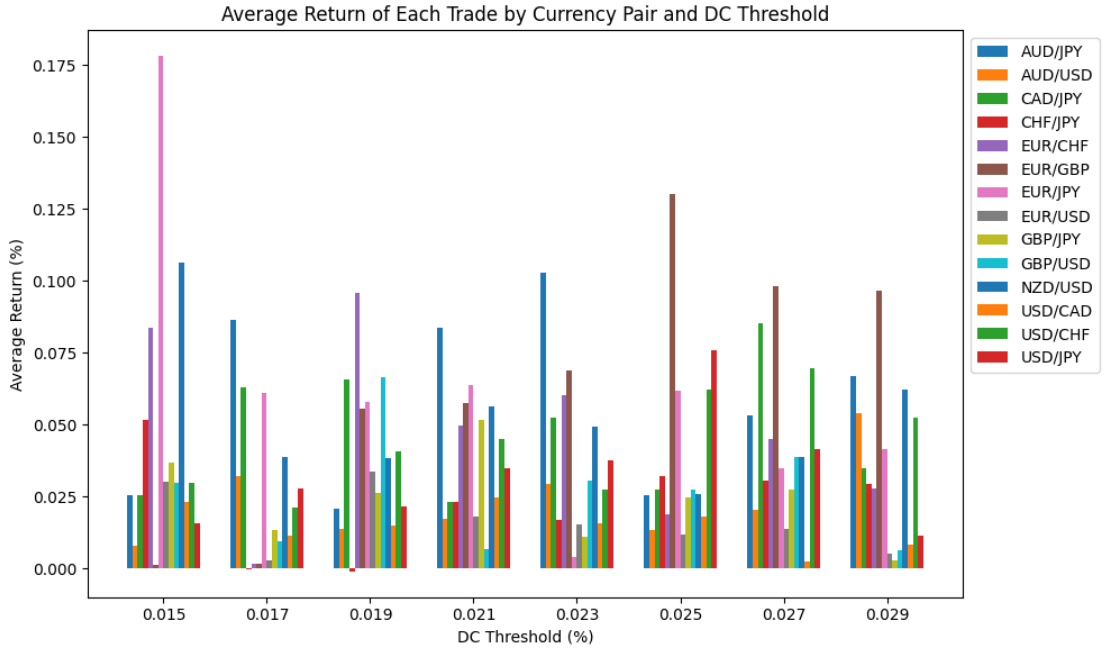

Figure 5.6: PADRL Mean Return per Trade

potential.

There is no clear goal with the trading of PADRL between periods of fast price oscillations, PADRL does however outperform FDRL at the 0.035% transaction cost level, therefore demonstrating that it is able to efficiently identify these periods of fast oscillations and make greater use of them and protect its balance enough during the other trading periods to produce these favourable results.

5.6 Summary

This chapter introduces the Positionally Aware Deep Reinforcement (PADRL) trading algorithm, a deep reinforcement learning based methodology for training agents to trade FX assets using data sampled with the DC sampling algorithm. It is the second model in a succession of DRL models that have been designed to trade high frequency currency pairs under a directional changes sampling paradigm. Each agent is trained per window across 14 different currency pairs and suggests either a buy or sell action at each DC event which is then executed in a simulated market environment. As before, transaction costs are modelled as a fixed cost of

0.035% of the position size, representing a combined approximation of commission fees and the bid-ask spread typically encountered in foreign exchange markets. As in the previous chapter, slippage is assumed to be negligible due to the high liquidity of the major currency pairs traded and is not explicitly modelled in this framework.

The predecessor to PADRL, the FDRL trading algorithm, demonstrated great performance as it generated extreme profit levels for most pairs across all DC thresholds. One drawback of the FDRL algorithm however was its trading filter, which only allowed the trained agents to trade when certain market conditions were met. The goal with PADRL was to take the basic framework of FDRL and improve it by making it more autonomous and more effective at a larger fixed transaction cost level.

Like FDRL, the PADRL strategy shows extremely positive returns across almost all 14 currency pairs. These high returns easily outperform all the benchmarks, including the passive Buy and Hold and active technical analysis benchmarks that are subject to the same fixed transaction costs, as well as the FDRL trading algorithm trained and tested on the new 0.035% fixed transaction cost level. These results suggest that the PADRL strategy is a much more efficient algorithm for trading, and provides evidence for the successful application of DC sampling and deep reinforcement learning to high frequency FX data. PADRL also exposes the fact that the introduction of positional variables negates the requirement for a trading filter, as used in FDRL, demonstrating that a fully autonomous set of agents can successfully trade in the FX market under the conditions described.

PADRL has demonstrated that the combination of DC sampling and DRL provides the tools needed in order to successfully trade in the high frequency foreign exchange environment. The conditions of these results however are not realistic enough to implement in a real trading scenario without significant adjustment due to the fixed transaction costs and market entry and exit frequency during

consolidation periods. The following chapter therefore focusses on the next model in this series of models, the Spread Aware Deep Reinforcement Learning (SADRL) trading algorithm. SADRL builds on the learnings of both PADRL and FDRL to build a trading framework under which to train DRL agents to trade using the bid-ask spread as opposed to fixed transaction costs to move closer to a more realistic and implementable trading strategy.

Chapter 6

Spread-Based Deep Reinforcement Learning Trading

6.1 Motivation

In this chapter, a novel high-frequency trading framework called Spread Aware Deep Reinforcement Learning (SADRL) is introduced. SADRL is the next iteration in the FDRL and PADRL succession of trading frameworks that uses the deep reinforcement learning techniques described in Section 3.2.4 and the directional change sampling algorithm to optimise trading decisions in high-frequency FX environments. The FDRL trading methodology demonstrated that a trading strategy that has been developed using DC sampling and deep reinforcement learning, can outperform passive strategies and strategies developed using fixed interval sampling. The next iteration of this strategy, PADRL, demonstrated that the FDRL method can be improved by adding positional variables to the feature set, allowing the agents to become fully autonomous and execute more profitable policies at a fixed transaction cost level of 0.035%.

The filter limitations of FDRL were addressed by PADRL, however the results of PADRL were still subject to a fixed transaction cost which is not realistic as transaction costs are dynamic and therefore change over time. The strategies im-

plemented by both FDRL and PADRL involve using the low volatility oscillations of the price at certain points in the data to rapidly enter and exit positions and in turn generate considerable profit. This strategy however is not effective when subject to the bid-ask spread of the market as the spread tends to widen when the price goes through these low volatility periods.

The SADRL framework addresses the fixed transaction cost limitation of FDRL and PADRL by using ideas from both and implementing a system that generates profit using the bid-ask spread. SADRL uses Trust Region Policy Optimisation (TRPO) to learn trading policies. TRPO is an alternative to the PPO algorithm used in FDRL and PADRL that is more computationally expensive but can often lead to better results. As well as an updated algorithm, SADRL also has access to a number of different indicators that have been derived from fixed interval sampling but applied to the DCC point of DC trends. A novel candlestick sampling approach has also been implemented to provide a more efficient view of the market as shown in Figure 6.1.

The candlestick construction methodology is inspired by traditional fixed-interval candlestick formation and builds on this by leveraging the event-driven nature of DC sampling. Each candlestick in this framework is constructed across two consecutive DC trends, capturing the market's directional movements in a manner that aligns with the underlying price dynamics rather than arbitrary time intervals. For a given candlestick, the opening price is defined as the DCC point of trend t marking the confirmation of the directional change. The closing price is set at the DCC point of trend $t+1$ representing the subsequent trend confirmation. The high and low values of each candlestick are determined by the the DCE points of the two trends that bound the candlestick. This construction ensures that the high corresponds to the maximum extreme points reached during the period, whilst the low reflects the minimum extreme points. As illustrated in Figure 6.1, green candlesticks indicate net upward price movements across the two-trend span, whilst red candlesticks represent net downward movements. This approach effectively

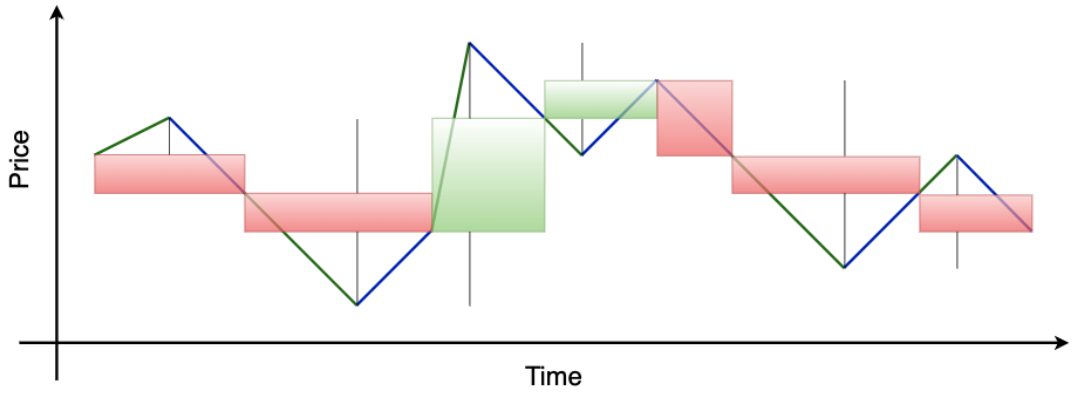

Figure 6.1: Candlestick Reframing. Blue lines represent the DC move and green lines represent the OS move of the DC trend. Green candlesticks represent an upward move across two DC trends and red candlesticks represent a downward move over two DC trends. Each candlestick opens at the DC confirmation point of the first move and closes at the DC point of the next move, the lows and highs of each candlestick are the start point of the previous trend and the start point of the current trend.

captures the magnitude and direction of price action between consecutive trend confirmations, providing the SADRL agents with a compact yet information rich representation of market behaviour that encodes both the volatility (through the high-low range) and the net directional bias (through the open-close relationship) of recent price action.

The implementation of these new features and a more advanced DRL algorithm allows the SADRL agents to learn policies to efficiently trade the high frequency FX market. SADRL therefore outperforms FDRL, PADRL and all previous benchmarks when subject to the more realistic bid-ask spread, as opposed to the fixed transactions costs. SADRL does this by implementing a trading strategy that differs from the alternate entries and exits observed in FDRL and PADRL trading behaviour.

The remainder of this chapter is structured as follows, Section 6.2 explains the methodology used to conduct the experiments, with Section 6.3 then discussing the experimental setup. The results are presented in Section 6.4 and interpreted in Section 6.5. Section 6.6 then summarises the findings of the study.

6.2 Methodology

The methodology begins with a discussion of the data preparation process for training the SADRL agents, as outlined in Section 6.2.1. The design of the action space, state space and reward function are then discussed in Sections 6.2.2, 6.2.3 and 6.2.3 respectively, to detail how each agent's environment is simulated.

6.2.1 Data Preparation

To transform the raw tick data into the required form before splitting into the appropriate train, validation and test sets, the data was passed through the pipeline of preparation steps shown in Figure 6.2. The first step of this pipeline was sampling the raw tick data using the DC sampling algorithm shown in Algorithm 1. This sampling process produced a set of DC trends over the sampling period which can the be used to create the candlesticks in the next step of the pipeline, using the methodology displayed in Figure 6.1. Once the candlesticks were created, indicators were generated based on the DCC prices of each trend. SADRL differs from FDRL and PADRL by removing all DC indicators and replacing them with the traditionally fixed interval based indicators defined in Table 6.1 which act on the DC sampled data, using the DCC point as the equivalent of the close of the period in the same way it would be applied to data sampled at a fixed interval. The motivation for switching to the more traditional fixed interval based indicators but applying them to the DC sampled data was based on preliminary testing which involved both DC and traditional indicators. When removing the DC-based indicators the models performed just as well as when both types of indicators were used, demonstrating no clear benefit of having DC indicators in the SADRL framework.

The non-stationary nature of time-series data can complicate the learning process for many DRL algorithms, so differencing was applied to all features in order to make the dataset stationary. The data was then split into rolling windows which were then split further into training, validation and test sets.

To verify the effectiveness of the differencing transformation, comprehensive stationarity testing was conducted on the differenced DCC price series across all currency pairs and DC thresholds using both the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. The ADF test evaluates the null hypothesis of non-stationarity (presence of a unit root), whilst the KPSS test evaluates the null hypothesis of stationarity, providing complementary perspectives on the time series properties.

The stationarity analysis covered 112 test cases (14 currency pairs $\times$ 8 DC thresholds) and demonstrated overwhelmingly positive results. The ADF test conclusively rejected the null hypothesis of non-stationarity for all 112 cases ($p < 0.001$), with test statistics ranging from -104.4 to -327.4, substantially more negative than the critical values at all significance levels. The KPSS test confirmed stationarity for 104 of 112 cases (92.9%), with only EUR/USD and USD/JPY showing conflicting results across all theta values. For these two pairs, the KPSS test suggested the presence of deterministic trends ($p < 0.05$), whilst the ADF test remained strongly significant. This pattern is characteristic of trend-stationary processes, where the series exhibits stationarity around a deterministic trend rather than pure difference-stationarity. Given that the ADF test is specifically designed for detecting unit roots and all test statistics were highly significant, the differencing transformation was deemed effective for rendering the data suitable for DRL training. The consistent stationarity across different DC thresholds (theta values ranging from 0.015% to 0.029%) further validated the robustness of the DC sampling approach, demonstrating that the fundamental statistical properties of the differenced series remain stable regardless of the sampling granularity. This stationarity verification provides theoretical justification for the subsequent application of technical indicators and ensures that the feature space presented to the DRL agents exhibits the stable statistical properties necessary for effective policy learning.

Once the data had been made stationary and split into the correct windows

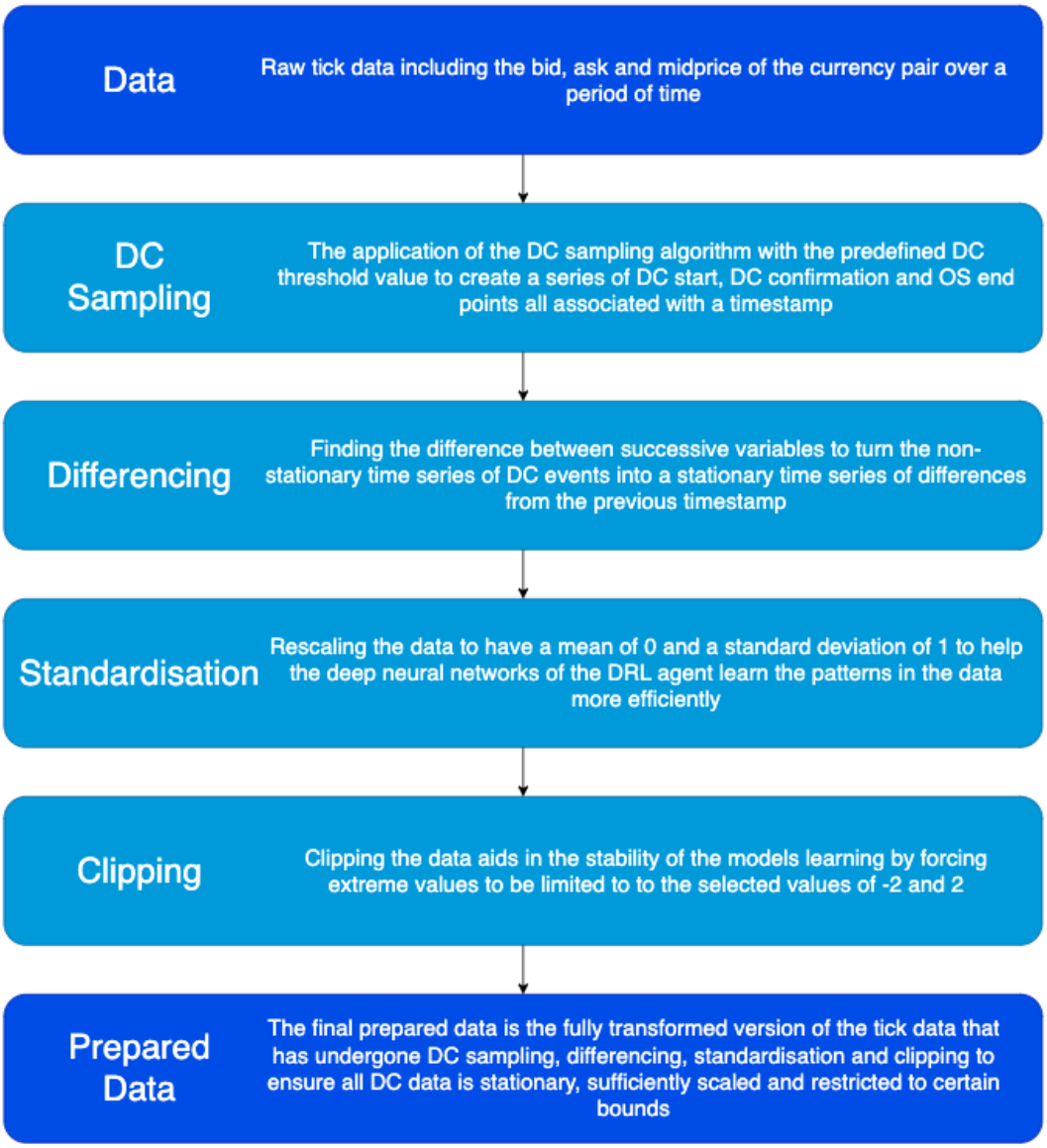

Figure 6.2: Diagram of data preparation pipeline

Table 6.1: Indicators

Indicator	Description	Equation	Period
MA	Moving Average of DCC price	$\frac{\sum_{i=1}^{n} P_i}{n}$	10, 20, 30, 40, 50
RSI	Relative Strength Index of DCC price	$100 - \frac{100}{1+RS}$	14, 21, 30
MACD	Moving Average Convergence Divergence of DCC price	MACD	(12, 26, 9), (9, 21, 7), (6, 13, 5)
BB	Bollinger Bands of DCC price	$MA \pm 2 \times SD$	14

and sets, the next task was to make time series data normally distributed using standardisation. This was done to improve model performance by ensuring stationarity in the data and therefore enhance learning efficiency and stability during training. The mean and standard deviation values required for standardisation were calculated using the training set and then applied to the validation and test sets of the corresponding window. The DRL algorithm implementation applied automatic clipping to the features so the data was scaled using normalisation to fit within these bounds. The clipping range was set to [-2, 2] to include the first two standard deviations of data and the scaling value was calculated from the training set and applied to the validation and test data in order to compress the data into the suitable bounds before training to further improve model training stability and meet the constraints of the training algorithm.

6.2.2 Action Space

The discrete action space used in the SADRL trading strategy was defined by actions of 0, 1 or 2, action 0 executes a buy trade (or exits a short position), action 1 executes a sell trade (or exits a long position) and action 2 denotes a hold (either do not make a trade and stay out the market or hold the current position in the market). This is different from the action spaces used in FDRL and PADRL as the SADRL setup did not suffer from the same inactivity problems and the ability to hold must be allowed to ensure the same rapid trading observed

in FDRL and PADRL that caused the results to not work using spread does not occur for SADRL. An important part of training the SADRL agents was to ensure the correct exploration and exploitation trade off, this was controlled by the sampling of actions during training. A custom action sampling policy was defined in order to provide a stochastic element to the action sampling, increasing exploration of the policy space. Using this policy, the epsilon value, defining the probability of selecting a random action, was tested (see Section 6.3 for further details).

6.2.3 State Space

The state space was the tensor representation of the environment at a given time step to the agent. There were two key components of the state space, the market state and the positional state. A combination of these two key components was used in the SADRL system to represent the whole state. The market state consisted of the indicators defined in Table 6.1 and candlestick information. The candlestick information consisted of two variables, the difference between the high and low of the candlestick and the difference between the open and close of the candlestick. These two variables were included to give the agent information on how extreme DC trends are. Given the scale of these values, the differences were multiplied by a factor of 1000 for all non-JPY pairs to adjust the scale of these values to match the scale of the other features and therefore help the neural networks under the hood of the SADRL agents learn. The spread value was also provided to the agent to help the agent enter at a good price and predict the cost of entering a trade based on the what the price may be when it was time to exit the trade.

The agent has no internal memory and therefore cannot track variables like its current position, market exposure or potential return. This information was provided to the agent by using 3 features: current trade direction (-1 for a short position, 0 for no position and 1 for a long position), current profit (multiplied by 10 to rescale the value to match that of other input variables) and potential return

(potential profit if the agent were to exit on the current time step multiplied by 1000 to scale correctly for non-JPY pairs). The concatenation of these positional variables with market variables to create a single vector at each time step captured the information required for the DRL agent to make informed trading decisions at each time step.

The state space $\mathcal{S}$ in SADRL maintains the dual-component structure introduced in PADRL but with modified market features adapted for spread-based trading. Each state $s_t \in \mathcal{S}$ is formally defined as:

$$s_t = [m_t, p_t] \in \mathbb{R}^{d_m+k} \tag{6.1}$$

where $m_t \in \mathbb{R}^{d_m}$ represents the market state vector at the current time step t, and $p_t \in \mathbb{R}^k$ represents the positional state vector with dimensionality $k = 4$. The market state vector m_t has dimensionality d_m comprising: 5 moving averages (MA) with periods 10, 20, 30, 40, 50; 3 relative strength index (RSI) values with periods 14, 21, 30; 9 MACD components (3 MACD configurations each yielding 3 values: MACD line, signal line, and histogram); 3 Bollinger Band values (upper, middle, lower) with period 14 and 2 candlestick features (high-low difference and open-close difference); and 1 spread value. This yields $d_m = 5 + 3 + 9 + 3 + 2 + 1 = 23$ market features per time step. The positional vector is defined as $p_t = [dir_t, prof_t, ret_t, spread_t]$, where $dir_t \in \{-1, 0, 1\}$ encodes the current trade direction, $prof_t$ represents the current profit (scaled by a factor of 10), ret_t represents the potential return if exiting at the current time step (scaled by 1000 for non-JPY pairs), and $spread_t$ captures the current spread value. The complete state representation therefore has dimensionality $d_m + k = 23 + 4 = 27$.

