Novel Trading Algorithms augmented by Intrinsic Time and Machine Learning

By

QI ZHAO

A thesis submitted for the degree of

Doctor of Philosophy

Centre for Computational Finance and Economic Agents (CCFEA)

University of Essex

Abstract

The Directional Change (DC) Framework is an innovative event-driven approach designed to characterize price movements in micro financial markets using uneven time intervals as a system clock. This research investigates novel techniques for market price prediction and trading strategies based on the DC framework. We developed two distinct DC-based strategies: trend-following and counter-trend strategies. Empirical evidence highlights that the trend-following strategy offers higher potential profitability, albeit with relatively higher risk compared to the counter-trend strategy. To enhance the profitability of DC-based strategies, we integrated machine learning (ML) algorithms to predict the type and length of overshoot (OS) events. We employed five prominent ML models—Random Forest, Support Vector Machine (SVM), Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN). Additionally, we incorporated a reinforcement learning (RL) framework to further improve the performance of the DC-based trading strategies. Our DCRL strategy, combined with a deep neural network policy network (ResNet), demonstrated exceptional ability in selecting optimal strategies under natural market states.

The study utilized minute-level data from the foreign exchange market, focusing on eight major currency pairs over a 15-year period from 2006 to 2020. The performance of the prediction models was evaluated using metrics such as Rate of Return (ROR), Maximum Drawdown (MDD), and Sharpe Ratio.

However, the study has certain limitations: the dataset is confined to the foreign exchange market, which may affect the representativeness and generalizability of the findings. The time span may not be sufficient to support long-term trend analysis or the impact of specific events. Additionally, large-scale data analysis and complex models require high computational power, potentially posing technical and resource constraints.

Future research directions could include developing specialized financial data network models, incorporating multi-agent reinforcement learning, expanding datasets and asset classes,

and further exploring the DC framework.

This study advances the integration of machine learning algorithms within the DC framework to improve trading strategy profitability and introduces multi-core processing to significantly enhance the learning efficiency of RL models, a substantial advancement that can greatly accelerate future research in this domain.

Acknowledgment

I would like to express my heartfelt gratitude to my supervisors for their unwavering support and invaluable guidance throughout the entire research journey. Their expertise, encouragement, and dedicated assistance have played a pivotal role in shaping this dissertation.

My deepest appreciation goes to my family and friends, whose unwavering support and understanding have been a constant source of strength. Their encouragement and belief in my abilities have been instrumental in reaching this academic milestone.

During the pursuit of my doctoral degree, the world faced the challenges posed by the global COVID-19 pandemic. I am profoundly thankful to all those who extended their support during these trying times. The collective efforts and assistance received from colleagues, peers, and various individuals have been indispensable in overcoming the difficulties posed by the pandemic.

This academic endeavor would not have been possible without the encouragement, understanding, and assistance provided by my mentors, loved ones, and the supportive community around me. I am sincerely grateful for the collaborative spirit and resilience demonstrated by all those who contributed to my academic and personal growth.

Contents

Abstract iii

Acknowledgement v

List of Figures xi

List of Tables xv

Notation xvii

1 Introduction 1

1.1 Methodology and Contributions 3

1.2 Thesis Structure 6

2 Literature Review 9

2.1 Trading Strategies for Financial Markets 10

2.1.1 The FX Market 10

2.1.2 Essential Terminologies of Trading Strategies and FX Market . . . 11

2.1.3 Common Trading Strategies 13

2.1.4 Evaluating the performance 16

2.2 Directional Change 17

2.2.1 Intrinsic Time 18

2.2.2 Directional Change Framework Construction Method 19

2.2.3 The Scaling Laws of the Directional Change Framework 26

2.2.4 The DC-based Trading strategies 33

2.3 Machine Learning Techniques 36

2.3.1 Random Forest . . . 38
2.3.2 Support Vector Machines . . . 40
2.3.3 Artificial Neural Networks . . . 41
2.3.4 Long Short-Term Memory . . . 43
2.3.5 Convolutional Neural Networks . . . 44
2.3.6 Residual Convolutional Neural Network . . . 46
2.4 Reinforcement Learning . . . 48
2.4.1 Basic Principles and Components of Reinforcement Learning . . . 48
2.4.2 Reinforcement Learning in Financial Markets and Trading . . . 50
2.5 Research Dataset . . . 52

3 Trend-following vs Counter-trending: a comparative analysis 55
3.1 Introduction . . . 56
3.2 Related Work . . . 57
3.2.1 The Directional Change framework: A brief overview . . . 58
3.2.2 Scaling laws . . . 59
3.2.3 The Directional Change based trading strategies . . . 61
3.2.4 The Simple DC-TF Trading Strategy . . . 65
3.2.5 The Simple DC-CT Trading Strategy . . . 71
3.2.6 DC-based Dynamic Stop-Loss . . . 73
3.3 Result and Analysis . . . 76
3.3.1 Research Data . . . 77
3.3.2 Evaluation Indicators . . . 78
3.3.3 Result of Two Opposite Trading Strategy Comparison . . . 80
3.3.4 Statistical Analysis of DC-based Strategy Performance . . . 84
3.3.5 Result of the DC-CT and DC-TF Strategies with the DC-based Dynamic Stop-Loss . . . 87
3.4 Conclusions . . . 92

4 Machine Learning to Manage Risk in the DC-based Trend-following Strategies 95
4.1 Introduction . . . 96
4.2 Directional Change Framework and the DC-based Trading strategies . . . 97
4.2.1 Directional Change Framework . . . 97

4.2.2 Scaling Laws 98

4.2.3 Simple Trend-following DC-based Trading Strategy 99

4.3 FOS: A Trading Strategy based on Forecasting Overshoot of the DC Framework 100

4.3.1 Machine Learning Algorithms 100

4.3.2 The Forecasting OS Event Two-classification Model Based on Machine Learning Algorithms 104

4.4 Experiments 108

4.4.1 Research Data 108

4.4.2 Hyperparameters and Features 108

4.4.3 Result and Analysis 112

4.5 Conclusion 115

5 Reinforcement Learning for a Novel Trading Algorithm 117

5.1 Introduction 117

5.2 Problem Definition 120

5.2.1 Mathematical Formalism of Trading Algorithm 121

5.2.2 Transaction Cost 123

5.3 The Simple DC-based Trend-following Strategy 124

5.4 Reinforcement Learning with Directional Change 126

5.4.1 Environment and State 127

5.4.2 Reward Function 130

5.4.3 Training Procedure 131

5.4.4 Policy Iteration 132

5.4.5 Policy Network 134

5.4.6 Experience Management 135

5.4.7 CPU Parallel Technology 138

5.5 Experiments 140

5.5.1 Trading Strategies Performance Evaluation 142

5.5.2 Experimental Results 143

5.6 Conclusion 147

6 Conclusions **149**

6.1 Research Summary 149

6.2 Research Contributions 150

6.3 Research Limitations 152

6.4 Future Work 153

Appendices

Bibliography **157**

List of Figures

2.1 The time series price chart for AUN/NZD. The blue line indicates AUD/NZD mid-prices sampled minute by minute. 20

2.2 An example of a DC-based summary of the price series from Figure 2.1. $\theta_{DC} = 0.5\%$. The blue line indicates AUD/NZD mid-prices sampled minute by minute. Solid red lines represent DC events, and Solid green lines represent OS events. This is to say that the market follows an event cycle of 4 events: ... $\rightarrow$ downtrend DC event $\rightarrow$ downtrend OS event $\rightarrow$ uptrend DC event $\rightarrow$uptrend OS event $\rightarrow$ downtrend DC event $\rightarrow$... 24

2.3 Intrinsic time transformations: According to the procedure defined in the main text, the original time series is decomposed into DC and Overshoot (OS) events. Two DC thresholds (0.5% and 0.3%) are used to measure intrinsic time events. In practice, intrinsic time defined by different resolution thresholds results in a transformation from a tick-by-tick time series representation to an event-based price curve, evolving according to the intrinsic time increments τ_i. Thus, the dependence on physical time is replaced by a dependence on intrinsic market activity, which is defined at the selected granularity level, revealing the underlying coastlines. 29