The SADRL state space differs from PADRL in several important ways. First, the market state m_t uses traditional technical indicators (moving averages, RSI, MACD, Bollinger Bands) rather than the DC-specific features employed in PADRL, reflecting a different feature engineering approach that prioritises conventional technical analysis. Second, the candlestick features provide additional granular-

ity about price movement patterns within each DC event, enabling the agent to distinguish between different types of consolidation periods and trend behaviours based on intrabar volatility. The scaling factors applied to positional features (10 for current profit, 1000 for potential return on non-JPY pairs) are critical for ensuring that positional information remains numerically comparable to market features during neural network training, preventing any single feature type from dominating the gradient updates. This adapted state representation provides the agent with the necessary information to develop sophisticated, spread-aware trading strategies suitable for realistic trading conditions.

Reward Function The reward function was defined as the profit of each trade. Extensive preliminary testing revealed that risk-adjusted reward functions did not consistently outperform a profit-based approach, which encouraged a more dynamic and opportunistic trading strategy. By focusing on profit maximisation, the agent was incentivised to execute short but lucrative trades, aligning with the ultimate objective of maximising returns over a given episode. While the FDRL and PADRL strategies demonstrated that fixed transaction costs could be exploited under this framework, incorporating the bid-ask spread introduced a natural constraint. This forced the agent to adopt more sophisticated trading behaviours, as the cost of rapid trade execution often exceeded potential profits. This approach ultimately ensured that the agent not only pursued profit but also developed an inherent sensitivity to bid-ask spread, leading to more efficient trading decisions.

6.3 Experimental Setup

6.3.1 Data

Following the approach in the previous chapter, raw tick data is sourced from TrueFX.com, covering USD and non-USD currency pairs within their respective available periods. The data is sampled using eight DC thresholds (0.015%–0.029%)

to balance DC event frequency and price movement significance. Each sampled series is segmented using a sliding window approach, generating 784 windows across all pairs and thresholds. Training, validation, and test sets are structured within a 24-week span (16 weeks for training, 4 for validation, and 4 for testing). This framework allows for evaluating the trading algorithm's performance across varying DC thresholds, ensuring robustness under different market conditions.

6.3.2 Hyperparameter Tuning

Hyperparameter tuning in this chapter follows the same structured approach as previously described for PADRL, using a grid search over a subset of validation sets. The key hyperparameters of `batch_size` and `n_epochs` were again optimised using this method, with the same values of 65,536 and 10 found to yield the best balance between performance and computational efficiency. As with PADRL a validation environment was integrated for SADRL using an evaluation callback within `Stable Baselines 3` (SB3). This callback, triggered every 50,000 time steps, assessed the model's performance on the validation set and retained the best performing model based on balance outcomes. If a validation simulation surpasses the highest validation balance attained in prior runs, the new model supersedes the previously identified best performing model. Following the completion of training, the remaining model, deemed the best performing one, is used for evaluation on the test set.

6.3.3 Performance Testing

As with FDRL and PADRL, trading strategies are evaluated using three performance metrics: total return which measures overall profitability, maximum drawdown which assesses risk by quantifying potential losses and Calmar ratio which balances return and risk. These three performance metrics are used for both comparing SADRL to benchmark algorithms and for DRL algorithm testing. Since FDRL and PADRL only used the PPO deep reinforcement learning algorithm,

SADRL took a much more rigorous approach and was trained with an extra three DRL algorithms to test if other DRL algorithms provided more fruitful results or exhibited different trading behaviour. The four DRL algorithms tested were DQN, A2C, PPO and TRPO. The SADRL system was trained using each algorithm and performance was measured using total return, maximum drawdown and Calmar ratio.

6.3.4 Benchmarks

The following benchmarks offer more advanced benchmarks that involve trading strategies such as Mean Reversion, MACD+RSI and Bollinger Bands. These strategies build on those defined in Section 5.3.4 by introducing clear trading rules that use multiple indicators. TADRL is also introduced to develop a benchmark that uses the SADRL framework but with fixed interval sampling to investigate the effects of adding DC-based sampling to the methodology.

Mean Reversion The Mean Reversion (MeanRev.) strategy is a technical analysis based strategy that uses data sampled at a fixed interval to build a strategy based on the assumption that the price will always revert back to the mean. In the case of price data the data is non-stationary so the mean is represented by a rolling mean.

MACDRSI The MACDRSI strategy is a technical analysis based strategy that combines the momentum indicators of Moving Average Convergence Divergence (MACD) and the Relative Strength Index (RSI). The strategy uses data sampled at a fixed interval to identify trading opportunities where both indicators align to signal potential trend reversals or continuations. MACD helps detect shifts in trend direction and momentum, while RSI provides information about overbought or oversold market conditions. By using these indicators together, the strategy aims to reduce false signals and improve entry and exit timing.

Bollinger Bands The Bollinger Bands (Boll.Bands) strategy is a technical analysis based strategy that uses Bollinger Bands to identify price volatility and potential reversal points. The strategy uses data sampled at a fixed interval and defines upper and lower bands around a moving average set two standard deviations away. When the price moves close to or beyond these bands, it may indicate that the asset is overbought or oversold, triggering potential buy or sell signals. The width of the bands also helps to gauge market volatility, allowing the strategy to adjust to changing market conditions.

TADRL The TADRL benchmark is the Technical Analysis Deep Reinforcement Learning strategy. This strategy is an exact match to the process used in SADRL with the same technical analysis indicators, however the indicators are calculated from data sampled at a fixed interval (a grid search on a restricted data set demonstrated that a fixed interval of a minute produced the best results). These results allow us to compare how the addition of Directional Changes sampling helps build a more profitable trading strategy.

6.4 Results

The following section examines how the performance of the four DRL algorithms, DQN, A2C, PPO, and TRPO, differs when trained using the SADRL training methodology. After identifying the best performing algorithm on the validation set, the best performing algorithm was used to retrained the model on the training and validation set and these results were compared to the previous strategies discussed in this thesis, namely FDRL and PADRL, using the test set. The SADRL results were then compared with two sets of strategy benchmarks. The first set in Section 6.4.2 includes the same passive and technical indicator based benchmarks in previous chapters to determine the effects of using DC sampled data when compared to other strategies that use fixed interval sampling without the DRL component. The second set of benchmarks in Section 6.4.3 looks at more ad-

vanced benchmarks that use technical analysis with DRL and established technical analysis strategies to help identify the improvements in performance provided by the DRL and DC sampling.

6.4.1 DRL Algorithm Performance

This section analyses the performance of DQN, A2C, PPO and TRPO in the SADRL trading system per performance metric. Every algorithm trading simulation is subject to the bid-ask spread during both training and testing on the validation set. The results for each algorithm are a direct substitution of the DRL trading algorithm in the SADRL framework, described by the methodology in Section 6.2.

Total Return TRPO and PPO are the strongest performing DRL algorithms for total return in the SADRL system, as shown by Table 6.2. Both PPO and TRPO consistently outperform DQN and A2C across all pairs and thresholds, with 73% (82/112) of all pair-threshold combinations showing PPO or TRPO as the best-performing DRL algorithm for training SADRL. The DQN algorithm produces the most return on only 20 pair-threshold combinations, and A2C achieves this on just 10 occasions. All remaining pair-threshold combinations are best achieved by either PPO or TRPO.

DQN produces losses of the entire balance on EUR/JPY at every DC threshold, with other pairs such as GBP/JPY and USD/JPY showing similar patterns of reliably negative returns. DQN performs well on AUD/USD, generating a positive return across all DC thresholds. Across all thresholds, DQN performs consistently well on currency pairs involving USD (apart from USD/JPY) and performs poorly on most pairs that involve JPY, suggesting that the behaviour of the US dollar and Japanese yen is the main determining factor in DQN's performance.

A2C shows much less fluctuation than DQN but still includes a few poorly performing pair-threshold combinations, most consistently with CHF/JPY, where

Table 6.2: Comparison of Total Return (%) by DC threshold for DQN, A2C, PPO and TRPO. The best value per currency pair at each DC threshold is denoted in boldface and the best value per threshold is underlined.

θ	Pair	DQN	A2C	PPO	TRPO
0.015%	AUD/JPY	-62.65	**5.92**	1.98	3.9
	AUD/USD	9.7	**40.73**	9.69	31.12
	CAD/JPY	-2.28	1.94	**9.85**	2.33
	CHF/JPY	-35.56	0.0	**1.44**	0.3
	EUR/CHF	1.41	**1.51**	1.48	0.71
	EUR/GBP	**6.72**	0.66	1.7	1.5
	EUR/JPY	-100.0	3.49	24.26	**40.42**
	EUR/USD	**36.91**	12.27	11.26	11.45
	GBP/JPY	-98.4	0.0	**5.98**	0.48
	GBP/USD	2.04	13.16	9.22	**13.67**
	NZD/USD	6.67	5.21	2.48	**10.19**
	USD/CAD	9.86	16.85	**24.02**	5.18
	USD/CHF	4.19	9.72	5.2	**11.49**
	USD/JPY	-34.68	1.62	11.39	**11.52**
0.017%	AUD/JPY	-69.1	-3.97	**7.87**	6.68
	AUD/USD	8.01	3.69	**24.54**	8.36
	CAD/JPY	-97.61	4.0	-6.74	**6.13**
	CHF/JPY	-44.88	-45.49	**3.38**	2.2
	EUR/CHF	1.17	**1.64**	0.16	1.11
	EUR/GBP	**4.07**	2.35	0.36	3.49
	EUR/JPY	-100.0	-48.9	-29.99	**141.95**
	EUR/USD	0.45	3.15	**13.27**	13.19
	GBP/JPY	-97.69	-39.47	**7.86**	4.51
	GBP/USD	8.72	0.51	**30.06**	10.91
	NZD/USD	7.62	2.34	-4.77	**9.84**
	USD/CAD	9.4	1.91	**24.62**	10.83
	USD/CHF	**7.25**	0.63	6.86	5.22
	USD/JPY	-51.5	-64.29	**12.14**	**12.14**
0.019%	AUD/JPY	-80.56	-12.02	-51.12	**4.49**
	AUD/USD	7.42	3.04	12.81	**23.21**
	CAD/JPY	-99.99	0.6	**5.25**	4.91
	CHF/JPY	-83.94	-84.14	**3.76**	2.95
	EUR/CHF	1.67	1.13	-9.76	**2.38**
	EUR/GBP	**10.04**	0.23	1.36	2.09
	EUR/JPY	-100.0	-2.24	-4.82	**3.47**
	EUR/USD	-1.89	3.83	**23.8**	11.28
	GBP/JPY	-100.0	0.0	4.94	**9.13**
	GBP/USD	**14.79**	2.04	8.61	4.35
	NZD/USD	6.94	-5.66	**18.3**	3.36
	USD/CAD	8.35	4.1	**19.49**	12.17
	USD/CHF	7.04	0.85	7.19	**7.59**
	USD/JPY	-88.9	-9.6	**13.91**	-0.27
0.021%	AUD/JPY	-90.04	1.96	2.16	**7.1**
	AUD/USD	16.77	32.35	**54.13**	18.14
	CAD/JPY	-98.0	0.15	**10.8**	9.96
	CHF/JPY	-98.91	-86.73	1.15	**5.19**
	EUR/CHF	1.13	-0.16	1.89	**2.53**
	EUR/GBP	**5.07**	1.31	0.75	1.38
	EUR/JPY	-96.53	-66.8	1.07	**8.09**
	EUR/USD	2.61	**76.81**	14.96	13.43
	GBP/JPY	-99.99	-2.56	**7.93**	3.73
	GBP/USD	9.56	3.31	**29.36**	13.73
	NZD/USD	10.14	2.99	-18.91	**13.93**
	USD/CAD	5.04	**17.22**	3.42	2.71
	USD/CHF	**6.36**	1.65	3.36	5.9
	USD/JPY	-99.22	-15.9	**8.33**	-3.91

θ	Pair	DQN	A2C	PPO	TRPO
0.023%	AUD/JPY	-100.0	0.64	4.23	**4.42**
	AUD/USD	2.49	1.55	**21.83**	19.95
	CAD/JPY	-100.0	-6.54	3.91	**7.11**
	CHF/JPY	-84.95	0.0	**5.31**	1.39
	EUR/CHF	**2.37**	1.58	-0.03	1.46
	EUR/GBP	**8.14**	1.25	0.5	4.16
	EUR/JPY	-100.0	-64.69	-6.3	**127.57**
	EUR/USD	10.7	9.07	**38.75**	27.07
	GBP/JPY	-95.97	4.87	3.57	**14.37**
	GBP/USD	17.54	3.93	**27.52**	20.84
	NZD/USD	**18.33**	2.47	4.83	9.33
	USD/CAD	**8.6**	2.04	6.1	6.26
	USD/CHF	4.75	1.01	**12.95**	4.26
	USD/JPY	-22.43	-15.11	**1.37**	-7.42
0.025%	AUD/JPY	-94.47	-3.98	7.99	**9.49**
	AUD/USD	13.27	**30.89**	21.95	28.33
	CAD/JPY	-99.06	1.02	-2.32	**5.28**
	CHF/JPY	-85.69	-21.65	4.89	**7.11**
	EUR/CHF	1.0	0.64	1.05	**20.56**
	EUR/GBP	**3.4**	1.32	0.25	1.32
	EUR/JPY	-99.36	-46.34	17.18	**121.69**
	EUR/USD	5.43	13.3	**29.4**	13.86
	GBP/JPY	-100.0	-13.13	5.7	**10.98**
	GBP/USD	13.98	2.8	**26.48**	4.01
	NZD/USD	**14.82**	4.39	11.05	7.23
	USD/CAD	**18.68**	11.95	14.82	12.79
	USD/CHF	**9.41**	1.28	6.16	1.11
	USD/JPY	-89.26	-65.35	7.64	**9.16**
0.027%	AUD/JPY	-61.44	-2.64	-0.83	**5.11**
	AUD/USD	17.21	13.84	20.76	**26.12**
	CAD/JPY	-97.01	5.39	4.6	**6.06**
	CHF/JPY	-36.45	-80.39	-0.6	**4.96**
	EUR/CHF	0.73	1.33	**6.83**	1.98
	EUR/GBP	**8.68**	0.61	2.07	2.77
	EUR/JPY	-100.0	-8.11	**202.9**	180.62
	EUR/USD	11.79	**26.71**	13.55	5.97
	GBP/JPY	-50.19	11.53	2.83	**17.23**
	GBP/USD	5.97	1.77	-61.82	**15.48**
	NZD/USD	**14.17**	2.26	8.26	9.05
	USD/CAD	9.22	10.85	**11.34**	7.04
	USD/CHF	**9.49**	3.34	4.41	7.27
	USD/JPY	-79.37	**4.26**	3.46	-55.2
0.029%	AUD/JPY	-0.49	-9.56	**7.87**	6.37
	AUD/USD	5.53	**50.9**	12.13	13.18
	CAD/JPY	-100.0	1.5	**9.05**	5.7
	CHF/JPY	-18.74	-51.13	1.47	**5.4**
	EUR/CHF	**2.98**	0.57	2.43	0.78
	EUR/GBP	1.01	-1.75	**3.08**	2.43
	EUR/JPY	-100.0	-52.81	1.48	**3.22**
	EUR/USD	10.36	34.81	**43.72**	17.95
	GBP/JPY	-99.66	-3.89	**8.12**	7.92
	GBP/USD	10.05	4.04	16.68	**17.01**
	NZD/USD	-5.78	3.0	-3.0	**7.03**
	USD/CAD	7.16	10.74	**11.79**	5.25
	USD/CHF	**11.83**	2.02	4.72	11.6
	USD/JPY	-98.94	-9.58	**9.8**	-34.17

the algorithm does not trade at all at 0.015%. For the remaining thresholds, it generates losses ranging from -86.73% to 1.58%. EUR/JPY also shows consistently poor results across most DC thresholds apart from 0.015%. A2C exhibits lower return volatility than DQN; however, this stability comes at the cost of generating

Table 6.3: Statistical test results for total return, according to the non-parametric Friedman test with the Conover post hoc test. Significantly improved performance over the TRPO strategy at the $\alpha = 0.05$ level is shown in boldface.

Friedman test p-value		**3.649e-15**
	Ave. Rank	p_{Con}
TRPO (c)	1.92	-
PPO	2.06	0.5379
A2C	3.00	**8.524e-10**
DQN	3.02	**2.437e-10**

fewer considerable gains compared to PPO or TRPO.

PPO is far more resilient to the data than DQN and A2C, as it produces consistently positive returns across all 14 currency pairs for nearly every DC threshold. A few anomalous large losses do occur, such as EUR/JPY at 0.017%, AUD/JPY at 0.019%, NZD/USD at 0.021%, and GBP/USD at 0.027%, with values of -29.99%, -51.12%, -18.91%, and -61.82% respectively. Aside from these anomalies, only 10 other losses are reported, totalling 14 loss-making pair-threshold combinations out of 112 (88% generate positive returns). PPO achieves some very high returns, the highest being 202.9% for EUR/JPY at a DC threshold of 0.027%. The overarching theme of PPO is that it delivers very favourable returns, and in some cases, extreme returns. The trade-off, however, is the presence of occasional anomalies with significant losses.

TRPO performs well across most pairs and thresholds, with only 5 of the 112 pair-threshold combinations resulting in negative returns (96% of combinations generate positive returns). As with PPO, a few anomalies occur, such as USD/JPY at 0.027% and 0.029%, which produce losses of -55.2% and -34.17% respectively. Despite these, TRPO produces the most favourable results overall, as shown in Table 6.3, with an average rank of 1.92 compared to 2.06, 3.00, and 3.02 for PPO, A2C, and DQN respectively.

Table 6.3 presents the results of a non-parametric Friedman test and post hoc Conover test, where the null hypothesis, stating that there is no statistically significant difference in performance across the trading strategies, is tested. The Friedman test produces a p-value of 3.649e-15, indicating that the null hypothesis

can be rejected. Conover's post hoc test shows that TRPO significantly outperforms A2C and DQN, with p-values of 8.524e-10 and 2.437e-10 respectively, but does not significantly outperform PPO at the 5% significance level (p-value = 0.5379), suggesting that there is insufficient evidence to conclude a statistically significant performance difference between TRPO and PPO.

Maximum Drawdown In Table 6.4, the maximum drawdown results are shown for each DRL algorithm. From this table, it can be observed that A2C, TRPO, and PPO often produce similar results, with DQN clearly demonstrating the worst performance of the four DRL algorithms. All pair-threshold combinations that have a total return of 100% in Table 6.2 therefore show a drawdown of 100% in Table 6.4, as demonstrated by EUR/JPY at DC thresholds 0.015%, 0.017%, 0.019%, 0.023%, and 0.029%, GBP/JPY at 0.019% and 0.025%, and CAD/JPY at 0.023% for DQN. DQN produces the lowest maximum drawdown on just 7/112 (6%) pair-threshold combinations, highlighting that it is the worst DRL algorithm for managing drawdown.

A2C shows relatively low maximum drawdown results across all thresholds and pairs, with 46/112 (41%) combinations producing a drawdown of less than 1%. However, some pairs still produce significant drawdowns, with EUR/JPY showing values above 40% on all thresholds apart from 0.015%. A number of pair-threshold combinations show 0% drawdown but no associated return, indicating that no trades were entered,this is investigated in more detail during the Calmar ratio analysis.

PPO produces several very low maximum drawdown values across thresholds, with a few anomalies where drawdown exceeds 20%. These occur for USD/JPY at 0.015%, 0.023%, 0.025%, 0.027%, and 0.029%, EUR/JPY at 0.017%, 0.023%, and 0.027%, AUD/JPY at 0.019%, and NZD/USD at 0.021%. PPO generates the best drawdown result for 22/112 (20%) pair-threshold combinations. Table 6.5 shows the average rank for each DRL algorithm, with very little difference between A2C and PPO, which have average ranks of 2.09 and 2.44 respectively. DQN remains

Table 6.4: Comparison of Maximum Drawdown (%) by DC threshold for DQN, A2C, PPO and TRPO. Best value per currency pair at each DC threshold is denoted in boldface and the best value per threshold is underlined.

θ	Pair	DQN	A2C	PPO	TRPO
0.015%	AUD/JPY	61.93	1.71	0.83	**0.25**
	AUD/USD	1.27	**0.71**	1.75	8.47
	CAD/JPY	11.54	**0.0**	0.27	1.51
	CHF/JPY	36.09	**0.0**	2.79	**0.0**
	EUR/CHF	0.76	**0.28**	1.13	1.0
	EUR/GBP	3.0	**1.0**	4.05	1.61
	EUR/JPY	100.0	**0.0**	8.83	1.25
	EUR/USD	6.1	1.72	**0.69**	3.66
	GBP/JPY	98.43	**0.0**	0.31	0.02
	GBP/USD	**0.31**	0.45	1.41	1.55
	NZD/USD	6.47	**4.31**	5.13	5.04
	USD/CAD	3.67	**0.22**	11.73	5.68
	USD/CHF	3.42	18.24	**0.67**	2.15
	USD/JPY	35.7	**0.42**	23.34	8.26
0.017%	AUD/JPY	69.16	6.2	2.29	**0.66**
	AUD/USD	1.57	1.93	2.66	**0.52**
	CAD/JPY	97.63	**0.06**	11.42	0.54
	CHF/JPY	47.41	42.91	1.43	**1.36**
	EUR/CHF	0.78	**0.66**	2.19	0.75
	EUR/GBP	2.15	**0.74**	2.46	4.15
	EUR/JPY	100.0	51.63	42.29	**2.88**
	EUR/USD	4.6	4.35	**0.45**	2.03
	GBP/JPY	97.69	40.16	**2.36**	2.59
	GBP/USD	2.68	0.51	**0.45**	1.8
	NZD/USD	7.95	**0.49**	16.3	3.63
	USD/CAD	3.26	**0.34**	14.13	0.41
	USD/CHF	3.6	2.12	0.98	**0.95**
	USD/JPY	51.11	65.2	**1.63**	16.1
0.019%	AUD/JPY	80.82	15.68	52.44	**2.26**
	AUD/USD	2.47	**0.33**	8.08	6.86
	CAD/JPY	99.99	**0.0**	1.19	1.77
	CHF/JPY	84.04	84.14	1.17	**0.71**
	EUR/CHF	1.62	1.33	12.36	**1.16**
	EUR/GBP	**1.97**	2.04	3.12	5.33
	EUR/JPY	100.0	44.44	14.11	**1.13**
	EUR/USD	4.43	3.41	2.64	**1.55**
	GBP/JPY	100.0	**0.0**	1.41	1.21
	GBP/USD	3.07	**0.23**	1.58	0.78
	NZD/USD	11.88	7.55	**5.06**	7.96
	USD/CAD	2.64	**1.09**	14.49	2.81
	USD/CHF	3.66	**1.06**	8.77	1.62
	USD/JPY	89.22	11.06	**1.63**	19.65
0.021%	AUD/JPY	90.06	3.46	0.66	**0.37**
	AUD/USD	0.58	**0.33**	0.53	3.95
	CAD/JPY	98.03	0.88	0.78	**0.76**
	CHF/JPY	98.91	87.06	3.34	**0.65**
	EUR/CHF	6.32	1.1	**0.73**	1.42
	EUR/GBP	**0.64**	1.95	3.28	17.65
	EUR/JPY	96.51	66.83	4.16	**3.25**
	EUR/USD	3.88	11.48	**1.51**	3.54
	GBP/JPY	99.99	5.88	**1.23**	2.22
	GBP/USD	1.42	0.44	**0.3**	1.2
	NZD/USD	**1.39**	2.5	29.29	7.11
	USD/CAD	**1.21**	7.25	4.98	3.93
	USD/CHF	2.39	**0.33**	0.97	3.19
	USD/JPY	99.22	20.18	**0.72**	11.54

θ	Pair	DQN	A2C	PPO	TRPO
0.023%	AUD/JPY	100.0	0.56	0.77	**0.43**
	AUD/USD	4.09	**0.31**	2.91	2.58
	CAD/JPY	100.0	10.25	0.47	**0.35**
	CHF/JPY	85.06	**0.0**	0.98	1.39
	EUR/CHF	0.69	**0.64**	2.47	0.73
	EUR/GBP	2.5	**0.77**	7.48	1.66
	EUR/JPY	100.0	65.88	33.81	**5.51**
	EUR/USD	**0.4**	6.58	6.66	3.69
	GBP/JPY	96.01	2.16	4.14	**0.75**
	GBP/USD	3.48	0.8	**0.73**	4.58
	NZD/USD	24.79	**1.49**	11.16	10.72
	USD/CAD	6.29	**0.82**	2.52	7.45
	USD/CHF	3.08	**1.06**	18.6	3.95
	USD/JPY	23.53	**17.16**	21.23	24.95
0.025%	AUD/JPY	94.57	4.92	**0.75**	1.59
	AUD/USD	**0.4**	11.67	9.5	7.42
	CAD/JPY	99.1	2.34	6.54	**2.22**
	CHF/JPY	86.01	22.92	**0.5**	1.5
	EUR/CHF	2.74	**0.59**	1.16	2.96
	EUR/GBP	2.46	**1.48**	2.72	2.32
	EUR/JPY	99.36	70.19	17.18	**2.89**
	EUR/USD	**3.98**	7.53	7.99	6.3
	GBP/JPY	100.0	20.07	1.86	**1.51**
	GBP/USD	1.32	**0.81**	2.51	1.2
	NZD/USD	2.35	**2.1**	7.74	5.16
	USD/CAD	5.25	**0.79**	1.38	8.35
	USD/CHF	**3.05**	3.7	11.2	3.51
	USD/JPY	89.41	66.5	**41.83**	47.9
0.027%	AUD/JPY	62.05	4.35	8.11	**0.39**
	AUD/USD	9.99	0.79	2.18	**0.48**
	CAD/JPY	97.08	1.78	**0.51**	1.13
	CHF/JPY	37.65	80.39	4.85	**0.68**
	EUR/CHF	2.75	**0.38**	1.75	9.45
	EUR/GBP	**0.55**	0.64	2.26	2.57
	EUR/JPY	100.0	60.68	34.29	**1.98**
	EUR/USD	1.33	**0.46**	1.29	3.02
	GBP/JPY	51.42	1.02	**0.45**	1.04
	GBP/USD	2.84	**0.72**	67.52	1.01
	NZD/USD	10.23	**0.42**	4.81	85.71
	USD/CAD	3.53	8.39	**0.98**	5.93
	USD/CHF	4.83	**0.47**	2.89	14.75
	USD/JPY	79.61	**3.84**	34.39	58.56
0.029%	AUD/JPY	5.67	10.19	**0.25**	1.41
	AUD/USD	5.82	29.06	3.76	**2.5**
	CAD/JPY	100.0	2.48	0.58	**0.3**
	CHF/JPY	23.34	51.08	4.65	**1.59**
	EUR/CHF	14.16	0.82	**0.76**	0.99
	EUR/GBP	2.7	2.82	2.49	**0.96**
	EUR/JPY	100.0	68.19	**4.39**	53.16
	EUR/USD	4.16	**0.66**	2.86	4.76
	GBP/JPY	99.66	13.72	1.81	**1.52**
	GBP/USD	0.78	**0.5**	2.57	3.22
	NZD/USD	21.59	**2.41**	19.72	2.56
	USD/CAD	4.29	**0.9**	7.5	11.48
	USD/CHF	**0.44**	0.61	4.9	1.63
	USD/JPY	98.96	37.97	**33.66**	37.5

the worst-performing algorithm with an average rank of 3.20. This similarity in ranking suggests that there is no statistically significant difference between A2C and PPO.

TRPO ranks as the second-best performing algorithm for maximum drawdown

Table 6.5: Statistical test results for maximum drawdown, according to the non-parametric Friedman test with the Conover post hoc test. Significantly improved performance over the TRPO strategy at the $\alpha = 0.05$ level is shown in boldface.

Friedman test p-value		**2.758e-10**
	Ave. Rank	p_{Con}
A2C	2.09	0.5851
TRPO (c)	2.27	-
PPO	2.44	0.5851
DQN	3.20	**5.8057e-09**

with an average rank of 2.27. Although it produces the lowest drawdown in just 28% of cases, TRPO maintains low drawdowns across nearly every pair and threshold, with only 11 cases exceeding 10%. Table 6.5 again presents the results of a non-parametric Friedman test and post hoc Conover test, where the null hypothesis, stating that there is no statistically significant difference in performance across the strategies, is tested. The Friedman test produces a p-value of 2.758e-10, indicating statistical significance. When Conover's post hoc test is applied, it shows that TRPO significantly outperforms DQN at the 5% significance level. However, both A2C and PPO return p-values of 0.5851, suggesting there is not sufficient evidence to conclude a statistically significant outperformance of TRPO over either A2C or PPO.

Calmar Ratio Table 6.6 shows the Calmar ratio of the four different DRL algorithms tested in the SADRL trading system. Calmar ratio measures the risk adjusted return of each trading simulation, so the algorithms that may produce larger profits but incur a large risk in order to do so will be outperformed by algorithms that can provide slightly less returns but for considerably less risk. From Table 6.6 it is clear that DQN struggles to compete with the other DRL algorithms as it produces the best Calmar ratio in only 16/112 (14%) pair-threshold combinations. DQN also produces a number of negative Calmar ratios as would be expected from the negative returns seen in 6.2. The highest Calmar ratio produced by DQN is 32.82, there appears to be a theme of DQN producing either large Calmar ratios or barely being able to produce a positive result, supporting

Table 6.6: Comparison of Calmar Ratio by DC threshold for DQN, A2C, PPO and TRPO. Best value per currency pair at each DC threshold is denoted in boldface and the best value per threshold is underlined.

θ	Pair	DQN	A2C	PPO	TRPO
0.015%	AUD/JPY	-1.01	3.47	2.4	**15.76**
	AUD/USD	7.63	**57.71**	5.54	3.67
	CAD/JPY	-0.2	**inf**	36.71	1.55
	CHF/JPY	-0.99	**inf**	0.51	**inf**
	EUR/CHF	1.86	**5.29**	1.3	0.71
	EUR/GBP	**2.24**	0.67	0.42	0.93
	EUR/JPY	-1.0	**inf**	2.75	32.29
	EUR/USD	6.05	7.12	**16.32**	3.13
	GBP/JPY	-1.0	**inf**	19.3	25.52
	GBP/USD	6.5	**29.15**	6.55	8.8
	NZD/USD	1.03	1.21	0.48	**2.02**
	USD/CAD	2.69	**75.1**	2.05	0.91
	USD/CHF	1.22	0.53	**7.79**	5.34
	USD/JPY	-0.97	**3.85**	0.49	1.39
0.017%	AUD/JPY	-1.0	-0.64	3.44	**10.13**
	AUD/USD	5.12	1.92	9.24	**16.18**
	CAD/JPY	-1.0	**65.53**	-0.59	11.34
	CHF/JPY	-0.95	-1.06	**2.35**	1.62
	EUR/CHF	1.49	**2.48**	0.07	1.49
	EUR/GBP	1.89	**3.18**	0.15	0.84
	EUR/JPY	-1.0	-0.95	-0.71	**49.3**
	EUR/USD	0.1	0.72	**29.3**	6.51
	GBP/JPY	-1.0	-0.98	**3.33**	1.74
	GBP/USD	3.26	1.0	**67.14**	6.06
	NZD/USD	0.96	**4.73**	-0.29	2.71
	USD/CAD	2.89	5.63	1.74	**26.48**
	USD/CHF	2.01	0.3	**7.03**	5.53
	USD/JPY	-1.01	-0.99	**7.46**	0.75
0.019%	AUD/JPY	-1.0	-0.77	-0.97	**1.98**
	AUD/USD	3.0	**9.2**	1.58	3.39
	CAD/JPY	-1.0	**inf**	4.4	2.77
	CHF/JPY	-1.0	-1.0	3.23	**4.16**
	EUR/CHF	1.04	0.85	-0.79	**2.06**
	EUR/GBP	**5.1**	0.11	0.44	0.39
	EUR/JPY	-1.0	-0.05	-0.34	**3.08**
	EUR/USD	-0.43	1.12	**9.0**	7.3
	GBP/JPY	-1.0	**inf**	3.51	7.53
	GBP/USD	4.82	**8.89**	5.45	5.56
	NZD/USD	0.58	-0.75	**3.62**	0.42
	USD/CAD	3.16	3.76	1.35	**4.33**
	USD/CHF	1.92	0.81	0.82	**4.68**
	USD/JPY	-1.0	-0.87	**8.52**	-0.01
0.021%	AUD/JPY	-1.0	0.57	3.28	**19.19**
	AUD/USD	28.86	97.72	**102.7**	4.6
	CAD/JPY	-1.0	0.17	**13.83**	13.06
	CHF/JPY	-1.0	-1.0	0.34	**8.04**
	EUR/CHF	0.18	-0.14	**2.59**	1.78
	EUR/GBP	**7.89**	0.67	0.23	0.08
	EUR/JPY	-1.0	-1.0	0.26	**2.49**
	EUR/USD	0.67	6.69	**9.93**	3.8
	GBP/JPY	-1.0	-0.43	**6.44**	1.68
	GBP/USD	6.75	7.61	**99.35**	11.4
	NZD/USD	**7.29**	1.19	-0.65	1.96
	USD/CAD	**4.15**	2.38	0.69	0.69
	USD/CHF	2.66	**5.02**	3.46	1.85
	USD/JPY	-1.0	-0.79	**11.57**	-0.34

θ	Pair	DQN	A2C	PPO	TRPO
0.023%	AUD/JPY	-1.0	1.15	5.5	**10.4**
	AUD/USD	0.61	4.93	7.49	**7.72**
	CAD/JPY	-1.0	-0.64	8.4	**20.21**
	CHF/JPY	-1.0	**inf**	5.43	1.0
	EUR/CHF	**3.41**	2.49	-0.01	2.01
	EUR/GBP	**3.25**	1.63	0.07	2.51
	EUR/JPY	-1.0	-0.98	-0.19	**23.17**
	EUR/USD	**26.96**	1.38	5.82	7.34
	GBP/JPY	-1.0	2.25	0.86	**19.21**
	GBP/USD	5.04	4.89	**37.54**	4.55
	NZD/USD	0.74	**1.66**	0.43	0.87
	USD/CAD	1.37	**2.5**	2.42	0.84
	USD/CHF	**1.54**	0.95	0.7	1.08
	USD/JPY	-0.95	-0.88	**0.06**	-0.3
0.025%	AUD/JPY	-1.0	-0.81	**10.6**	5.98
	AUD/USD	**32.82**	2.65	2.31	3.82
	CAD/JPY	-1.0	0.43	-0.36	**2.38**
	CHF/JPY	-1.0	-0.94	**9.84**	4.73
	EUR/CHF	0.36	1.08	0.91	**6.94**
	EUR/GBP	**1.38**	0.89	0.09	0.57
	EUR/JPY	-1.0	-0.66	1.0	**42.09**
	EUR/USD	1.37	1.77	**3.68**	2.2
	GBP/JPY	-1.0	-0.65	3.06	**7.28**
	GBP/USD	**10.6**	3.45	10.54	3.33
	NZD/USD	**6.31**	2.09	1.43	1.4
	USD/CAD	3.56	**15.09**	10.75	1.53
	USD/CHF	**3.08**	0.35	0.55	0.32
	USD/JPY	-1.0	-0.98	0.18	**0.19**
0.027%	AUD/JPY	-0.99	-0.61	-0.1	**13.2**
	AUD/USD	1.72	17.55	9.52	**54.98**
	CAD/JPY	-1.0	3.02	**9.01**	5.38
	CHF/JPY	-0.97	-1.0	-0.12	**7.28**
	EUR/CHF	0.26	3.46	**3.9**	0.21
	EUR/GBP	**15.83**	0.94	0.91	1.08
	EUR/JPY	-1.0	-0.13	5.92	**91.03**
	EUR/USD	8.89	**57.72**	10.48	1.98
	GBP/JPY	-0.98	11.35	6.27	**16.53**
	GBP/USD	2.1	2.46	-0.92	**15.34**
	NZD/USD	1.39	**5.37**	1.71	0.11
	USD/CAD	2.62	1.29	**11.53**	1.19
	USD/CHF	1.97	**7.1**	1.52	0.49
	USD/JPY	-1.0	**1.11**	0.1	-0.94
0.029%	AUD/JPY	-0.09	-0.94	**31.32**	4.5
	AUD/USD	0.95	1.75	3.23	**5.27**
	CAD/JPY	-1.0	0.61	15.74	**18.82**
	CHF/JPY	-0.8	-1.0	0.31	**3.4**
	EUR/CHF	0.21	0.69	**3.22**	0.79
	EUR/GBP	0.37	-0.62	1.24	**2.54**
	EUR/JPY	-1.0	-0.77	**0.34**	0.06
	EUR/USD	2.49	**53.04**	15.29	3.77
	GBP/JPY	-1.0	-0.28	4.49	**5.19**
	GBP/USD	**12.84**	8.07	6.5	5.29
	NZD/USD	-0.27	1.24	-0.15	**2.75**
	USD/CAD	1.67	**11.97**	1.57	0.46
	USD/CHF	**27.01**	3.3	0.96	7.13
	USD/JPY	-1.0	-0.25	**0.29**	-0.91

Table 6.7: Statistical test results for Calmar ratio, according to the non-parametric Friedman test with the Conover post hoc test. Significantly improved performance over the TRPO strategy at the $\alpha = 0.05$ level is shown in boldface.

Friedman test p-value		**7.8179e-08**
	Ave. Rank	p_{Con}
TRPO (c)	2.14	-
PPO	2.33	0.1147
A2C	2.42	**1.595e-03**
DQN	3.11	**3.578e-11**

the suggestion that DQN can be effective but struggles with stability across different datasets and can fall into bad policy spaces and therefore produce the -1.0 values observed across a number of pair-threshold combinations.

A2C appears much more stable than DQN and produces the best result in 28/112 (25%) pair-threshold combinations. A2C also produces a number of results with no drawdown such as CAD/JPY, CHF/JPY, EUR/JPY and GBP/JPY at 0.015% and CAD/JPY and GBP/JPY at 0.019%, resulting in an infinite Calmar ratio. The comparatively low returns observed in Table 6.2 are the main driving force behind the relatively low Calmar ratios observed for A2C as the maximum drawdown results for A2C have an average rank of 2.09, compared to the average rank of 3.00 for total returns. As shown in Table 6.7, the average rank of A2C for Calmar ratio is 2.42, placing it as the second worst performing DRL algorithm on Calmar ratio.

PPO performs well on Calmar ratio, producing the best result on 30/112 (27%) of all pair-threshold combinations. PPO also generates some large Calmar ratios with a high of 102.7 for AUD/USD at 0.021%. The success shown by the PPO algorithm places it at an average rank of 2.33 across all four algorithms, 0.78 better than DQN and 0.09 better than A2C. The results for PPO are comparable to those for TRPO which produces the best performing Calmar ratio on 38/112 (34%) of the pair-threshold combinations, meaning PPO and TRPO produce the most favourable results in 61% of cases, often ranking second to one another, hence a difference in average rank of 0.19.

Table 6.7 shows the results of the non-parametric Friedman test and Conover's

post hoc test on the Calmar ratio results where the null hypothesis is tested, stating that the results across DQN, A2C, PPO and TRPO are not significant. The p-value of 7.8179e-08 from the Friedman test shows that the results are significant meaning this null hypothesis for the Friedman test can be rejected. Conover's post hoc test looks deeper into the pairwise significance results and exposes that fact that TRPO outperforms DQN and A2C with p-values of 3.578e-11 and 1.595e-03 respectively, therefore showing that these results are significant at a significance threshold of 5%. With regards to TRPO and PPO, the post hoc Conover test gives a value of 0.1147, showing that TRPO significantly outperforms PPO at a significance level of 15% but not 5%.

Summary Tables 6.2, 6.4 and 6.6 show that TRPO and PPO outperform DQN and A2C on the majority of occasions but this is mostly due to the total return performance. The maximum drawdown results in Table 6.4 paint a slightly different picture to the total return results in Table 6.2 as A2C is much more competitive, often outperforming both PPO and TRPO. The Calmar ratio results observed in Table 6.6 demonstrate the influence of the considerably higher observed returns for PPO and TRPO in comparison to the relatively similar maximum drawdown results to A2C, explaining why PPO and TRPO offer more return for similar levels of risk, as represented by the Calmar ratio results.

Tables 6.3, 6.5 and 6.7 demonstrate the average ranking and significance of these results, showing clearly that TRPO is the best performing algorithm across all metrics. From Conover's post hoc test it can be observed that TRPO significantly outperforms A2C and DQN on both total return and Calmar ratio with less significant results for maximum drawdown. These results suggest that TRPO is the best DRL algorithm for the SADRL system for risk adjusted return.

6.4.2 Technical Indicator Systems

The following analyses the performance of the SADRL, PADRL and FDRL trading systems per performance metric. The SADRL reports results for the TRPO DRL algorithm as it has been observed to be the most effective algorithm for trading in the SADRL system. SADRL, PADRL and FDRL are all subject to the bid-ask spread during both training and testing, PADRL and FDRL have therefore been retrained with the spread instead of applying the same models from the previous sections to make the results a fair comparison.

Total Return Table 6.8 shows the total return results for FDRL, PADRL and SADRL when trained using the bid-ask spread for all pairs and DC thresholds. The most apparent features of these results given the context of the previous chapters on FDRL and PADRL is their poor performance when subjected to a dynamic transaction cost. FDRL produces the best performing strategy in only 7% of cases, when FDRL does produce positive results these results tend to outperform both SADRL and PADRL. In the cases when FDRL produces returns above 50% as in EUR/USD at DC thresholds of 0.019%, 0.023% and 0.025% and AUD/USD at a DC threshold of 0.027%, it can be deduced that the data must lend itself to the FDRL style of trading, the short entry and exit positions that are much less frequent when subjected to the bid-ask spread. When FDRL does not produce positive results it tends to perform particularly badly, in some cases making losses of greater than 99% of the starting balance.