2.4 Statistical metrics for Directional-change and Overshoot Events. 31

2.5 Random Forest 39

2.6 ANNs 42

2.7 LSTM 43

2.8 Residul block 47

2.9 Reinforcement Learning Algorithm 50

3.1 chain of alternating DC and OS events 58
3.2 An example of a scaling law presented in [1], that shows (a) price changes in DC and followed OS is approximately same and the average length both equal to θ_{DC}, (b) the OS event lasts about the double amount of time that it took for the DC event to take place 60
3.3 The DC-based Trend-following strategy, which capitalizes on price trends using the DC framework. The strategy opens a position at the start of a confirmed DC event, following the trend direction. And expects to close the position before the next DC event occurs. 62
3.4 The DC-based Counter-trend strategy. The strategy opens a countertrend position when OSV equals 1. And expects to close the position after the next DC event has been reported. 63
3.5 Trading rules of Simple trend-following trading strategy. It shows a sample DC-TF trading strategy that opens a position at the DC Confirmation point and closes the position after reaching the predetermined profit target. 70
3.6 Trading rules of Simple Counter-trend trading strategy. It shows a sample DC-CT trading strategy that opens a position when the OS event lasts for a DC threshold length and closes the position after it reaches the predetermined profit target. 72
3.7 Normalized daily exchange rates. 78
3.8 Un-levered aggregated Profit & Loss(P&L) of the DC-based simple Counter-trend strategy and simple Trend-following strategy, across 8 currency pairs, for eight years. See details in Appendix. 83
3.9 Un-levered daily Profit and Loss of the DC-based sample trend Following trading model with the DC-based dynamic Stop-Loss. 89
3.10 Un-levered daily Profit and Loss of the DC-based sample Counter-trend trading model with the DC-based dynamic Stop-Loss. 89

4.1 Two scaling laws from [1], that shows (a) price changes in DC and followed OS is approximately same and equals to θ_{DC}, (b) if DC event takes t physical time, its subsequent OS takes approximately $2t$ 98

4.2 Trading rules of Simple trend-following trading strategy. 100
4.3 A simple example of type OS_a and OS_b. The red part is the DC events, while the black part is the OS events. 106
4.4 An example of a DC-based summary of the price series from Figure 2.1. $\theta_{DC} = 0.5\%$. OS_{BC}, OS_{FG}, OS_{HI}, OS_{NO} belong to OS_a and OS_{DE}, OS_{JK}, OS_{LM} belong to OS_b 107
4.5 Average Profit and Loss for the 'buy and hold' the original DC-based Trend-following and five ML-based strategies on the FX market from 2016 to 2020. 113

5.1 Trading rules of the Simple trend-following trading strategy. Within the figure, red arrows are indicative of the process encompassing both the opening and closing of a position. The commencement of the red arrow denotes the opening point of the position, which concurrently serves as the DC Confirmation point. Conversely, the arrow's endpoint signifies the closing point of the opened position, marking the juncture where the price has reached the predefined profit target, P_t. The orientation of the arrow illustrates the direction of the position: a downward trajectory indicates a short position, while an upward direction signals a long position. 125
5.2 Tensor state s 130
5.3 Reward update 131
5.4 ResCNN 135
5.5 the process of Gain experience and policy improvement. 137
5.6 the introduction of CPU parallel technology on policy iteration. Where N is the number of the core of the CPU. In this study, N is equal to 36 139
5.7 The performance of RL agent on FX market 141
5.8 Average Profit & Loss for our RL strategy based on the DC framework, MA strategy, FOS strategy, and the 'B&H' strategy on the FX market from 2017 to 2020 144

List of Tables

2.1 Explanation of DC-related variables 26

2.2 An example of the dataset showing minute-level Forex price data. The minute-by-minute period displayed is from 20:01 to 20:04 on January 1, 2007 for NZD/JPY 53

2.3 An example of a DC summary using USD/JPY mid-prices sampled minute-by-minute from 17:00:00 to 17:21:00 on 2012-02-05 54

3.1 The DC-based trading strategy employing trend-following or Counter-trend trading rules. 66

3.2 Annualized ROR for sample DC-CT on Various Pairs (0.01 to 0.10) 81

3.3 Annualized ROR for sample DC-CT on Various Pairs (0.11 to 0.20) 81

3.4 Annualized ROR for sample DC-TF on Various Pairs (0.01 to 0.10) 81

3.5 Annualized ROR for sample DC-TF on Various Pairs (0.11 to 0.20) 82

3.6 Comparison of trend-following and Counter-trend strategies 84

3.7 Paired t-test Results for DC-TF and DC-CT 87

3.8 DC-based dynamic stop-loss system parameters (Unit:hours 88

3.9 Comparison of DC-TF and DC-CT Strategies with Stop-loss 90

4.1 Hyperparameters and their best settings for RF, SVM, ANN, LSTM and CNN models used in this study. 103

4.2 OS length prediction features - A brief description of independent variables used for classifying OS_i events. There are a total of 13 features. Where P_{OS_i} is the start price of OS_i. 111