PADRL appears slightly more stable than FDRL with fewer significant losses but still a large number of losses across all 112 pair-threshold combinations. PADRL outperforms SADRL and FDRL in 9% of cases with 4 of these cases also generating the most return of all datasets in that DC threshold. Figure 6.3 confirms that FDRL and PADRL tend to perform well on the same pair-threshold combinations with a correlation coefficient of 0.72. This is perhaps unsurprising given they have both been optimised under an environment that uses fixed transaction costs and

a related methodology. The addition of positional variables is still evident here by the improved performance of PADRL over FDRL.

Table 6.8: Comparison of Total Return (%) by DC threshold for SADRL, PADRL and FDRL. Best value per currency pair at each DC threshold is denoted in boldface and the best value per DC threshold is underlined.

θ	Pair	SADRL	PADRL	FDRL	θ	Pair	SADRL	PADRL	FDRL
0.015%	AUD/JPY	**5.14**	-18.44	-25.41	0.023%	AUD/JPY	-17.04	**-16.64**	-51.58
	AUD/USD	5.36	**20.21**	-49.36		AUD/USD	17.77	**36.4**	26.3
	CAD/JPY	**3.11**	-17.27	-11.68		CAD/JPY	**7.99**	-19.16	-62.25
	CHF/JPY	**0.06**	-13.49	-59.81		CHF/JPY	**4.74**	-14.7	-76.04
	EUR/CHF	**1.99**	-21.11	-99.06		EUR/CHF	**0.87**	-23.06	-100.0
	EUR/GBP	**3.26**	-28.77	-99.82		EUR/GBP	**3.05**	-24.78	-100.0
	EUR/JPY	**-0.02**	-32.55	-99.99		EUR/JPY	**6.2**	-12.62	-100.0
	EUR/USD	9.2	-18.62	**31.56**		EUR/USD	17.41	42.44	**55.3**
	GBP/JPY	**0.72**	-11.85	-61.26		GBP/JPY	-24.43	**-13.24**	-88.51
	GBP/USD	**17.24**	-28.42	-92.22		GBP/USD	**12.59**	-24.03	-86.82
	NZD/USD	**12.28**	-42.69	-70.87		NZD/USD	**7.58**	-43.49	-75.02
	USD/CAD	**5.88**	-2.61	-15.67		USD/CAD	**4.55**	-22.95	-13.79
	USD/CHF	**6.24**	-14.75	-99.13		USD/CHF	**13.33**	-12.42	-96.78
	USD/JPY	**9.72**	-19.14	-3.57		USD/JPY	**-0.01**	-38.55	-32.2
0.017%	AUD/JPY	**5.99**	-13.53	-32.04	0.025%	AUD/JPY	**9.49**	-11.5	-86.92
	AUD/USD	14.02	**55.58**	-35.21		AUD/USD	10.71	**36.87**	5.36
	CAD/JPY	**8.28**	-21.64	-26.15		CAD/JPY	**3.79**	-19.09	-40.76
	CHF/JPY	**-1.73**	-14.84	-30.28		CHF/JPY	**5.89**	-11.22	-51.9
	EUR/CHF	**3.31**	-27.06	-99.97		EUR/CHF	**1.89**	-10.84	-100.0
	EUR/GBP	**3.16**	-16.97	-99.97		EUR/GBP	**0.79**	-9.48	-99.97
	EUR/JPY	**0.91**	-51.12	-100.0		EUR/JPY	**2.05**	-10.86	-100.0
	EUR/USD	8.53	**15.53**	-8.78		EUR/USD	10.97	24.08	**76.6**
	GBP/JPY	**6.38**	-12.79	-34.77		GBP/JPY	**11.73**	-13.44	-78.37
	GBP/USD	**25.1**	-43.25	-99.05		GBP/USD	**11.38**	-62.22	-86.17
	NZD/USD	**9.4**	-44.95	-85.65		NZD/USD	**13.43**	-36.95	-87.07
	USD/CAD	**6.9**	-15.28	-11.17		USD/CAD	**5.46**	-8.21	-12.37
	USD/CHF	**3.4**	-9.93	-88.97		USD/CHF	**5.02**	-14.54	-96.72
	USD/JPY	**4.33**	-26.41	-52.95		USD/JPY	**12.62**	-5.73	-47.49
0.019%	AUD/JPY	**7.63**	-19.68	-37.64	0.027%	AUD/JPY	**6.08**	-15.82	-66.81
	AUD/USD	6.83	**61.63**	37.23		AUD/USD	13.17	-55.28	**66.17**
	CAD/JPY	**6.66**	-16.57	-24.71		CAD/JPY	**7.52**	-13.51	-80.59
	CHF/JPY	**8.04**	-11.51	-60.43		CHF/JPY	**9.55**	-12.15	-72.61
	EUR/CHF	**3.35**	-19.05	-100.0		EUR/CHF	**2.22**	-19.41	-100.0
	EUR/GBP	**0.31**	-21.47	-99.7		EUR/GBP	**3.13**	-12.36	-99.98
	EUR/JPY	**3.99**	-17.73	-100.0		EUR/JPY	**12.42**	-14.93	-100.0
	EUR/USD	4.8	22.72	**68.67**		EUR/USD	13.68	12.73	**38.93**
	GBP/JPY	**8.59**	-10.75	-71.69		GBP/JPY	**14.01**	-77.62	-97.49
	GBP/USD	**6.13**	-47.77	-91.05		GBP/USD	**7.19**	-24.73	-94.99
	NZD/USD	**3.88**	-10.27	-86.34		NZD/USD	**5.24**	-39.69	-96.16
	USD/CAD	**4.05**	-27.25	-1.29		USD/CAD	5.38	7.93	**16.2**
	USD/CHF	**10.04**	-18.1	-97.13		USD/CHF	**2.18**	-22.79	-97.87
	USD/JPY	**-3.32**	-17.84	-66.97		USD/JPY	-37.52	**-12.0**	-68.74
0.021%	AUD/JPY	**8.6**	-11.29	-44.57	0.029%	AUD/JPY	**15.48**	-14.04	-86.05
	AUD/USD	**16.68**	-48.96	-74.67		AUD/USD	10.54	**26.53**	-36.39
	CAD/JPY	**12.63**	-20.58	-42.07		CAD/JPY	**6.05**	-19.47	-37.47
	CHF/JPY	**7.46**	-12.98	-31.7		CHF/JPY	**12.91**	-12.3	-86.67
	EUR/CHF	**2.67**	-11.02	-100.0		EUR/CHF	**1.94**	-68.91	-100.0
	EUR/GBP	**-0.51**	-8.68	-99.98		EUR/GBP	**2.44**	-14.25	-99.95
	EUR/JPY	**2.51**	-19.93	-99.99		EUR/JPY	**3.99**	-7.62	-100.0
	EUR/USD	15.52	**55.7**	27.79		EUR/USD	12.53	28.14	**44.13**
	GBP/JPY	**6.56**	-15.61	-96.82		GBP/JPY	**8.33**	-8.06	-97.22
	GBP/USD	**13.56**	-27.14	-58.36		GBP/USD	**17.34**	-36.11	-66.77
	NZD/USD	**5.89**	-42.72	-92.5		NZD/USD	**12.13**	-51.49	-93.35
	USD/CAD	**4.27**	-2.77	-11.81		USD/CAD	5.76	-4.06	**8.31**
	USD/CHF	**4.63**	-14.76	-92.11		USD/CHF	**10.1**	-44.27	-97.18
	USD/JPY	**14.38**	-37.69	-35.6		USD/JPY	-22.8	**-7.62**	-30.75

SADRL consistently outperforms PADRL and FDRL when looking at total return with the best performing results in 84% of cases. SADRL total returns

Table 6.9: Comparison of Total Return (%) for B&H, MAC and RSI. Best value per currency pair is shown in boldface.

Pair	B&H	MAC	RSI
AUD/JPY	-6.8	-1.12	**0.67**
AUD/USD	-4.73	-8.66	**1.49**
CAD/JPY	-8.17	-2.1	**3.84**
CHF/JPY	1.71	**5.85**	-3.85
EUR/CHF	1.76	-4.49	**2.62**
EUR/GBP	0.96	**2.36**	0.89
EUR/JPY	**3.04**	1.83	2.8
EUR/USD	0.56	**6.53**	-0.1
GBP/JPY	2.3	**6.69**	4.84
GBP/USD	-3.18	**1.22**	-2.39
NZD/USD	-1.59	-11.46	**-0.53**
USD/CAD	**5.95**	-2.55	0.79
USD/CHF	-3.99	**7.67**	-3.81
USD/JPY	4.06	**15.25**	-3.48

Table 6.10: Statistical test results for total return, according to the non-parametric Friedman test with the Conover post hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

Friedman test p-value		**3.743e-54**
	Ave. Rank	p_{Con}
SADRL(c)	1.88	-
MAC	2.88	**1.072e-08**
RSI	3.00	**7.631e-14**
B&H	3.24	**8.525e-16**
PADRL	4.61	**2.817e-51**
FDRL	5.39	**1.866e-72**

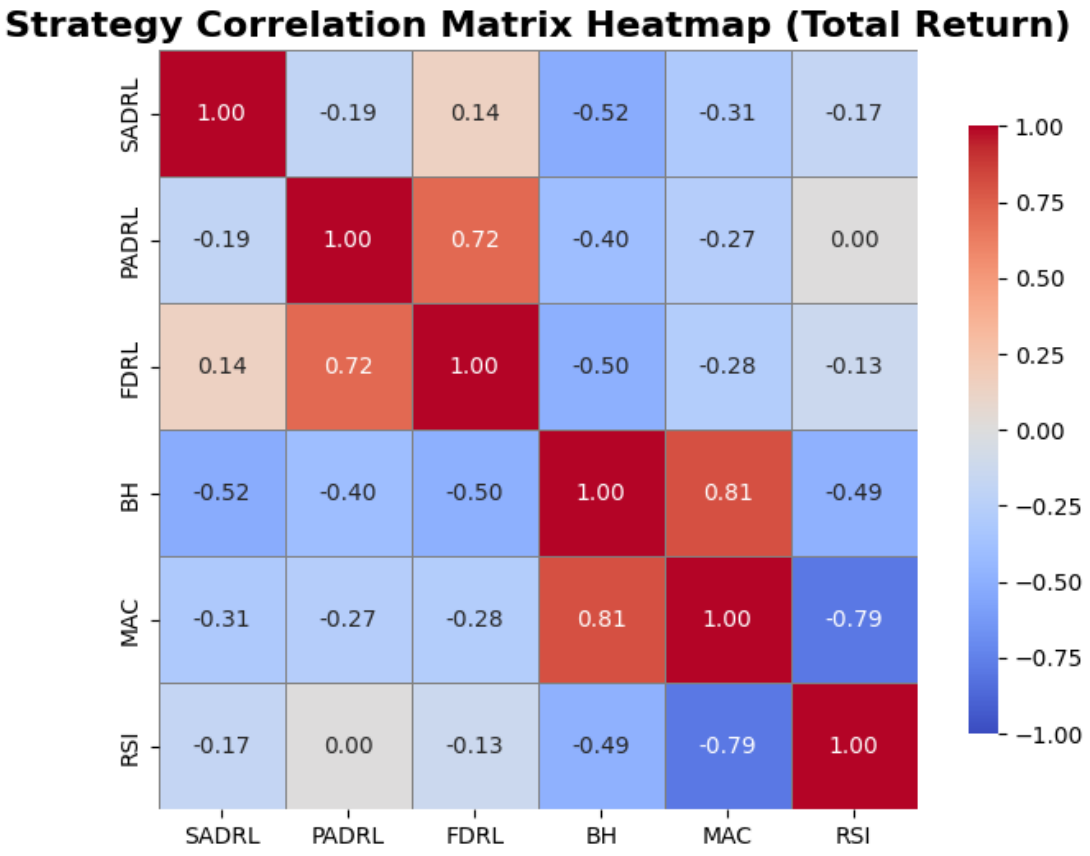

Figure 6.3: Total Return Strategy Correlation

appear to be much more stable than PADRL and FDRL as well with a standard deviation of 11.47 compared to 23.48 for PADRL and 46.20 for FDRL. Given the

mean return of SADRL is 5.45% across all pair-thresholds compared to -15.39% for PADRL and -57.90 for FDRL it is clear that SADRL generates much more return than PADRL and FDRL. Table 6.9 shows that RSI and MAC often outperform B&H with B&H, MAC and RSI producing the best total return on 2, 7 and 5 of the pairs respectively. Of these three benchmarks however, results very rarely exceed those of SADRL. This in mostly due to SADRL producing reliably positive results across most pairs on all thresholds while all benchmarks produce negative returns on 6 of the 14 pairs, clearly favouring SADRL.

Table 6.10 presents the results of the non-parametric Friedman test and the Conover post hoc test, which assesses and compares the average rankings between SADRL and all benchmark models for total return. SADRL achieves the best average rank with a value of 1.88, the resultant p-value from the Friedman test also show this as being statistically significant with a value of 3.743e-54, comfortably surpassing the 0.05 significance level meaning the null hypothesis stating that the results are statistically insignificant can be rejected. The p-values from the post hoc Conover test confirm that the SADRL total return results are statistically significant with SADRL producing a p-value of less than 0.05 when compared to every other trading strategy.

Maximum Drawdown Similarly to total return, the maximum drawdown results in Table 6.11 demonstrate how poorly PADRL and FDRL perform with only 11% of FDRL simulations outperforming PADRL and only one PADRL simulation outperforming FDRL and SADRL. FDRL has a number of cases where the whole initial balance is lost and therefore presents a rounded maximum drawdown of 100%, this is particularly clear for AUD/USD, EUR/CHF, EUR/GBP, GBP/USD, NZD/USD and USD/CHF. Of the pairs that do not result in an almost bust account, EUR/JPY, EUR/USD and USD/CAD consistently produce drawdowns that are close to or greater than 50% across all thresholds. The remaining pairs of AUD/JPY, CAD/JPY, CHF/JPY, GBP/JPY and USD/JPY produce maximum drawdowns competitive with SADRL and PADRL. Table 6.12

presents the maximum drawdown results for both the MAC and RSI strategies, showing that RSI outperforms MAC across all currency pairs except USD/JPY. In comparison, SADRL proves to be the more robust approach, consistently achieving maximum drawdowns below 1% a level of risk control neither benchmark strategy is able to match.

PADRL has much less variance in maximum drawdown than FDRL with a standard deviation of 15.13% compared to the 41.48% of FDRL. Although this results in a favourable average rank, PADRL only outperforms FDRL and SADRL on EUR/USD at 0.025% with a maximum drawdown of 1.41%. The mean maximum drawdown of PADRL is 23.57% with a median of 18.88%, compared to FDRL which has a mean of 59.19% and median of 71.54%, these results however do not provide the full picture as Table 6.13 shows that both FDRL and PADRL have a very similar average rank of 4.14 and 4.19 respectively, indicating that the mean and median of FDRL is skewed by the extreme values that are observed for certain pairs, ultimately resulting in a similar aggregated performance to PADRL.

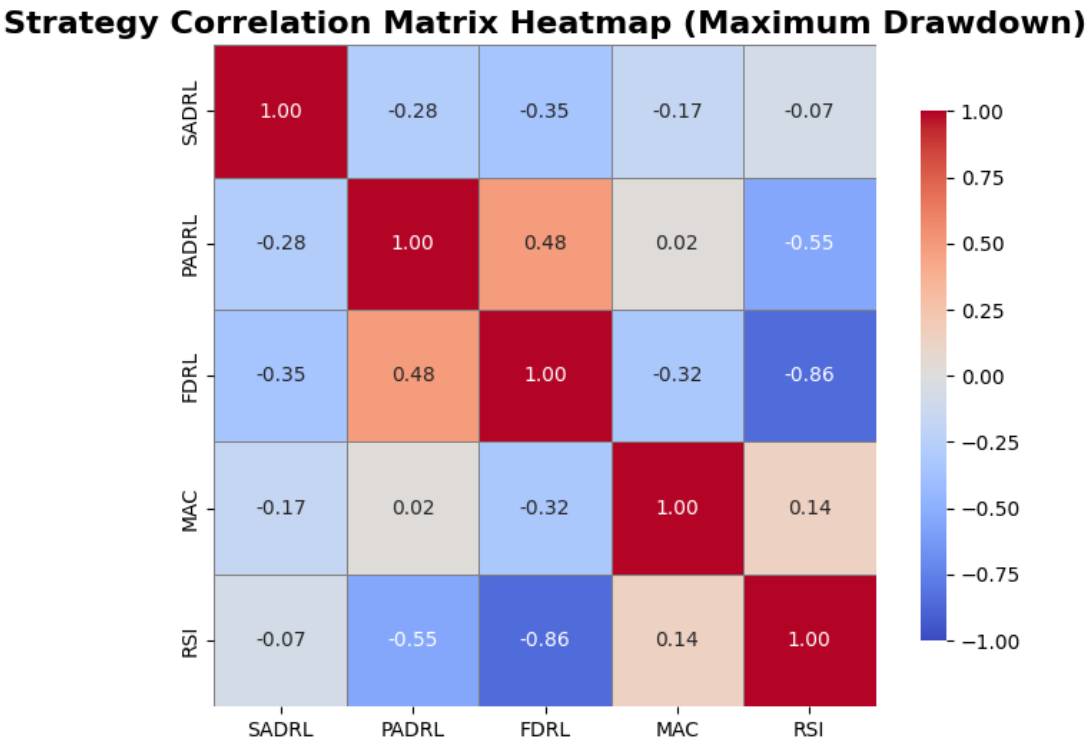

Figure 6.4: Maximum Drawdown Strategy Correlation

Both PADRL and FDRL are convincingly outperformed by SADRL as is clear

Table 6.11: Comparison of Maximum Drawdown (%) by DC threshold for SADRL, PADRL and FDRL. Best value per currency pair at each DC threshold is denoted in boldface and the best value per DC threshold is underlined.

θ	Pair	SADRL	PADRL	FDRL	θ	Pair	SADRL	PADRL	FDRL
0.015%	AUD/JPY	**0.48**	18.52	25.41	0.023%	AUD/JPY	22.44	**16.88**	51.58
	AUD/USD	**1.74**	11.89	99.99		AUD/USD	**1.24**	13.03	99.33
	CAD/JPY	**0.55**	17.27	11.68		CAD/JPY	**0.48**	19.23	62.25
	CHF/JPY	**0.64**	13.54	59.81		CHF/JPY	**1.11**	14.7	76.04
	EUR/CHF	**1.31**	21.19	99.99		EUR/CHF	**1.26**	33.43	100.0
	EUR/GBP	**1.8**	38.59	99.99		EUR/GBP	**2.65**	29.64	100.0
	EUR/JPY	**2.67**	42.93	99.99		EUR/JPY	**1.36**	23.8	100.0
	EUR/USD	**0.24**	18.68	58.3		EUR/USD	**0.2**	5.73	68.59
	GBP/JPY	**0.45**	12.06	61.26		GBP/JPY	29.85	**13.56**	88.51
	GBP/USD	**0.35**	29.25	99.99		GBP/USD	**1.54**	24.57	99.99
	NZD/USD	**1.74**	43.18	99.99		NZD/USD	**3.48**	43.59	99.99
	USD/CAD	**1.75**	10.68	91.19		USD/CAD	**0.91**	34.25	84.66
	USD/CHF	**0.79**	15.08	99.99		USD/CHF	**0.19**	12.42	99.99
	USD/JPY	**0.72**	19.18	3.57		USD/JPY	**10.97**	39.04	32.2
0.017%	AUD/JPY	**1.38**	13.65	32.04	0.025%	AUD/JPY	**1.18**	11.53	86.92
	AUD/USD	**1.31**	7.82	99.93		AUD/USD	**2.45**	10.36	94.07
	CAD/JPY	**0.42**	21.69	26.15		CAD/JPY	**1.08**	19.09	40.76
	CHF/JPY	**6.43**	14.84	30.28		CHF/JPY	**0.58**	11.22	51.9
	EUR/CHF	**0.84**	27.06	99.99		EUR/CHF	**0.67**	10.84	100.0
	EUR/GBP	**1.45**	17.08	99.99		EUR/GBP	**2.32**	26.88	99.99
	EUR/JPY	**1.99**	57.9	99.99		EUR/JPY	**2.08**	30.09	100.0
	EUR/USD	**1.14**	5.2	63.45		EUR/USD	**0.43**	1.41	47.76
	GBP/JPY	**1.12**	12.79	34.77		GBP/JPY	**2.03**	13.53	78.37
	GBP/USD	**1.8**	43.38	99.99		GBP/USD	**0.54**	62.24	99.99
	NZD/USD	**2.9**	44.95	99.99		NZD/USD	**3.65**	37.03	99.99
	USD/CAD	**1.58**	15.42	76.97		USD/CAD	**1.76**	16.85	81.6
	USD/CHF	**1.32**	10.17	99.99		USD/CHF	**0.75**	14.54	99.99
	USD/JPY	**0.49**	26.81	52.95		USD/JPY	**1.21**	5.83	47.49
0.019%	AUD/JPY	**0.78**	19.7	37.64	0.027%	AUD/JPY	**0.58**	16.17	66.81
	AUD/USD	**1.51**	10.74	84.5		AUD/USD	**0.55**	64.86	91.58
	CAD/JPY	**0.2**	16.6	24.71		CAD/JPY	**1.42**	13.52	80.59
	CHF/JPY	**1.31**	11.63	60.43		CHF/JPY	**0.57**	12.33	72.61
	EUR/CHF	**1.07**	29.83	100.0		EUR/CHF	**0.68**	19.41	100.0
	EUR/GBP	**2.01**	21.56	99.99		EUR/GBP	**1.42**	24.08	99.99
	EUR/JPY	**4.89**	31.35	100.0		EUR/JPY	**24.31**	31.1	100.0
	EUR/USD	**0.55**	9.09	47.75		EUR/USD	**0.62**	22.09	80.4
	GBP/JPY	**1.83**	10.97	71.69		GBP/JPY	**0.76**	77.68	97.49
	GBP/USD	**1.24**	47.87	99.99		GBP/USD	**0.69**	24.78	99.99
	NZD/USD	**4.22**	22.99	99.99		NZD/USD	**7.87**	39.74	99.99
	USD/CAD	**2.11**	27.36	66.46		USD/CAD	**2.28**	7.77	40.61
	USD/CHF	**0.97**	18.15	99.99		USD/CHF	**1.23**	22.92	99.99
	USD/JPY	**8.11**	18.48	66.97		USD/JPY	40.96	**12.0**	68.74
0.021%	AUD/JPY	**0.48**	11.29	44.57	0.029%	AUD/JPY	**0.83**	14.04	86.05
	AUD/USD	**1.07**	60.76	99.99		AUD/USD	**1.64**	30.29	99.99
	CAD/JPY	**0.55**	20.58	42.07		CAD/JPY	**0.5**	19.47	37.47
	CHF/JPY	**1.0**	13.1	31.7		CHF/JPY	**0.09**	12.41	86.67
	EUR/CHF	**1.54**	27.51	100.0		EUR/CHF	**0.66**	73.46	100.0
	EUR/GBP	**3.12**	14.1	99.99		EUR/GBP	**1.99**	25.68	99.99
	EUR/JPY	**1.78**	30.39	99.99		EUR/JPY	**1.31**	40.5	100.0
	EUR/USD	**0.79**	5.89	40.49		EUR/USD	**0.68**	10.99	88.64
	GBP/JPY	**1.65**	15.85	96.82		GBP/JPY	**2.76**	8.23	97.22
	GBP/USD	**1.43**	27.93	99.94		GBP/USD	**1.94**	36.17	99.99
	NZD/USD	**0.7**	42.72	99.99		NZD/USD	**2.96**	51.49	99.99
	USD/CAD	**1.07**	18.54	74.48		USD/CAD	**3.02**	11.92	76.1
	USD/CHF	**0.07**	14.76	99.99		USD/CHF	**1.47**	44.27	99.99
	USD/JPY	**1.3**	38.69	35.6		USD/JPY	30.67	**8.1**	30.75

from both the results in Table 6.11 and Table 6.13. SADRL produces the lowest maximum drawdown in 88% of cases and in the cases when SADRL does not outperform FDRL or PADRL it is always the second best trading strategy at that pair-threshold combination. SADRL also produces the lowest maximum draw-

Table 6.12: Comparison of Maximum Drawdown (%) for MAC and RSI. Best value per currency pair is shown in boldface.

Pair	MAC	RSI
AUD/JPY	10.21	**3.97**
AUD/USD	17.08	**4.55**
CAD/JPY	13.93	**5.02**
CHF/JPY	10.00	**7.79**
EUR/CHF	7.74	**1.79**
EUR/GBP	4.59	**3.15**
EUR/JPY	12.24	**3.68**
EUR/USD	5.78	**3.70**
GBP/JPY	6.40	**5.48**
GBP/USD	12.41	**8.97**
NZD/USD	12.60	**2.51**
USD/CAD	5.52	**2.61**
USD/CHF	6.39	**5.29**
USD/JPY	**8.03**	8.26

Table 6.13: Statistical test results for maximum drawdown, according to the non-parametric Friedman test with the Conover post hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

Friedman test p-value		**2.379e-83**
	Ave. Rank	p_{Con}
SADRL(c)	1.22	-
RSI	1.98	**1.055e-21**
MAC	2.97	**1.033e-73**
PADRL	3.88	**4.336e-128**
FDRL	4.94	**3.508e-187**

down at every DC threshold with maximum drawdowns of 1.12%, 1.15%, 1.39%, 1.38%, 1.07%, 2.27%, 1.41% and 1.94% for DC thresholds 0.015% up to 0.029% respectively. The mean of SADRL maximum drawdowns is 4.17% and the median is 3.11%, a considerable improvement on the previously stated mean and median values of FDRL and PADRL.

Table 6.13 shows the results of the non-parametric Friedman test and the Conover post hoc test between SADRL and all benchmarks for maximum drawdown, the average rankings for each strategy are also shown. SADRL achieves the highest average rank with a value of 1.57, an improvement on its next closest rival of RSI by 0.33. MAC, FDRL and PADRL achieve average ranks of 3.20, 4.14 and 4.19 accordingly. The results of the non-parametric Friedman test show that the results are statistically significant as all p-values do not exceed the significance threshold of 5%. From these results it can be deduced that SADRL significantly outperforms both the previously created DRL benchmarks of FDRL and PADRL, as well as the technical analysis benchmarks of MAC and RSI with respect to maximum drawdown. The null hypothesis stating that the results are statistically insignificant is therefore rejected.

Table 6.14: Comparison of Calmar Ratio by DC threshold for SADRL, PADRL and FDRL. Best value per currency pair at each DC threshold is denoted in boldface and the best value per threshold is underlined.

θ	Pair	SADRL	PADRL	FDRL	θ	Pair	SADRL	PADRL	FDRL
0.015%	AUD/JPY	**10.8**	-1.0	-1.0	0.023%	AUD/JPY	**-0.76**	-0.99	-1.0
	AUD/USD	**3.08**	1.7	-0.49		AUD/USD	**14.3**	2.79	0.26
	CAD/JPY	**5.61**	-1.0	-1.0		CAD/JPY	**16.57**	-1.0	-1.0
	CHF/JPY	**0.09**	-1.0	-1.0		CHF/JPY	**4.26**	-1.0	-1.0
	EUR/CHF	**1.52**	-1.0	-0.99		EUR/CHF	**0.69**	-0.69	-1.0
	EUR/GBP	**1.81**	-0.75	-1.0		EUR/GBP	**1.15**	-0.84	-1.0
	EUR/JPY	**-0.01**	-0.76	-1.0		EUR/JPY	**4.55**	-0.53	-1.0
	EUR/USD	**38.84**	-1.0	0.54		EUR/USD	**86.91**	7.4	0.81
	GBP/JPY	**1.58**	-0.98	-1.0		GBP/JPY	**-0.82**	-0.98	-1.0
	GBP/USD	**48.94**	-0.97	-0.92		GBP/USD	**8.18**	-0.98	-0.87
	NZD/USD	**7.06**	-0.99	-0.71		NZD/USD	**2.18**	-1.0	-0.75
	USD/CAD	**3.36**	-0.24	-0.17		USD/CAD	**5.0**	-0.67	-0.16
	USD/CHF	**7.88**	-0.98	-0.99		USD/CHF	**71.86**	-1.0	-0.97
	USD/JPY	**13.56**	-1.0	-1.0		USD/JPY	**0.0**	-0.99	-1.0
0.017%	AUD/JPY	**4.34**	-0.99	-1.0	0.025%	AUD/JPY	**8.04**	-1.0	-1.0
	AUD/USD	**10.7**	7.11	-0.35		AUD/USD	**4.37**	3.56	0.06
	CAD/JPY	**19.74**	-1.0	-1.0		CAD/JPY	**3.52**	-1.0	-1.0
	CHF/JPY	**-0.27**	-1.0	-1.0		CHF/JPY	**10.15**	-1.0	-1.0
	EUR/CHF	**3.93**	-1.0	-1.0		EUR/CHF	**2.84**	-1.0	-1.0
	EUR/GBP	**2.17**	-0.99	-1.0		EUR/GBP	**0.34**	-0.35	-1.0
	EUR/JPY	**0.46**	-0.88	-1.0		EUR/JPY	**0.99**	-0.36	-1.0
	EUR/USD	**7.49**	2.98	-0.14		EUR/USD	**25.48**	17.1	1.6
	GBP/JPY	**5.69**	-1.0	-1.0		GBP/JPY	**5.79**	-0.99	-1.0
	GBP/USD	**13.97**	-1.0	-0.99		GBP/USD	**21.1**	-1.0	-0.86
	NZD/USD	**3.24**	-1.0	-0.86		NZD/USD	**3.68**	-1.0	-0.87
	USD/CAD	**4.36**	-0.99	-0.15		USD/CAD	**3.1**	-0.49	-0.15
	USD/CHF	**2.57**	-0.98	-0.89		USD/CHF	**6.72**	-1.0	-0.97
	USD/JPY	**8.89**	-0.99	-1.0		USD/JPY	**10.46**	-0.98	-1.0
0.019%	AUD/JPY	**9.75**	-1.0	-1.0	0.027%	AUD/JPY	**10.57**	-0.98	-1.0
	AUD/USD	4.51	**5.74**	0.44		AUD/USD	**23.86**	-0.85	0.72
	CAD/JPY	**32.56**	-1.0	-1.0		CAD/JPY	**5.29**	-1.0	-1.0
	CHF/JPY	**6.16**	-0.99	-1.0		CHF/JPY	**16.61**	-0.99	-1.0
	EUR/CHF	**3.14**	-0.64	-1.0		EUR/CHF	**3.26**	-1.0	-1.0
	EUR/GBP	**0.15**	-1.0	-1.0		EUR/GBP	**2.2**	-0.51	-1.0
	EUR/JPY	**0.82**	-0.57	-1.0		EUR/JPY	**0.51**	-0.48	-1.0
	EUR/USD	**8.79**	2.5	1.44		EUR/USD	**22.11**	0.58	0.48
	GBP/JPY	**4.69**	-0.98	-1.0		GBP/JPY	**18.54**	-1.0	-1.0
	GBP/USD	**4.93**	-1.0	-0.91		GBP/USD	**10.37**	-1.0	-0.95
	NZD/USD	**0.92**	-0.45	-0.86		NZD/USD	**0.67**	-1.0	-0.96
	USD/CAD	**1.92**	-1.0	-0.02		USD/CAD	**2.36**	1.02	0.4
	USD/CHF	**10.35**	-1.0	-0.97		USD/CHF	**1.77**	-0.99	-0.98
	USD/JPY	**-0.41**	-0.97	-1.0		USD/JPY	**-0.92**	-1.0	-1.0
0.021%	AUD/JPY	**17.97**	-1.0	-1.0	0.029%	AUD/JPY	**18.61**	-1.0	-1.0
	AUD/USD	**15.52**	-0.81	-0.75		AUD/USD	**6.41**	0.88	-0.36
	CAD/JPY	**23.05**	-1.0	-1.0		CAD/JPY	**12.03**	-1.0	-1.0
	CHF/JPY	**7.44**	-0.99	-1.0		CHF/JPY	**137.23**	-0.99	-1.0
	EUR/CHF	**1.74**	-0.4	-1.0		EUR/CHF	**2.96**	-0.94	-1.0
	EUR/GBP	**-0.16**	-0.62	-1.0		EUR/GBP	**1.23**	-0.55	-1.0
	EUR/JPY	**1.41**	-0.66	-1.0		EUR/JPY	**3.04**	-0.19	-1.0
	EUR/USD	**19.58**	9.46	0.69		EUR/USD	**18.47**	2.56	0.5
	GBP/JPY	**3.97**	-0.98	-1.0		GBP/JPY	**3.02**	-0.98	-1.0
	GBP/USD	**9.46**	-0.97	-0.58		GBP/USD	**8.96**	-1.0	-0.67
	NZD/USD	**8.43**	-1.0	-0.92		NZD/USD	**4.1**	-1.0	-0.93
	USD/CAD	**4.0**	-0.15	-0.16		USD/CAD	**1.9**	-0.34	0.11
	USD/CHF	**66.53**	-1.0	-0.92		USD/CHF	**6.88**	-1.0	-0.97
	USD/JPY	**11.09**	-0.97	-1.0		USD/JPY	**-0.74**	-0.94	-1.0

Calmar Ratio Table 6.14 shows the Calmar Ratio results of SADRL, PADRL and FDRL across all currency pairs at each DC threshold from 0.015% to 0.029% in steps of 0.002%. Given the nature of the total returns and maximum drawdown results it is no surprise to see FDRL is the worst performing trading strategy with

Table 6.15: Comparison of Calmar Ratio for MAC and RSI. Best value per currency pair is shown in boldface.

Pair	MAC	RSI
AUD/JPY	-0.11	**0.17**
AUD/USD	-0.51	**0.33**
CAD/JPY	-0.15	**0.77**
CHF/JPY	**0.59**	-0.49
EUR/CHF	-0.58	**1.46**
EUR/GBP	**0.51**	0.28
EUR/JPY	0.15	**0.76**
EUR/USD	**1.13**	-0.03
GBP/JPY	**1.04**	0.88
GBP/USD	**0.10**	-0.27
NZD/USD	-0.91	**-0.21**
USD/CAD	-0.46	**0.30**
USD/CHF	**1.20**	-0.72
USD/JPY	**1.90**	-0.42

Table 6.16: Statistical test results for Calmar ratio, according to the non-parametric Friedman test with the Conover post hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

Friedman test p-value		**4.719e-61**
	Ave. Rank	p_{Con}
SADRL(c)	1.21	-
RSI	2.63	**5.531e-26**
MAC	2.75	**1.312e-25**
PADRL	4.07	**1.959e-77**
FDRL	4.35	**8.355e-93**

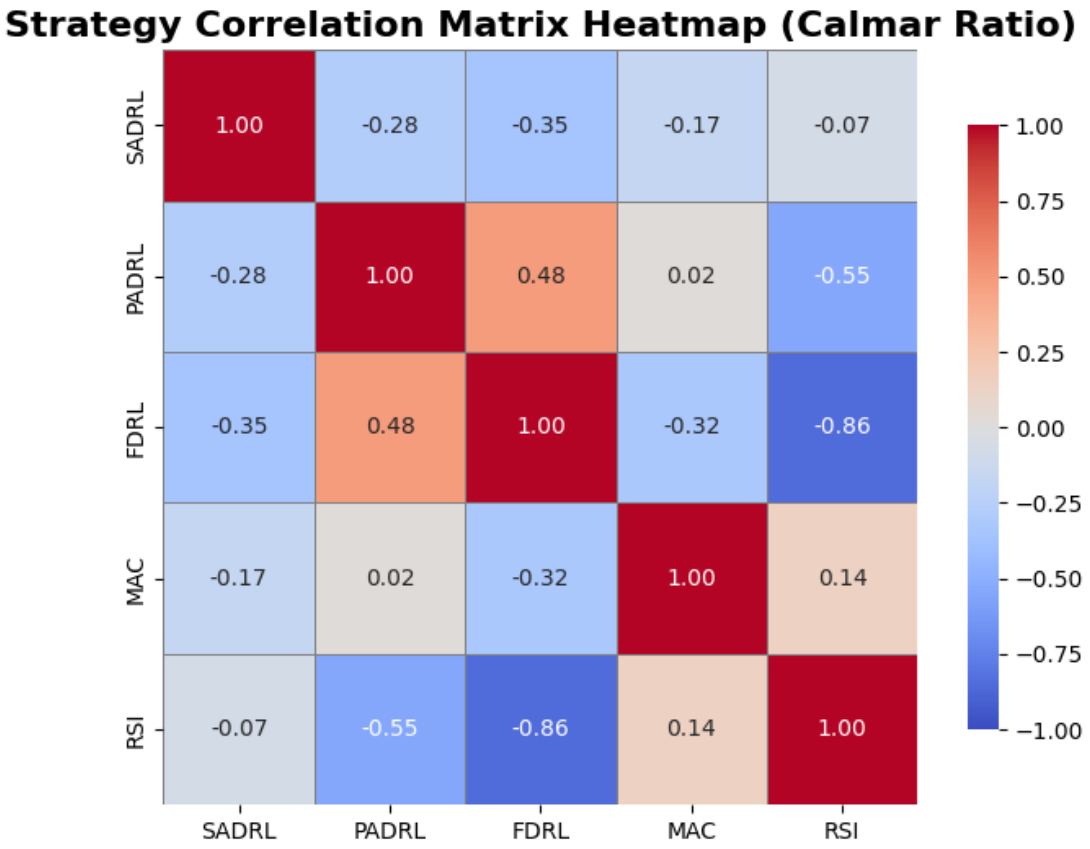

Figure 6.5: Calmar Ratio Strategy Correlation

only one combination generating the best result when compared to all other strategies. Only 13 of the FDRL results produce a positive value of which two are above

the 1.0 threshold where total return is at least equivalent to maximum drawdown. Table 6.16 shows that FDRL achieves an average rank of 4.35, indicating weaker performance compared to the technical analysis benchmarks, as well as SADRL and PADRL, when evaluated under the bid-ask spread.

PADRL yields the best Calmar Ratio on 10% of pair-threshold combinations with 12 of the 112 cases generating a Calmar ratio greater than the 1.0 value. Table 6.16 shows that PADRL has an average rank of 3.93, significantly worse than the 1.69 average ranking of SADRL. PADRL tends to produce mostly poor results apart from a few anomalies on AUD/USD at 0.017%, EUR/USD at 0.021%, EUR/USD at 0.023% and EUR/USD at 0.025% where PADRL achieved Calmar ratios of 7.11, 9.46, 7.40 and 17.10 respectively. PADRL is also consistently outperformed by the technical analysis benchmarks in Table 6.15, shown most clearly by the average rankings in Table 6.16 of 2.47 and 2.57 for RSI and MAC respectively.

SADRL produces the highest Calmar ratio in 90% of cases with some Calmar ratios even achieving values as high as 21.84 and 20.09. The mean Calmar ratio produced by SADRL is 2.47 and the median result is 1.50, both of which are above the 1.0 level used to represent an effective trading strategy. 57% of SADRLs Calmar ratios are above the 1.0 making this a consistent pattern over numerous data sets with only 16% producing a negative Calmar ratio. SADRL also produces an average rank of 1.69, considerably outperforming RSI, MAC, PADRL and FDRL each with an average rank of 2.47, 2.57, 3.93 and 4.35 respectively. SADRL tends to produce healthy Calmar ratios across all thresholds as well, with the highest Calmar ratio at every DC threshold apart from 0.025% where it gets beaten by one of the PADRL anomalies.

Table 6.16 shows the average rankings and results of the non-parametric Friedman test and the Conover post hoc test between SADRL and all benchmarks for Calmar ratio. The p-value of 3.15e-46 from the non-parametric Friedman test demonstrates that the results are statistically significant and the null hypothesis stating that the results are not statistically significant is therefore rejected. The

results of the post hoc Conover test show that SADRL significantly outperforms every benchmark, producing p-values of 2.15e-08, 5.56e-08, 1.42e-44 and 9.05e-60 for RSI, MAC, PADRL and FDRL respectively.

Summary Tables 6.8, 6.11, 6.14 display the results of the trading simulations for three DRL trading strategies, with two technical analysis benchmarks and the passive buy and hold strategy for total return displayed in Tables 6.9, 6.12 and 6.15 for total return, maximum drawdown and Calmar ratio respectively. For SADRL, PADRL and FDRL the results shown are across 14 different currency pairs and 8 different DC thresholds. The benchmarks rely on fixed interval sampling and have been optimised to show the best periods possible. What is immediately clear from the results is the level of performance achieved by SADRL across all three metrics. SADRL tends to outperform all benchmarks across most pairs and every DC threshold with few exceptions. The introduction of spread based transaction costs has impacted the results of PADRL and FDRL significantly as they can very rarely implement that same strategy as before due to the added complexity of a dynamic transaction cost and tendency for the spread to exceed the magnitude of the DC move. SADRL is therefore much more competitive with the technical analysis benchmarks and the buy and hold strategy as they make fewer trades and therefore incur much lower costs. SADRL still manages to outperform the technical analysis benchmarks with the buy and hold strategy being the only strategy to come close to the SADRL performance on total return but still not achieving a better average rank.

Tables 6.10, 6.13 and 6.16 show the average rank and significance of the total return, maximum drawdown and Calmar ratio results. SADRL is the best performing trading strategy in all three tables with average ranks of 2.33, 1.57 and 1.69 on total return, maximum drawdown and Calmar ratio respectively. SADRLs performance is also statistically significant as shown by the bold results of the non-parametric Friedman tests and the bold p-values for the post hoc Conover tests in every case, the null hypothesis stating that there is no significance between the

tested trading strategies is therefore rejected. From this it can be deduced that, under the trading circumstances defined in the methodology that apply a dynamic transaction cost of the bid-ask spread, SADRL significantly outperforms both its predecessors of FDRL and PADRL as well as the technical analysis benchmarks of RSI and MAC along with the passive buy and hold strategy.