4.3 Trading strategies evaluation table for FX market (2017-2020) 114

5.1 Trading strategies evaluation table for FX market (2017-2020) 145

Notation

DC/OS Event Parameters

DC	Directional Change
OS	Overshoot
e	Event vector
d	The direction of event
p_s	Start price
t_s	Start time
p_e	End price
t_e	End time
l	Length of event($l = p_e - p_s$)
T	Duration of time ($T = t_e - t_s$)
E	Events set vector
P_{DCC}	DC confirmation point
θ_{DC}	Pre-determined Directional Change threshold
FOS	Forecasting Overshoot Event model
$TSFDC$	A trading strategy based on forecasting directional change
$ZI-DCT$	Zero Intelligence Directional Change Trading
BA	Bakhach Agent
DBA	Dynamic Bakhach Agent

Machine Learning/Reinforcement Learning Hyperparameters

ML	Machine learning
RF	Random Forest

SVM	Support Vector Machine
NN	Neural Network
ANN	Artificial Neural Network
$LSTM$	Long short-term memory
CNN	Convolutional Neural Network
$ResNet$	Residual Neural Network
RL	Reinforcement learning
DRL	Deep Reinforcement learning
$DDRL$	Deep Direct Reinforcement learning
DQN	Deep Q-Network
$A3C$	Asynchronous advantage actor-critic
$A2C$	Advantage Actor-Critic
$MCTS$	Monte Carlo Tree Search
GAN	Generative Adversarial Network
GA	Genetic Algorithm
GP	Genetic programming-based dynamic portfolio trading system
PSO	Particle swarm optimisation
$CSLFA$	Continuous shuffled frog algorithm
MDP	Markov Decision Process
$\kappa(x,x')$	Kernel function
$ReLU$	Rectified Linear Unit
$(f*g)(x,y)$	Convolution operation of two functions f and g
s	States (in MDP)
A	Actions
P	Transition probability
R	Reward function
π^*	Policy network
$Q^\pi(s,a)$	Action-value function
$V^\pi(s)$	State-value function

Trading Performance Evaluation Indicators

ROR	Rate of Return
MDD	Maximum Drawdown
Sr	Sharpe ratio
LOB	Limit Order Book
$FTSE100$	UK Financial Times Stock Exchange 100 Index
$S\&P500$	Standard & Poor's 500 Index
μ	Transaction costs
$B\&H$	Buy and hold
PT	Profit target
P_t	Profit target (alternative notation)
S_θ	Stopping threshold
TF	Trend Following
CT	Counter-Trend
TA	Trading algorithms
HFT	High-frequency Trading
TTR	Technical trading rules
MA	Moving Average

Others

FX	Foreign exchange market
OTC	Over-the-counter
BIS	Bank of International Settlement
Pip	Percentage in Point
CPU	Central Processing Unit
GPU	Graphics Processing Unit

Chapter 1

Introduction

Accurate forecasting and decision-making remain fundamental challenges in financial markets, where increasing market complexity and rapidly changing conditions require more effective analytical and trading methods [2–4]. Traditionally, market participants have relied on technical analysis and fundamental analysis to support trading decisions. While these approaches have proven useful in many settings, they often struggle to capture the high-dimensional and non-linear relationships that characterize modern financial markets [5].

In recent years, machine learning (ML) and reinforcement learning (RL) have emerged as promising tools for financial analysis and algorithmic trading [6, 7]. ML techniques, including support vector machines, random forests, and neural networks, have demonstrated their ability to identify complex patterns in financial data and improve forecasting performance [8,9]. RL extends this capability by enabling trading agents to learn adaptive decision-making policies through interactions with the market environment, making it particularly suitable for dynamic and uncertain financial settings [10, 11]. Despite these advances, the effectiveness of ML and RL methods depends heavily on how market information is represented and incorporated into the learning process.

This thesis adopts the Directional Change (DC) framework as an event-based representation of market dynamics. Unlike conventional approaches that analyse price movements in physical time, the DC framework focuses on significant price changes that exceed a predefined threshold, thereby capturing market activity through a sequence of directional change and overshoot events [12]. Previous studies have shown that this framework provides valuable insights into volatility estimation, market behaviour, risk management, and trading strategy development [1, 13].