6.4.3 Strategy Benchmarking

The following section compares the performance of SADRL to TA+NoDRL strategies and a TA+DRL strategy defined in Section 6.3.4 to further test whether DC+DRL is the winning combination when setup as the SADRL framework. The SADRL strategy reports results for the TRPO DRL algorithm as it has been observed to be the most effective algorithm for trading in the SADRL system. All strategies are subject to the bid-ask spread during both training and testing. This approach makes results much more realistic as it uses the real prices that the currency pairs were bought and sold at in real time. All non-DRL, fixed interval strategies present results for the frequency of fixed interval sampling which produces the highest return out of 1 minute, 15 minute, hour and 4 hour intervals in order to provide the greatest competition for SADRL as possible. The results displayed in Tables 6.17, 6.19 and 6.21 show the averaged results for SADRL across all DC thresholds to match the number of results for the other fixed interval strategies. When ranking and calculating significance, the results for each pair are expanded out to match the number of SADRL datasets across all DC thresholds.

Total Return Table 6.17 displays the total return results of the SADRL, TADRL, MeanReversion, MACDRSI and BollingerBands strategies. The BollingerBands strategy clearly produced the most impressive results with positive returns for 12/14 currency pairs, with the other two being -0.17% for GBP/USD and -12.12% for USD/JPY. SADRL Produced a positive average total return across all thresholds for 13/14 pairs, with -2.83% for USD/JPY being the one pair that produced

a negative result. Of the remaining strategies, MACDRSI and MeanReversion perform similarly with varying positive and negative results across all 14 currency pairs with MeanReversion producing the highest return of 17.70% on AUD/USD. The TADRL strategy performed poorly across most of the currency pairs, producing mostly negative results with 11/14 currency pairs producing negative returns. This result demonstrates that DRL is a tool that can produce effective results when used with DC sampling as in SADRL but may struggle when applied to currency data sampled at a fixed interval, hence the difference in performance between SADRL and TADRL.

Table 6.17: Total Return (%) across all 14 currency pairs for numerous benchmark strategies testing the effects of removing DRL and DC components. All DC strategies are reported as averages across all DC thresholds (0.015% to 0.029% in steps of 0.02%). The best value per currency pair is denoted in boldface.

Currency Pair	SADRL	TADRL	MACDRSI	MeanRev.	BollingerBands
AUD/JPY	5.17	-14.62	7.02	-5.12	**11.24**
AUD/USD	11.89	6.65	5.57	**17.70**	16.76
CAD/JPY	7.00	-5.73	0.79	**8.98**	7.16
CHF/JPY	5.86	-2.95	0.21	-1.97	**6.76**
EUR/CHF	2.28	-6.78	1.37	1.34	**10.09**
EUR/GBP	1.95	-5.48	2.45	-0.88	**7.65**
EUR/JPY	4.01	-2.00	1.98	5.10	**8.16**
EUR/USD	**11.58**	1.39	9.27	13.32	11.45
GBP/JPY	**3.99**	-2.47	0.23	-9.89	**3.99**
GBP/USD	**13.82**	-1.59	-0.31	-6.84	-0.17
NZD/USD	8.73	-4.59	5.37	4.08	**9.56**
USD/CAD	5.28	-0.86	-0.64	14.00	**9.90**
USD/CHF	6.87	-2.54	3.07	-1.05	**10.10**
USD/JPY	-2.83	**1.96**	-3.55	-12.57	-12.12

Table 6.18: Statistical test results for total return, according to the non-parametric Friedman test with the Conover post hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

Friedman test p-value		**3.568e-40**
	Ave. Rank	p_{Con}
Boll.Bands	1.85	0.221
SADRL(c)	2.20	-
MACDRSI	3.24	**1.213e-08**
MeanRev.	3.24	**1.803e-10**
TADRL	4.47	**1.879e-33**

Table 6.18 displays the significance results of SADRL when compared to the described benchmarks. The null hypothesis of the Friedman significance test states that there is not a significant enough difference in the distribution of the results for each strategy to assume outperformance by any one strategy. Given the p-value of 3.568e-40, the null hypothesis can be safely rejected. Table 6.18 also presents the rankings of each trading strategy along with their p-values in relation to the SADRL strategy. These results show that although the BollingerBands strategy produced a higher average rank of 1.85 than SADRL with an average rank of 2.20, this was not statistically significant as this produced a p-value of 0.221, failing to provided a low enough p-value to reject the null hypothesis.

Maximum Drawdown Table 6.19 displays the results of the total return when backtesting all benchmark strategies. SADRL produces the lowest maximum drawdown for 7/14 currency pairs, and the second lowest for 5 of the remaining 7 currency pairs. The worst average maximum drawdown for SADRL is 11.80% for USD/JPY and the best is 0.58% for EUR/USD, a value which is not beaten by any of the other strategies. The MACDRSI Strategy appears the most consistent strategy producing a maximum drawdown of between 0.18% and 5.23% across all 14 currency pairs. The same however is not the case for TADRL, MeanReversion and BollingerBands which have much more variation in their maximum drawdown results with each strategy producing highs of 14.62%, 20.81% and 14.81% respectively.

Table 6.20 presents the significance results of SADRL when compared to the described benchmarks. A p-value of 4.831e-54 from the Friedman test indicates that the null hypothesis, stating that there is no significant difference in the distribution of results across strategies, can be rejected. The average ranking and poshoc Conover p-values shown in Table 6.20 show that SADRL ranks the best across all strategies with an average rank of 1.65. SADRL statistically outperforms the TADRL, BollingerBands and MeanReversion strategies which all produce p-values below the 5% significance threshold required in order to be considered statistically

Table 6.19: Maximum Drawdown (%) across all 14 currency pairs for numerous benchmark strategies testing the effects of removing DRL and DC components. All DC strategies are reported as averages across all DC thresholds (0.015% to 0.029% in steps of 0.02%). The best value per currency pair is denoted in boldface.

Currency Pair	SADRL	TADRL	MACDRSI	MeanRev.	BollingerBands
AUD/JPY	3.52	14.62	**1.72**	9.48	7.49
AUD/USD	**1.44**	1.61	2.82	6.15	3.88
CAD/JPY	**0.65**	5.79	1.11	9.69	5.09
CHF/JPY	**1.47**	3.59	2.66	15.71	4.59
EUR/CHF	1.00	7.09	**0.18**	5.05	4.05
EUR/GBP	2.09	6.99	**1.76**	5.16	5.25
EUR/JPY	5.05	4.10	**1.36**	15.69	5.61
EUR/USD	**0.58**	3.59	0.80	5.77	6.38
GBP/JPY	5.06	6.51	**0.86**	18.00	5.85
GBP/USD	**1.19**	5.65	2.37	15.05	11.82
NZD/USD	3.44	4.83	**2.42**	8.84	5.37
USD/CAD	**1.81**	2.17	2.86	4.96	2.64
USD/CHF	**0.85**	4.21	4.21	10.32	7.41
USD/JPY	11.80	**3.89**	5.23	20.81	14.81

Table 6.20: Statistical test results for maximum drawdown, according to the non-parametric Friedman test with the Conover post hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

Friedman test p-value		**4.831e-54**
	Ave. Rank	p_{Con}
SADRL(c)	1.65	-
MACDRSI	2.01	0.058
TADRL	3.14	**3.875e-35**
Boll.Bands	3.62	**4.083e-52**
MeanRev.	4.58	**3.067e-84**

significant. The MACDRSI strategy however produced a p-value of 0.058, meaning this can only be accepted as significant at a p-value slightly higher than 5%.

Calmar Ratio Table 6.21 displays the results of the total return when back-testing all benchmark strategies. SADRL is clearly the best performing algorithm with the highest average Calmar Ratio for 11/14 currency pairs. TADRL is the worst performing algorithm with 5/14 currency pairs producing a Calmar Ratio worse than -0.8. The remaining three strategies produce some good results with 11/14 BollingerBands Calmar Ratios producing values above 1.0, with MACDRSI and MeanReversion producing 3 and 11 instances of the same respectively.

Table 6.21: Calmar Ratio across all 14 currency pairs for numerous benchmark strategies testing the effects of removing DRL and DC components. All DC strategies are reported as averages across all DC thresholds (0.015% to 0.029% in steps of 0.02%). The best value per currency pair is denoted in boldface.

Currency Pair	SADRL	TADRL	MACDRSI	MeanRev.	BollingerBands
AUD/JPY	**9.91**	-1.00	4.08	-0.54	1.50
AUD/USD	**10.34**	4.13	1.97	2.88	4.32
CAD/JPY	**14.80**	-0.99	0.71	0.93	1.41
CHF/JPY	**22.71**	-0.82	0.08	-0.13	1.47
EUR/CHF	2.51	-0.96	**7.58**	0.27	2.50
EUR/GBP	1.11	-0.78	1.39	-0.17	**1.46**
EUR/JPY	**1.47**	-0.49	1.46	0.32	1.45
EUR/USD	**28.46**	0.39	11.58	2.31	1.79
GBP/JPY	**5.31**	-0.38	0.27	-0.55	0.68
GBP/USD	**15.74**	-0.28	-0.13	-0.45	-0.01
NZD/USD	**3.78**	-0.95	2.22	0.46	1.78
USD/CAD	3.25	-0.39	-0.22	2.82	**3.75**
USD/CHF	**21.82**	-0.60	0.73	-0.10	1.36
USD/JPY	**5.24**	0.50	-0.68	-0.60	-0.82

Table 6.22: Statistical test results for Calmar ratio, according to the non-parametric Friedman test with the Conover post hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

Friedman test p-value		**1.181e-45**
	Ave. Rank	p_{Con}
SADRL(c)	1.65	-
Boll.Bands	2.40	**6.613e-10**
MACDRSI	2.78	**2.110e-13**
MeanRev.	3.73	**4.865e-32**
TADRL	4.44	**3.937e-61**

Table 6.22 presents the significance results of SADRL when compared to the described benchmarks. Given a p-value of 1.181e-45 the null hypothesis, stating that all results are from the same distribution, can be rejected, implying that no trading strategy outperforms another. The average rank results show that SADRL outperforms all other benchmark strategies with an average rank of 1.65, 0.75 greater than that of it's closest rival BollingerBands, at 2.40. Comparison of the post hoc p-values for each strategy reveals that all results are statistically significant, with p-values ranging from 6.613e-10 to 3.937e-61.

These results suggest that SADRL is the significantly the best performing

algorithm when considering the risk-reward trade-off of each trading strategy. Although SADRL is outperformed by BollingerBands for total return, these results were not significant, BollingerBands also produces significantly worse maximum drawdown result. This demonstrates that the combination of DC sampling and DRL can outperform common technical analysis benchmark strategies as well as a technical analysis benchmark that is trained using the same indicators and methodology on the same data sampled at a fixed interval.

6.5 Interpretation

The following interpretation section focusses on the trading behaviour of the SADRL trading algorithm to gain a better understanding of some of the decisions made by the SADRL agents. The opaque nature of neural networks often masks the reasons behind the decisions made by these DRL trading agents so analysing their trades in more detail will uncover some of the market conditions that influence certain trades.

6.5.1 Trading Behaviour

Figure 6.6 shows the trades made by SADRL over February of 2023, this is a good example of the type of market the agents thrive in. All agents tend to learn a profit taking strategy, meaning they hold on to trades for extended periods of time when the price is trending and will take profit once they reach a desired profit thresholds. Figure 6.6 illustrates that the agent tends to hold trades for shorter durations during sharp price movements, often taking profit following a rebound to a level above the entry point.

This trading behaviour is a stark contrast from the trading behaviour exhibited by the FDRL and PADRL models where the sharp and frequent changes in price are used as the main source of profit. The bid-ask spread limitations prevent this behaviour for SADRL which instead learns to focus on a longer term holding

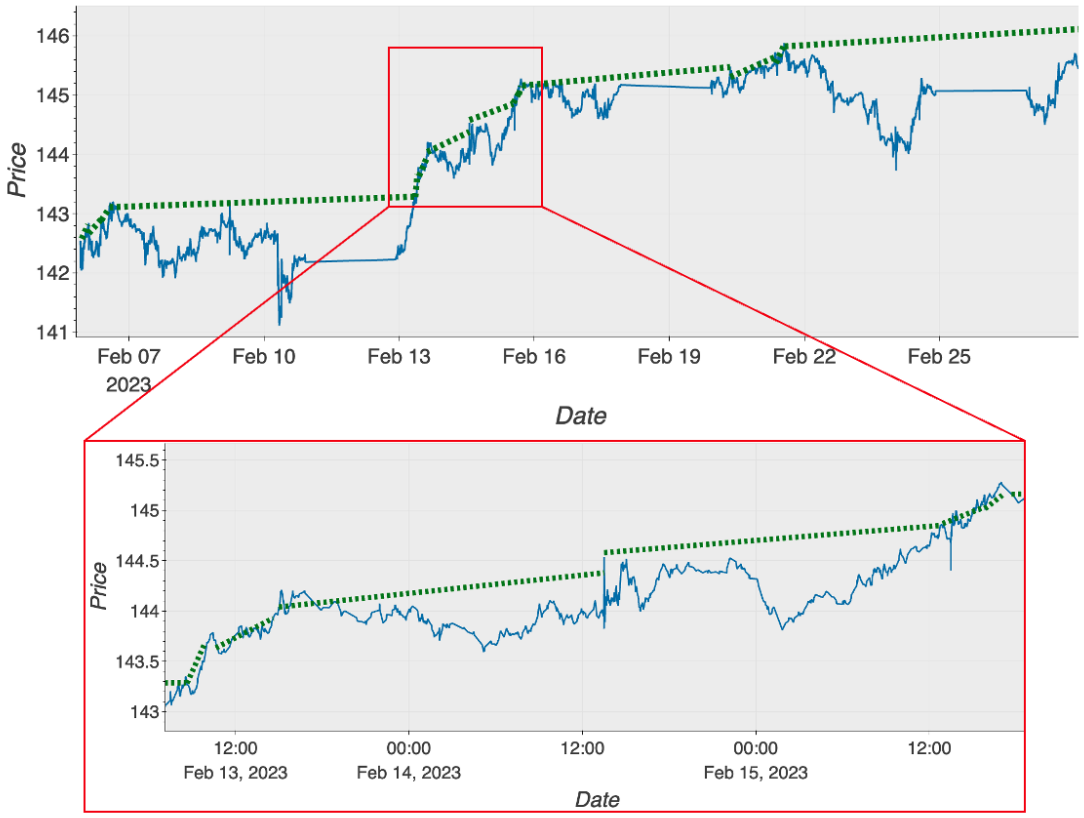

Figure 6.6: Trading behaviour of a CHF/JPY at $\theta = 0.029\%$ over February 2023. The blue line represents the series of DCC prices at each event. The green dashed lines represent profitable trades over the period they are plotted and are mapped from the entry and and exit prices (buy at ask price and sell at bid price) which deviate slightly from the DCC price, which is mapped to the mid price of the bid. The extended period of the flat prices are when the market is closed for the weekend between Friday 10pm GMT and Sunday 10pm GMT.

strategy.

SADRL performs particularly well in trending environments but not so well when the trend reverses. Figure 6.7 provides an example of how SADRL performs when markets trend in the opposite direction. At the bottom of the first zoomed-in image, the final movement of the downward trend is visible, followed by a reversal into an uptrend before stabilising into a sideways trend. SADRL makes significant gains on the preceding downtrend but this uptrend causes a losing trade to be taken before exiting with a loss and re-entering another downward position. This pattern is also evident in the final trade within the window, where a significant loss occurs following a sharp price increase, contrary to the agent's apparent trading

preference.

SADRL also has the tendency to hold trades for a prolonged periods of time when the market moves both with it and struggles to identify the correct time to take profit. Figure 6.8 demonstrates this concept well as it shows the first prolonged period where the agent takes a long position and fails to take profit in time and eventually loses money. The agent had numerous opportunities to take profit but avoided them all and eventually settled for when the price returned back to its original price and dropped slightly below and it made a loss. The second long period trade that lasts until the end of the figure demonstrates the same behaviour but in reverse and shows when this behaviour actually helps the agent. A long position is taken before the market drops significantly, the agent then holds the trade for a prolonged period of time before the price rises back to the entry price and the agent manages to exit with only a small loss compared to what could've been if the trade would've carries on falling.

6.6 Summary

This chapter introduces the Spread Aware Deep Reinforcement (SADRL) trading algorithm, a deep reinforcement learning based methodology for training agents to trade FX assets using data sampled with the DC sampling algorithm. It is the third model in a succession of DRL models that have been designed to trade high frequency currency pairs under a directional changes sampling paradigm. Each agent is trained per window across 14 different currency pairs and suggests either a buy or sell action at each DC event which is then executed in a simulated market environment. Unlike previous models that used fixed transaction costs, SADRL explicitly uses historical bid-ask spread data to replicate the buy and sell prices of the asset as close to a realistic market as possible. The bid-ask spread represents the primary transaction cost in foreign exchange markets, with broker commissions either being incorporated into the spread or assumed to be minimal for high volume institutional traders. Slippage remains assumed to be negligible for

Figure 6.7: Trading behaviour of AUD/JPY at $\theta = 0.029\%$ between 19th September to 15th October 2022. The blue line represents the series of DCC prices at each event. The green dashed lines represent profitable trades over the period they are plotted and are mapped from the entry and and exit prices (buy at ask price and sell at bid price) which deviate slightly from the DCC price, which is mapped to the mid price of the bid. The extended period of the flat prices are when the market is closed for the weekend between Friday 10pm GMT and Sunday 10pm GMT.

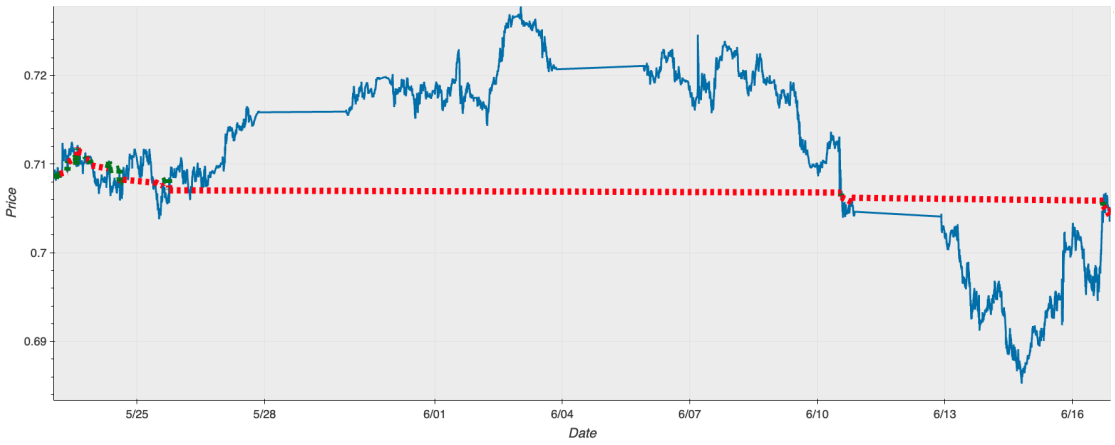

Figure 6.8: Trading behaviour of a AUD/USD at $\theta = 0.029\%$ between 23rd May to 17th June 2023. The blue line represents the series of DCC prices at each event. The green dashed lines represent profitable trades over the period they are plotted and are mapped from the entry and and exit prices (buy at ask price and sell at bid price) which deviate slightly from the DCC price, which is mapped to the mid price of the bid. The extended period of the flat prices are when the market is closed for the weekend between Friday 10pm GMT and Sunday 10pm GMT.

the liquid currency pairs traded, though it is acknowledged that this assumption may not hold during periods of extreme market volatility or for less liquid pairs.

The FDRL and PADRL trading algorithms were the two predecessors to the SADRL trading algorithm and both demonstrated great performance as they generated extreme profit levels for most pairs across all DC thresholds. The main drawback of FDRL not being fully autonomous was handled in the development of PADRL and was carried through into the design of the SADRL trading algorithm. PADRL however also contained drawbacks, the most problematic of which was its fixed transaction cost threshold of 0.035%. Although this would be considered high in many works, the conditions under which PADRL made a significant amount of its profit were when the bid-ask spread widened, meaning the trades that PADRL was able to make were mostly infeasible in the real market. SADRL addressed this problem by enhancing the PADRL with improved training and data preparation methods. This allowed the SADRL to produce positive results that beat all benchmarks when using the bid-ask spread, making this trading algorithm much more realistic.

The SADRL trading algorithm has demonstrated that, with the novel method-

ology described in Section 6.2 a deep reinforcement learning trading system that uses the directional changes sampling method can be built to profitably trade the foreign exchange market at high frequency with realistic transaction costs based on historical spread data. The SADRL extends the current state of the art models in this space by applying these DRL methods using realistic transaction costs, an approach which is rarely taken in the literature at high frequencies due to the resultant reduction in profit.

Chapter 7

Conclusion

This chapter presents a summary of the thesis, highlighting the progressive findings detailed in each contribution chapter. Sections 7.1, 7.2, and 7.3 outline the novelty, contributions and limitations associated with the FDRL, PADRL, and SADRL frameworks respectively. Following the summary of each contribution, Section 7.5 explores potential directions for future research stemming from the findings of this thesis.

7.1 Summary of FDRL

The first contribution chapter of this thesis (Chapter 4), details the development of the Filtered Deep Reinforcement Learning (FDRL) strategy. FDRL is a framework for training a set of deep reinforcement learning agents that are capable of trading the foreign exchange markets at high frequencies. Transaction costs are modelled as a fixed cost of 0.025% representing a combined approximation of commission fees and the bid-ask spread. The novel framework defined by FDRL involves both the preparation of the foreign exchange currency pair price data as well as the training methodology of deep reinforcement learning agents. The data preparation relies on the DC sampling algorithm to sample the raw tick data whenever there is a significant change in price, therefore providing the agent with much cleaner signals to learn from in an effort to aid its ability to learn. The

FDRL framework demonstrates that, using the DC sampling algorithm in tandem with deep reinforcement learning, it is able to train DRL agents to profitably trade under a fixed transaction cost level of 0.025% and outperform the passive buy and hold benchmark as well as the two RSI and MAC technical analysis benchmarks.

The novelty of the FDRL trading strategies lies in the amalgamation of the DC sampling algorithm and a deep reinforcement learning approach to trading. From the literature review in Section 3.3, a number of previous works were identified that use DC sampling effectively on an array of asset classes, including stocks and foreign exchange markets. Many of these applications however used very different types of machine learning algorithms, such as evolutionary algorithms or supervised machine learning algorithms, all of which work at much lower trading frequencies. Section 3.3 did identify two works that apply more traditional table-based reinforcement learning methods to DC sampled data. The application of deep reinforcement learning to DC sampled high frequency foreign exchange data however is the novel application pursued in this thesis.

FDRL uniquely contributes to the research space by creating a framework under which a system of agents can trade profitably across 14 different currency pairs and 8 different DC thresholds. These results provided preliminary evidence that DRL can be used in a real market environment. The extreme returns observed in the testing of the FDRL algorithm were subject to some limitations that would need to be addressed if the strategy were to be transitioned from theory to practice. The results of the algorithm are subject to a 0.025% fixed transaction cost approximating commission and bid-ask spread, which in most cases would be considered quite high however, due to the widening of the actual bid-ask spread at the periods FDRL has learnt to trade, these fixed transaction costs are an underestimate of the dynamic costs that would be faced in real market conditions, making the results somewhat unattainable in a real market environment. Slippage is also assumed to be negligible throughout the analysis, which may not hold during periods of extreme volatility.

Another limitation of FDRL is the integration of a trading filter, by using a trading filter some autonomy was removed from what would be an underperforming model. The trading filter forces the agents to only trade in periods that suited their trading style and suppresses any trading outside of these bounds. Autonomy is important for these agents as it allows them to set strategies and plans that are uninterrupted in execution, by introducing a filter the execution of the strategy is interrupted which in turn disrupts any implicit planning that the agent might be undertaking. The lack of positional awareness for the FDRL agents played a role in hindering the agents learning, which may have caused the need for the rule-based interjection to generate profit. These shortcomings are therefore addressed in the design of the PADRL trading strategy.

7.2 Summary of PADRL

The second contribution of this thesis (Chapter 5) provides the techniques used to develop and test the Positionally Aware Deep Reinforcement Learning (PADRL) framework. PADRL was designed to address the lack of positional awareness of FDRL and ultimately create a fully autonomous trading system that outperforms its predecessor. Previous applications of machine learning in the space of DC-based trading algorithms often rely on a stateless approach to trading, the agents that are used to trade often have no metric to measure their current position in the market. Most of the literature includes strategies that are based on observing data in the market and then generating trading signals, irrespective of the current position, which a rule-based agent then decides whether to act on or not based on incoming variables regarding the agents position, this process is rarely learned and much more often hardcoded. PADRL uniquely contributes to this space by creating a system of agents that can devise trading strategies to account for the agents position in a market as well as the external market variables and then make an informed trading decision.

By employing significant enhancements to the FDRL algorithm, both in terms

of the training algorithm and the environment design centred around the introduction of positional awareness, a more profitable trading strategy can be developed that outperforms FDRL in a more realistic trading environment. This strategy incorporates the positional awareness of PADRL with a concept of spread, allowing the agent to learn a policy that can profitably trade the market while still being subject to the bid-ask spread costs. The concept of self-awareness of the agent is key to the topic of DRL, so by providing positional variables in the state space to allow the agent to understand its current position in the market, the results of PADRL can be profitable at a fixed transaction cost of 0.035%. Despite being much more realistic, the policy learnt by the DRL agents effectively gets round some of the limitations set out by a real market and therefore still suffers from some limitations to application.

The most significant limitation of FDRL and PADRL is the use of a fixed transaction cost. One of the main findings of the research was the difficulty of solving this problem at high frequencies. High frequency data often means small price movements and with small price movements the challenge is not just about capitalising on accurate predictions of price movement, which FDRL and PADRL do very well, but also about identifying when these price movements are large enough to absorb the cost of the bid-ask spread. The next iteration in this series of high frequency trading strategy frameworks is the SADRL framework which addresses the shortcomings of the PADRL framework.

7.3 Summary of SADRL

The third and final contribution of this thesis, in Chapter 6 enhances the PADRL trading strategy by developing the Spread Aware Deep Reinforcement Learning (SADRL) strategy. This strategy incorporates the positional awareness of PADRL with explicit modelling of dynamic bid-ask spreads using historical data, allowing the agent to learn a policy that can profitably trade the market while being subject to realistic transaction costs. Unlike fixed cost models, SADRL accounts for

the time-varying nature of spreads, with commission either incorporated into the spread or assumed minimal for institutional traders, and slippage remaining assumed negligible for liquid pairs. SADRL outperforms both the FDRL and PADRL strategies at a realistic dynamic transaction cost and does so by learning to trade less and focus on longer term trades within the market despite still often trading at a high frequency. SADRL uniquely contributes to this space by building a model training framework that trades profitably when subjected to realistic transactions costs by way of the bid-ask spread.

The SADRL framework incorporates a number of novel features to enhance performance, some of which are novel to this family of directional change and deep reinforcement learning frameworks and others which are novel to the space itself. The introduction of traditional technical analysis indicators is a novel feature that was inspired by the successful use of technical indicators in the literature. The reframing of the DC sampled data using candlesticks is another innovation of SADRL that was inspired by existing techniques but modified slightly by applying to DC sampled data. The introduction of both the technical indicators and the candlestick reframing along with the positional awareness introduced in the earlier PADRL framework help the agents learn longer term dependencies and therefore implement strategies with much more foresight.

The SADRL framework handles the main limitations of both FDRL and PADRL by training agents to implement longer term strategies in order to generate profitable returns under the bid-ask spread transaction costs. SADRL itself also has some limitations however, firstly, most of the strategies implemented by SADRL agents are trend following, meaning they catch onto a trend and perform well in that trend, but when the trend reverses they often struggle to make the correct decisions. Another limitation of SADRL is its ability to optimise trading over prolonged periods. When analysing the behaviour of the SADRL trading agents it was found on a number of occasions that the agent would hold trades while in profit and often wait too long and exit at a small loss instead of taking profit.

These two limitations could form the basis of future work to extend the family of models described in this thesis that executes trades with improved efficiency.

7.4 Comparison of Frameworks

Having outlined the individual contributions and limitations of each framework, this section provides a direct comparison to establish which framework performs best according to specific metrics and under what conditions.

Under fixed transaction costs at 0.035%, PADRL demonstrates clear superiority over FDRL, achieving an average rank of 1.37 for Calmar ratio compared to FDRL's 1.80 (p = 4.139e-3). However, both frameworks deteriorate dramatically when subjected to dynamic bid-ask spreads. FDRL shows particularly poor performance, with losses exceeding 99% in numerous pair-threshold combinations, whilst PADRL produces the best performance in only 9% of pair-threshold combinations. Both frameworks exhibit strong correlations between DC threshold and performance under fixed costs, with FDRL showing particularly strong correlations (correlation coefficients above 0.8 for most pairs), suggesting threshold-dependent strategies that fail to generalise to realistic trading conditions.

When all three frameworks are retrained and tested with bid-ask spreads, SADRL achieves substantial performance advantages across all metrics, producing an average rank of 1.69 for Calmar ratio compared to 3.93 for PADRL and 4.35 for FDRL, with highly significant differences (p-values of 1.42e-44 and 9.05e-60 respectively). SADRL maintains healthy returns across 11 of 14 currency pairs with average Calmar ratios above 1.0, whilst FDRL and PADRL struggle to achieve profitability. For maximum drawdown, SADRL achieves an average rank of 1.57 compared to 3.57 for PADRL and 4.43 for FDRL. SADRL also significantly outperforms traditional technical analysis benchmarks under spread conditions, including Mean Reversion, MACD+RSI, and Bollinger Bands strategies (p-values ranging from 6.613e-10 to 3.937e-61). The fixed-interval DRL benchmark TADRL performs particularly poorly with an average rank of 4.44, confirming that the

combination of DC sampling and deep reinforcement learning provides advantages that neither approach achieves independently.

The choice of reinforcement learning model emerges as critical in the SADRL framework. Whilst FDRL and PADRL employ only PPO, SADRL was evaluated with four different models: DQN, A2C, PPO, and TRPO. TRPO substantially outperforms alternatives across all metrics, with its constrained optimisation approach particularly well-suited to the complex state spaces and sparse reward structures characteristic of spread-aware trading. DQN performs poorly in comparison to policy-based models, whilst A2C demonstrates competitive maximum drawdown performance but falls short on total return and Calmar ratio.

Each framework demonstrates optimal performance under specific conditions. FDRL performs best at higher DC thresholds (0.025%-0.029%) on pairs exhibiting clear directional trends and low volatility oscillations, particularly EUR/CHF, EUR/GBP, and EUR/JPY, generating Calmar ratios exceeding 1,000 for certain pair-threshold combinations. However, these conditions represent a narrow operating envelope that rarely materialises in actual trading environments. PADRL extends FDRL's operating conditions by demonstrating greater consistency across DC thresholds, achieving optimal performance on EUR-based pairs where win rates consistently exceed 55%. PADRL shows weaker correlation between DC threshold and performance compared to FDRL (moderate correlations of 0.3-0.6 rather than strong correlations above 0.8), indicating greater robustness to threshold selection. SADRL demonstrates the broadest operating envelope, performing well across diverse market conditions and achieving particularly strong results on pairs previously problematic for FDRL and PADRL, including CAD/JPY, CHF/JPY, and USD/CHF, with Calmar ratios of 14.80, 22.71, and 21.82 respectively. EUR/USD shows exceptional SADRL performance with a Calmar ratio of 28.46, compared to marginal or negative results for FDRL and PADRL under spread conditions.

The computational requirements differ substantially across frameworks. FDRL represents the most efficient approach, training agents over 200,000 timesteps,

whilst PADRL increases demands through its significant expansion in training duration to 3,000,000 timesteps. SADRL represents the most computationally intensive framework due to TRPO's constrained optimisation and expanded state spaces. The performance advantages under realistic trading conditions justify these computational costs. From a practical deployment perspective, SADRL emerges as the only framework suitable for live trading due to its proven performance under dynamic spread conditions.

In conclusion, SADRL emerges as unequivocally superior when evaluated according to the Calmar ratio under realistic transaction cost conditions, significantly outperforming both its predecessors and traditional benchmarks. FDRL performs best under narrow, rarely materialised conditions at fixed costs. PADRL improves upon FDRL's consistency but remains unsuited to realistic trading. SADRL establishes the benchmark against which future developments in this domain should be measured.

7.5 Future Research

The FDRL, PADRL and SADRL family of frameworks progressively improve with each new model both in terms of performance and how realistic their simulations are however, due to how unexplored the space of DRL applications to DC sampling is, there exists plenty of directions future research could head. Future research can include work on both the practical and theoretical aspect of these frameworks.

The simulations in this thesis are accurate according to the historical data, however one aspect of a real market that is difficult to replicate retrospectively is slippage. Slippage refers to the difference in the price at which a buy or sell order is made and the price it is executed. When trading at high frequencies this can become a problem because a trade order could be posted but slippage could cause an unexpected change in entry price, by entering a trade the agent itself can affect the liquidity of the market, impacting the price level. The simulations run in this thesis assume no slippage and no effect of the agent in the market, so developing

a system that is tested in real time would provide more insight into the real word profitability of the agents.

These frameworks have been developed specifically for foreign exchange data but could be applicable to many other asset classes. Commodities and cryptocurrencies have similarities in their price movements to currency pairs, especially in the case of cryptocurrencies where there are fewer fundamental factors at play. Fundamentals were not factored into the design of FDRL, PADRL and SADRL due to the high frequencies at which they trade but with the discovery that SADRL prefers to trade over longer periods, fundamentals could skew the performance of the agents. For asset classes like cryptocurrencies where these fundamentals are much less important the SADRL framework could perform particularly well.

The reinforcement learning approach itself has plenty of room for future research. The action space, environment and reward function were all the result of chasing the highest yielding setups on a subset of the data pool on which the final models were tested. This means that there is plenty of opportunity to test out different action spaces, perhaps involving a continuous action space that control position size as well as the direction of the trade. The environment could be altered by introducing more technical indicators or positional variables and the reward function could be modified to reflect risk adjusted return to limit risk of the model as the current approach is designed to incentivise frequent trading, a feature that may not be required with alterations to the environment or action space.

Eight different DC thresholds were used in this work to provide a range of high frequency samples that would produce different outlooks on the same data. Since these DC thresholds are applied to the same data, it is possible to get two different outlooks on the the same data at any given point. This opens up the possibility of using features from the same underlying data sampled at multiple thresholds to train the agents to provide a longer and shorter term outlook, with a larger and smaller DC threshold value. Ensembling is also an effective approach to enhancing

existing algorithms as mentioned in the literature review. By introducing a voting mechanism it is possible to implement a number of different models that vote on trading actions at each time step, this has been shown to be an effective approach in the past.

As well as ensembling, a hierarchical approach could also be implemented that trains a root DRL agent to implement different DRL agents during different regimes. It was observed with SADRL that agents trained in up trending environments tend to perform well in uptrends in the test set, but applying this same agent to a downtrend did not experience the same success. If multiple agents were to be trained for a number of different environments and then controlled by a separate agent, model or technical indicator that determines the appropriate trend or agent to apply to the current market then this could build a powerful system of agents, each of which is a specialist in a given market trend.

While this thesis demonstrates the effectiveness of combining DC sampling with deep reinforcement learning, the inherent scale-invariant properties of directional change, the scaling laws extensively documented in the literature, remain largely unexploited by the learning algorithms themselves. The scaling laws identified in [5] and subsequent research, including the well established relationship $2DC \approx OS$, represent additional structural information about market behaviour that could enhance strategy performance beyond what reinforcement learning alone achieves. The current frameworks treat DC sampling primarily as a data preparation mechanism, with RL agents learning policies without explicit awareness of underlying scale-invariant relationships. Future research could integrate this knowledge through several approaches: hybrid architectures combining RL with multi-objective learning that optimises both profit and adherence to scaling relationships, extending the state space to include features derived from scaling law deviations, such as the ratio of observed OS magnitude to expected value and exploiting scale-invariance for transfer learning across thresholds and currency pairs, potentially reducing the need to train separate models for each

configuration. Such enhancements might address limitations identified in SADRL, particularly if scaling law deviations could signal trend exhaustion or reversals.

The findings in Section 4.5, showing that currency pairs with lower OS proportions generate substantially higher returns, suggest that when markets conform more closely to idealised scale-invariant patterns, strategies perform more reliably. Future work could systematically investigate how the strength of various scaling laws correlates with trading performance, enabling dynamic threshold selection where agents adapt based on real-time conformity to scaling relationships. However, implementing these approaches requires careful design to use scale-invariant properties as soft constraints rather than rigid rules, allowing agents to learn when deviations represent noise versus profitable opportunities. Empirical validation of how well scaling laws hold across the specific datasets used in this thesis would be essential, as results suggest these relationships vary in strength across market regimes. By integrating scale-invariant DC properties with deep reinforcement learning, future research could achieve strategies that are more profitable, theoretically grounded, and better able to generalise, moving from demonstrating that DC sampling and DRL combine effectively, to exploiting the deeper mathematical structure underlying DC sampling.

References

[1] R. Kissell, *Algorithmic trading methods: Applications using advanced statistics, optimization, and machine learning techniques*. Academic Press, 2020.

[2] Z. Hu, Y. Zhao, and M. Khushi, "A survey of forex and stock price prediction using deep learning," *Applied System Innovation*, vol. 4, no. 1, p. 9, 2021.

[3] S. P. Chatzis, V. Siakoulis, A. Petropoulos, E. Stavroulakis, and N. Vlachogiannakis, "Forecasting stock market crisis events using deep and statistical machine learning techniques," *Expert systems with applications*, vol. 112, pp. 353–371, 2018.

[4] Y. Tang, Z. Song, Y. Zhu, H. Yuan, M. Hou, J. Ji, C. Tang, and J. Li, "A survey on machine learning models for financial time series forecasting," *Neurocomputing*, vol. 512, pp. 363–380, 2022.

[5] D. M. Guillaume, M. M. Dacorogna, R. R. Davé, U. A. Müller, R. B. Olsen, and O. V. Pictet, "From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets," *Finance and stochastics*, vol. 1, no. 2, pp. 95–129, 1997.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[7] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[8] G. Rayment, M. Kampouridis, and A. Adegboye, "Predicting directional change reversal points with machine learning regression models," in *IEEE Symposium on Computational Intelligence for Financial Engineering & Risk (CIFEr)*, 2023.

[9] G. Rayment and M. Kampouridis, "High frequency trading with deep reinforcement learning agents under a directional changes sampling framework," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2023, pp. 387–394.

[10] Rayment, George and Kampouridis, Michael, "Enhancing high-frequency trading with deep reinforcement learning using advanced positional awareness under a directional changes paradigm," *IEEE Xplore*, 2024.

[11] S. Patro, P. P. Sahoo, I. Panda, and K. K. Sahu, "Technical analysis on financial forecasting," *arXiv preprint arXiv:1503.03011*, 2015.

[12] J. J. Groen, R. Paap, and F. Ravazzolo, "Real-time inflation forecasting in a changing world," *Journal of Business & Economic Statistics*, vol. 31, no. 1, pp. 29–44, 2013.

[13] L. Monteforte and G. Moretti, "Real-time forecasts of inflation: The role of financial variables," *Journal of Forecasting*, vol. 32, no. 1, pp. 51–61, 2013.

[14] P. Mondal, L. Shit, and S. Goswami, "Study of effectiveness of time series modeling (arima) in forecasting stock prices," *International Journal of Computer Science, Engineering and Applications*, vol. 4, no. 2, p. 13, 2014.

[15] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data," *PloS one*, vol. 14, no. 2, p. e0212320, 2019.

[16] M. Ullah, M. Shaikh, P. Channar, and S. Shaikh, "Financial forecasting: an individual perspective," *International Journal of Management (IJM)*, vol. 12, no. 3, pp. 60–69, 2021.

[17] J. Z. Berman, A. T. Tran, J. G. Lynch Jr, and G. Zauberman, "Expense neglect in forecasting personal finances," *Journal of Marketing Research*, vol. 53, no. 4, pp. 535–550, 2016.

[18] E. F. Fama, "Efficient capital markets," *Journal of finance*, vol. 25, no. 2, pp. 383–417, 1970.

[19] S. J. Grossman and J. E. Stiglitz, "On the impossibility of informationally efficient markets," *The American economic review*, vol. 70, no. 3, pp. 393–408, 1980.

[20] J.-J. Laffont and E. S. Maskin, "The efficient market hypothesis and insider trading on the stock market," *Journal of Political Economy*, vol. 98, no. 1, pp. 70–93, 1990.

[21] E. J. Wilson and H. A. Marashdeh, "Are co-integrated stock prices consistent with the efficient market hypothesis?" *Economic record*, vol. 83, pp. S87–S93, 2007.

[22] M. Sewell, "History of the efficient market hypothesis," *Rn*, vol. 11, no. 04, p. 04, 2011.

[23] J. Prakash, "A study of weak, semi-strong and strong forms of market efficiency: review of literature," *Journal of global research & analysis*, vol. 1, p. 98, 2012.

[24] R. Dias, P. Heliodoro, N. Teixeira, and T. Godinho, "Testing the weak form of efficient market hypothesis: Empirical evidence from equity markets," *International Journal of Accounting, Finance and Risk Management*, vol. 5, no. 1, p. 40, 2020.

[25] B. M. Hussin, A. D. Ahmed, and T. C. Ying, "Semi-strong form efficiency: Market reaction to dividend and earnings announcements in malaysian stock exchange." *IUP Journal of Applied Finance*, vol. 16, no. 5, 2010.

[26] T. Potocki and T. Swist, "Empirical test of the strong form efficiency of the warsaw stock exchange: the analysis of wig 20 index shares." *South-Eastern Europe Journal of Economics*, vol. 10, no. 2, 2012.

[27] I. Choi, "Testing the random walk hypothesis for real exchange rates," *Journal of Applied Econometrics*, vol. 14, no. 3, pp. 293–308, 1999.

[28] A. W. Lo and A. C. MacKinlay, "Stock market prices do not follow random walks: Evidence from a simple specification test," *The review of financial studies*, vol. 1, no. 1, pp. 41–66, 1988.

[29] A. Geromichalos and K. M. Jung, "An over-the-counter approach to the forex market," *International economic review*, vol. 59, no. 2, pp. 859–905, 2018.

[30] K. B. Pratt, "Locating patterns in discrete time-series," Master's thesis, University of South Florida, 2001.

[31] J. Yin, Y.-W. Si, and Z. Gong, "Financial time series segmentation based on turning points," in *Proceedings 2011 international conference on system science and engineering*. IEEE, 2011, pp. 394–399.

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

[33] M. O. Özorhan, İ. H. Toroslu, and O. T. Şehitoğlu, "Short-term trend prediction in financial time series data," *Knowledge and Information Systems*, vol. 61, pp. 397–429, 2019.

[34] E. P. Tsang, R. Tao, and S. Ma, "Profiling financial market dynamics under directional changes," *Quantitative finance, http://www. tandfonline. com/-doi/abs/10.1080/14697688.2016*, vol. 1164887, 2015.

[35] E. P. Tsang, R. Tao, A. Serguieva, and S. Ma, "Profiling high-frequency equity price movements in directional changes," *Quantitative finance*, vol. 17, no. 2, pp. 217–225, 2017.

[36] E. Tsang and J. Chen, "Regime change detection using directional change indicators in the foreign exchange market to chart brexit," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 3, pp. 185–193, 2018.

[37] J. B. Glattfelder, A. Dupuis, and R. B. Olsen, "Patterns in high-frequency fx data: discovery of 12 empirical scaling laws," *Quantitative Finance*, vol. 11, no. 4, pp. 599–614, 2011.

[38] M. Aloud, M. Fasli, E. Tsang, A. Dupuis, and R. Olsen, "Stylized facts of trading activity in the high frequency fx market: An empirical study," *Journal of Finance and Investment Analysis*, vol. 2, no. 4, pp. 145–183, 2013.

[39] M. E. Aloud, "Time series analysis indicators under directional changes: The case of saudi stock market," *International Journal of Economics and Financial Issues*, vol. 6, no. 1, pp. 55–64, 2016.

[40] H. Ao and E. Tsang, "Trading algorithms built with directional changes," in *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*. IEEE, 2019, pp. 1–7.

[41] A. Dupuis and R. B. Olsen, "High frequency finance: using scaling laws to build trading models," *Handbook of exchange rates*, pp. 563–584, 2012.

[42] M. Aloud, "Directional-change event trading strategy: Profit-maximizing learning strategy," in *Proceedings of the Seventh International Conference*

on Advanced Cognitive Technologies and Applications, Nice, France, vol. 22, 2015.

[43] A. Bakhach, E. P. Tsang, and H. Jalalian, "Forecasting directional changes in the fx markets," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016, pp. 1–8.

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

[45] N. Alkhamees and M. Fasli, "Event detection from time-series streams using directional change and dynamic thresholds," in *2017 IEEE international conference on big data (Big Data)*. IEEE, 2017, pp. 1882–1891.

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

[47] A. M. Bakhach, E. P. Tsang, and V. Raju Chinthalapati, "Tsfdc: A trading strategy based on forecasting directional change," *Intelligent Systems in Accounting, Finance and Management*, vol. 25, no. 3, pp. 105–123, 2018.

[48] A. Bakhach, V. L. R. Chinthalapati, E. P. Tsang, and A. R. El Sayed, "Intelligent dynamic backlash agent: A trading strategy based on the directional change framework," *Algorithms*, vol. 11, no. 11, p. 171, 2018.

[49] A. Golub, J. B. Glattfelder, and R. B. Olsen, "The alpha engine: Designing an automated trading algorithm," in *High-Performance Computing in Finance*. Chapman and Hall/CRC, 2018, pp. 49–76.

[50] M. P. Taylor and H. Allen, "The use of technical analysis in the foreign exchange market," *Journal of international Money and Finance*, vol. 11, no. 3, pp. 304–314, 1992.

[51] S. Nison, *Beyond candlesticks: New Japanese charting techniques revealed*. John Wiley & Sons, 1994, vol. 56.

[52] G. L. Morris and R. Litchfield, "Candlestick charting explained: Timeless techniques for trading stocks and futures," *(No Title)*, 1995.

[53] S. Wang, Z.-Q. Jiang, S.-P. Li, and W.-X. Zhou, "Testing the performance of technical trading rules in the chinese markets based on superior predictive test," *Physica A: Statistical Mechanics and its Applications*, vol. 439, pp. 114–123, 2015.

[54] C. Rao, "Multiple linear regression analysis," *Multivariate Statistics Made Simple: A Practical Approach*, p. 115, 2018.

[55] C. Dismuke and R. Lindrooth, "Ordinary least squares," *Methods and designs for outcomes research*, vol. 93, no. 1, pp. 93–104, 2006.

[56] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[57] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.

[58] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for l1 regularization: A comparative study and two new approaches," in *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings 18*. Springer, 2007, pp. 286–297.

[59] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," *arXiv preprint arXiv:1205.2653*, 2012.

[60] P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford, "Accelerating stochastic gradient descent for least squares regression," in *Conference On Learning Theory*. PMLR, 2018, pp. 545–604.

[61] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade: Second Edition*. Springer, 2012, pp. 421–436.

[62] V. Vovk, "Kernel ridge regression," in *Empirical inference: Festschrift in honor of vladimir n. vapnik*. Springer, 2013, pp. 105–116.

[63] B. Schölkopf, "The kernel trick for distances," *Advances in neural information processing systems*, vol. 13, 2000.

[64] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine learning*. Elsevier, 2020, pp. 101–121.

[65] M. Awad, R. Khanna, M. Awad, and R. Khanna, "Support vector regression," *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, pp. 67–80, 2015.

[66] Y.-Y. Song and L. Ying, "Decision tree methods: applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[67] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[68] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2021.

[69] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, no. 7, pp. 579–588, 2009.

[70] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation, parallel distributed processing, explorations

in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986," *Biometrika*, vol. 71, no. 599-607, p. 6, 1986.

[71] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[73] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[74] G. Philipp, D. Song, and J. G. Carbonell, "The exploding gradient problem demystified-definition, prevalence, impact, origin, tradeoffs, and solutions," *arXiv preprint arXiv:1712.05577*, 2017.

[75] M. Liu, Z. Zhuang, Y. Lei, and C. Liao, "A communication-efficient distributed gradient clipping algorithm for training deep neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 204–26 217, 2022.

[76] S. Merity, B. McCann, and R. Socher, "Revisiting activation regularization for language rnns," *arXiv preprint arXiv:1708.01009*, 2017.

[77] Q. V. Le, N. Jaitly, and G. E. Hinton, "A simple way to initialize recurrent networks of rectified linear units," *arXiv preprint arXiv:1504.00941*, 2015.

[78] Y. Wang and F. Tian, "Recurrent residual learning for sequence classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 938–943.

[79] S. Linnainmaa, "The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors," Ph.D. dissertation, Master's Thesis (in Finnish), Univ. Helsinki, 1970.

[80] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in neural information processing systems*, vol. 2, 1989.

[81] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Networks*, vol. 138, pp. 14–32, 2021.

[82] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[83] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[84] M. E. Harmon and S. S. Harmon, "Reinforcement learning: A tutorial," *WL/AAFC, WPAFB Ohio*, vol. 45433, pp. 237–285, 1996.

[85] M. L. Puterman, "Markov decision processes," *Handbooks in operations research and management science*, vol. 2, pp. 331–434, 1990.

[86] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.

[87] R. Bellman, "Dynamic programming and stochastic control processes," *Information and control*, vol. 1, no. 3, pp. 228–239, 1958.

[88] R. A. Howard, "Dynamic programming and markov processes." 1960.

[89] W. B. Powell, "Approximate dynamic programming: lessons from the field," in *2008 Winter Simulation Conference*. IEEE, 2008, pp. 205–214.

[90] T. Degris, P. M. Pilarski, and R. S. Sutton, "Model-free reinforcement learning with continuous action in practice," in *2012 American control conference (ACC)*. IEEE, 2012, pp. 2177–2182.

[91] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[92] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[93] J. F. Hernandez-Garcia and R. S. Sutton, "Understanding multi-step deep reinforcement learning: A systematic study of the dqn target," *arXiv preprint arXiv:1901.07510*, 2019.

[94] V. Mnih, "Asynchronous methods for deep reinforcement learning," *arXiv preprint arXiv:1602.01783*, 2016.

[95] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, "Bridging the gap between value and policy based reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.

[96] J. Schulman, "Trust region policy optimization," *arXiv preprint arXiv:1502.05477*, 2015.

[97] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[98] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[99] S. S. Roy, D. Mittal, A. Basu, and A. Abraham, "Stock market forecasting using lasso linear regression model," in *Afro-European Conference for Industrial Advancement: Proceedings of the First International Afro-European Conference for Industrial Advancement AECIA 2014*. Springer, 2015, pp. 371–381.

[100] J. Liu, Y. Tian, and Q. Yan, "Modelling and forecasting of commodity trading price," in *Journal of Physics: Conference Series*, vol. 1060, no. 1. IOP Publishing, 2018, p. 012075.

[101] R. Karim, M. K. Alam, and M. R. Hossain, "Stock market analysis using linear regression and decision tree regression," in *2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)*. IEEE, 2021, pp. 1–6.

[102] S. Pang, L. Song, and N. Kasabov, "Correlation-aided support vector regression for forex time series prediction," *Neural Computing and Applications*, vol. 20, pp. 1193–1203, 2011.

[103] C. Stasinakis, G. Sermpinis, I. Psaradellis, and T. Verousis, "Krill-herd support vector regression and heterogeneous autoregressive leverage: evidence from forecasting and trading commodities," *Quantitative Finance*, vol. 16, no. 12, pp. 1901–1915, 2016.

[104] R. A. Kamble, "Short and long term stock trend prediction using decision tree," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2017, pp. 1371–1375.

[105] S. Islam, A. Sholahuddin, and A. Abdullah, "Extreme gradient boosting (xgboost) method in making forecasting application and analysis of usd exchange rates against rupiah," in *Journal of Physics: Conference Series*, vol. 1722, no. 1. IOP Publishing, 2021, p. 012016.

[106] C. G. Rojas and M. Herman, "Foreign exchange forecasting via machine learning," *Obtenido de http://cs229. stanford. edu/proj2018/report/76. pdf*, 2018.

[107] T. Mahmud, T. Akter, S. Anwar, M. T. Aziz, M. S. Hossain, and K. Andersson, "Predictive modeling in forex trading: A time series analysis approach," in *2024 Second International Conference on Inventive Computing and Informatics (ICICI)*. IEEE, 2024, pp. 390–397.

[108] M. El Mahjouby, M. T. Bennani, M. Lamrini, M. El Far, B. Bossoufi, and T. A. Alghamdi, "Machine learning algorithms for forecasting and categorizing euro-to-dollar exchange rates," *IEEE Access*, 2024.

[109] S. Bhange, K. Vidya, and S. Naik, "Predicting foreign exchange rates using machine learning techniques," in *International Conference on ICT for Sustainable Development*. Springer, 2023, pp. 493–508.

[110] N. Gurung, M. R. Hasan, M. S. Gazi, and M. Z. Islam, "Algorithmic trading strategies: Leveraging machine learning models for enhanced performance in the us stock market," *Journal of Business and Management Studies*, vol. 6, no. 2, pp. 132–143, 2024.

[111] M. T. Adesina, S. D. Esebre, A. T. Adewuyi, M. Yussuf, O. A. Adigun, T. D. Olajide, C. I. Michael, and D. ILOH, "Algorithmic trading and machine learning: Advanced techniques for market prediction and strategy development," *World Journal of Advanced Research and Reviews*, vol. 23, no. 2, pp. 979–990, 2024.

[112] N. Sukma and C. S. Namahoot, "An algorithmic trading approach merging machine learning with multi-indicator strategies for optimal performance," *IEEE Access*, 2024.

[113] E. Saberi, J. Pirgazi, and A. Ghanbari sorkhi, "A machine learning approach for trading in financial markets using dynamic threshold breakout labeling," *The Journal of Supercomputing*, vol. 80, no. 17, pp. 25 188–25 221, 2024.

[114] A. Poptani, "Enhancing intraday trading through machine learning: A nse nifty 50 analysis," in *2024 5th International Conference on Innovative Trends in Information Technology (ICITIIT)*. IEEE, 2024, pp. 1–6.

[115] J. Gypteau, F. E. Otero, and M. Kampouridis, "Generating directional change based trading strategies with genetic programming," in *European*

Conference on the Applications of Evolutionary Computation. Springer, 2015, pp. 267–278.

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

[117] M. Kampouridis and F. E. Otero, "Evolving trading strategies using directional changes," *Expert Systems with Applications*, vol. 73, pp. 145–160, 2017.

[118] A. Adegboye and M. Kampouridis, "Machine learning classification and regression models for predicting directional changes trend reversal in fx markets," *Expert Systems with Applications*, vol. 173, p. 114645, 2021.

[119] A. Adegboye, M. Kampouridis, and F. Otero, "Improving trend reversal estimation in forex markets under a directional changes paradigm with classification algorithms," *International Journal of Intelligent Systems*, vol. 36, no. 12, pp. 7609–7640, 2021.

[120] ——, "Algorithmic trading with directional changes," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5619–5644, 2023.

[121] O. Salman, M. Kampouridis, and D. Jarchi, "Trading strategies optimization by genetic algorithm under the directional changes paradigm," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 1–8.

[122] O. Salman, T. Melissourgos, and M. Kampouridis, "Optimization of trading strategies using a genetic algorithm under the directional changes paradigm with multiple thresholds," *IEEE XPlore*, 2023.

[123] X. Long, M. Kampouridis, and D. Jarchi, "An in-depth investigation of genetic programming and nine other machine learning algorithms in a financial

forecasting problem," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 01–08.

[124] X. Long, M. Kampouridis, and P. Kanellopoulos, "Genetic programming for combining directional changes indicators in international stock markets," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2022, pp. 33–47.

[125] ——, "Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators," in *2023 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2023, pp. 1–8.

[126] X. Long and M. Kampouridis, "α-dominance two-objective optimization genetic programming for algorithmic trading under a directional changes environment," in *2024 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*. IEEE, 2024, pp. 1–8.

[127] L. K. Y. Loh, H. K. Kueh, N. J. Parikh, H. Chan, N. J. H. Ho, and M. C. H. Chua, "An ensembling architecture incorporating machine learning models and genetic algorithm optimization for forex trading," *FinTech*, vol. 1, no. 2, pp. 100–124, 2022.

[128] H. Hajimiri, "Use of genetic algorithm in algorithmic trading to optimize technical analysis in the international stock market (forex)," *Journal of Cyberspace Studies*, vol. 6, no. 1, pp. 21–29, 2022.

[129] C. Evans, K. Pappas, and F. Xhafa, "Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation," *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1249–1266, 2013.

[130] S. Ahmed, S.-U. Hassan, N. R. Aljohani, and R. Nawaz, "Flf-lstm: A novel prediction system using forex loss function," *Applied Soft Computing*, vol. 97, p. 106780, 2020.

[131] S. Galeshchuk and S. Mukherjee, "Deep learning for predictions in emerging currency markets," in *International conference on agents and artificial intelligence*, vol. 2. SCITEPRESS, 2017, pp. 681–686.

[132] M. S. Islam and E. Hossain, "Foreign exchange currency rate prediction using a gru-lstm hybrid network," *Soft Computing Letters*, vol. 3, p. 100009, 2021.

[133] A. J. Dautel, W. K. Härdle, S. Lessmann, and H.-V. Seow, "Forex exchange rate forecasting using deep recurrent neural networks," *Digital Finance*, vol. 2, pp. 69–96, 2020.

[134] A. Nemavhola, C. Chibaya, and N. M. Ochara, "Application of the lstm-deep neural networks-in forecasting foreign currency exchange rates," in *2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*. IEEE, 2021, pp. 1–6.

[135] M. Ayitey Junior, P. Appiahene, and O. Appiah, "Forex market forecasting with two-layer stacked long short-term memory neural network (lstm) and correlation analysis," *Journal of Electrical Systems and Information Technology*, vol. 9, no. 1, p. 14, 2022.

[136] S. Lahmiri and S. Bekiros, "Deep learning forecasting in cryptocurrency high-frequency trading," *Cognitive Computation*, vol. 13, pp. 485–487, 2021.

[137] M.-C. Hung, A.-P. Chen, and W.-T. Yu, "Ai-driven intraday trading: Applying machine learning and market activity for enhanced decision support in financial markets," *IEEE Access*, 2024.

[138] P. D. Nguyen, N. N. Thao, D. T. Kim Chi, H.-C. Nguyen, B.-N. Mach, and T. Q. Nguyen, "Deep learning-based predictive models for forex market trends: Practical implementation and performance evaluation," *Science Progress*, vol. 107, no. 3, p. 00368504241275370, 2024.

[139] S. Yadav, A. Singh, S. K. Singh, V. P. Singh, V. A. Vuyyuru, and A. Balakumar, "Improving market efficiency and profitability in high-frequency trading

using neural network-based deep learning techniques," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2024, pp. 1–6.

[140] K. T. Kantoutsis, A. N. Mavrogianni, and N. P. Theodorakatos, "Transformers in high-frequency trading," in *Journal of Physics: Conference Series*, vol. 2701, no. 1. IOP Publishing, 2024, p. 012134.

[141] J.-H. Liou, Y.-T. Liu, and L.-C. Cheng, "Price spread prediction in high-frequency pairs trading using deep learning architectures," *International Review of Financial Analysis*, vol. 96, p. 103793, 2024.

[142] T. L. Meng and M. Khushi, "Reinforcement learning in financial markets," *Data*, vol. 4, no. 3, p. 110, 2019.

[143] J. Moody and M. Saffell, "Reinforcement learning for trading," *Advances in Neural Information Processing Systems*, vol. 11, 1998.

[144] Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," *IEEE Access*, vol. 7, pp. 108 014–108 022, 2019.

[145] J. Carapuço, R. Neves, and N. Horta, "Reinforcement learning applied to forex trading," *Applied Soft Computing*, vol. 73, pp. 783–794, 2018.

[146] F. Rundo, "Deep lstm with reinforcement learning layer for financial trend prediction in fx high frequency trading systems," *Applied Sciences*, vol. 9, no. 20, p. 4460, 2019.

[147] Y.-C. Tsai, C.-C. Wang, F.-M. Szu, and K.-J. Wang, "Deep reinforcement learning for foreign exchange trading," in *Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices: 33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2020, Kitakyushu, Japan, September 22-25, 2020, Proceedings 33*. Springer, 2020, pp. 387–392.

[148] S. Lele, K. Gangar, H. Daftary, and D. Dharkar, "Stock market trading agent using on-policy reinforcement learning algorithms," *Available at SSRN 3582014*, 2020.

[149] J. B. Chakole, M. S. Kolhe, G. D. Mahapurush, A. Yadav, and M. P. Kurhekar, "A q-learning agent for automated trading in equity stock markets," *Expert Systems with Applications*, vol. 163, p. 113761, 2021.

[150] S. Lin and P. A. Beling, "An end-to-end optimal trade execution framework based on proximal policy optimization," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 4548–4554.

[151] M. E. Aloud and N. Alkhamees, "Intelligent algorithmic trading strategy using reinforcement learning and directional change," *IEEE Access*, vol. 9, pp. 114 659–114 671, 2021.

[152] N. Alkhamees and M. Aloud, "Dcrl: Approach for pattern recognition in price time series using directional change and reinforcement learning," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, 2021.

[153] S. Sarkar, "Harnessing deep q-learning for enhanced statistical arbitrage in high-frequency trading: A comprehensive exploration," *arXiv preprint arXiv:2311.10718*, 2023.

[154] V. Arangi, S. J. S. Krishna, K. Santosh, S. Paliwal, B. Abdurasul, and I. I. Raj, "Reinforcement learning-optimized trading strategies: A deep q-network approach for high-frequency finance," in *2024 International Conference on Data Science and Network Security (ICDSNS)*. IEEE, 2024, pp. 1–6.

[155] M. Massahi and M. Mahootchi, "A deep q-learning based algorithmic trading system for commodity futures markets," *Expert Systems with Applications*, vol. 237, p. 121711, 2024.

[156] G. Cao, Y. Zhang, Q. Lou, and G. Wang, "Optimization of high-frequency trading strategies using deep reinforcement learning," *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, vol. 6, no. 1, pp. 230–257, 2024.

[157] K. Farooghi and H. Khaloozadeh, "Cooperative multi-agent deep reinforcement learning for forex algorithmic trading using proximal policy optimization," in *2024 10th International Conference on Control, Instrumentation and Automation (ICCIA)*. IEEE, 2024, pp. 1–6.

[158] C. Zong, C. Wang, M. Qin, L. Feng, X. Wang, and B. An, "Macrohft: Memory augmented context-aware reinforcement learning on high frequency trading," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4712–4721.

[159] M. Qin, S. Sun, W. Zhang, H. Xia, X. Wang, and B. An, "Earnhft: Efficient hierarchical reinforcement learning for high frequency trading," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 13, 2024, pp. 14 669–14 676.

[160] A. Kumar, R. Rizk, and K. Santosh, "Transformer-based reinforcement learning model for optimized quantitative trading," in *2024 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2024, pp. 1454–1455.

[161] Y. Ansari, S. Gillani, M. Bukhari, B. Lee, M. Maqsood, and S. Rho, "A multifaceted approach to stock market trading using reinforcement learning," *IEEE Access*, 2024.

[162] R. Vetrina and K. Kobergb, "Reinforcement learning in optimisation of financial market trading strategy parameters," *COMPUTER*, vol. 16, no. 7, pp. 1793–1812, 2024.

[163] S. Sun, M. Qin, W. Zhang, H. Xia, C. Zong, J. Ying, Y. Xie, L. Zhao, X. Wang, and B. An, "Trademaster: a holistic quantitative trading platform empowered by reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[164] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[165] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html