In this thesis, the DC framework serves as the central modelling paradigm that underpins all stages of the research. Rather than analysing market dynamics in physical time, the DC framework represents price movements through event-based observations defined by directional change and overshoot events. This representation provides the foundation for trading strategy construction, market state characterization, and predictive model. In all, it provides a unified representation of market dynamics that supports trading strategy design, Overshoot prediction, and RL-based decision-making.

The primary objective of this study is to advance understanding of the DC framework within financial analysis and trading, investigating its applications and potential for integration with ML techniques. Specifically, this research aims to explore how the DC framework can enhance financial data analysis and to examine its role within the paradigm of RL for developing trading strategies. Through this investigation, we seek to establish theoretical insights that contribute to the development of more adaptive and effective trading strategies, validated with real-world financial data from the foreign exchange (FX) market—a domain known for its high volatility and liquidity [14]. This focus ensures that our findings are directly applicable and relevant to FX market participants, including banks, corporations, and individual traders, for whom accurate and adaptive strategies are essential.

The contributions of this study are primarily conceptual and methodological. Specifically, this thesis provides a systematic classification and analysis of trading strategies based on DC scaling laws, develops a series of trading models within the DC framework, and examines how DC can be applied both independently and in combination with ML and RL techniques. More broadly, the research establishes an integrated framework that links event-based market representation, predictive modelling, and adaptive decision-making.

Meanwhile, the empirical results should be interpreted within the scope of the experimental design. The analysis is based entirely on FX market data, a specific historical sample period, selected directional change thresholds, and specific ML and RL architectures. Therefore, while the results provide evidence of the effectiveness of the proposed method in the examined scenario, its quantitative performance may not generalize to other asset classes, market mechanisms, threshold configurations, or learning frameworks. Thus, the main contribution of this paper is to demonstrate the potential of the DC framework combined with

ML and RL as the foundation for financial analysis and trading strategy development, rather than claiming that any specific empirical implementation has universal superiority.

In the following sections, we present an overview of related work, introduce the foundational DC framework, and detail the machine learning and reinforcement learning methodologies employed in this study. We then describe the data and feature selection processes, outline our proposed trading strategy optimization techniques, and discuss the experimental design and results. Finally, we explore potential extensions and improvements to our approach, concluding with a summary of our findings and recommendations for future research directions.

This thesis is structured around three interconnected research modules that collectively form a unified framework for financial analysis and trading. The first module focuses on the identification and characterization of market events using the DC framework. It investigates trend-following and counter-trend trading strategies based on DC event structures and scaling laws. Building on this event-based representation of market dynamics, the second module examines the prediction of Overshoot (OS) magnitudes using machine learning methods. The objective is to improve the understanding and forecasting of price movements that occur after DC events. The third module extends this predictive framework to trading decisions through reinforcement learning. In this stage, DC-derived market states and OS forecasts are incorporated into an adaptive trading policy that learns from market interactions. Together, these three modules establish a logical progression from market event identification, to price movement prediction, and finally to dynamic decision-making. This progression provides an integrated framework that combines market representation, predictive modelling, and adaptive strategy optimisation within the FX market. Since the effectiveness of DC events has been repeatedly verified in the FX market, our research is currently entirely based on the FX market.

1.1 Methodology and Contributions

This dissertation delineates the development and exploration of advanced trading strategies, leveraging novel methodologies in the realm of micro-financial markets. In the ensuing discourse, we traverse three methodological terrains: DC, Machine Learning, and reinforcement learning, and their respective contributions to developing robust trading strategies.

Chapter three delves into the application of the DC framework, a novel approach specifically designed to analyze micro-level price movements in financial markets. This chapter categorizes trading strategies derived from DC-based scaling laws into Directional Change based counter-trend (DC-CT) and Directional Change based trend following (DC-TF) trading strategy, highlighting their overlaps, distinctions, and respective effectiveness under varying FX market conditions. Additionally, it explores the role of stop-loss systems in enhancing their performance, ultimately leading to the development of an improved DC-driven strategy.

Chapter four takes this exploration a step further, integrating the DC Framework with Machine Learning algorithms to devise precise and profitable trading strategies. This section dissects the process of leveraging DC-based trend-following strategies, along with five distinct ML prediction models, to anticipate volatility in the FX market. The contribution of the novel sampling method in the augmentation of trading strategy performance is detailed in this chapter.

Chapter five transitions into the domain of Reinforcement Learning, highlighting its crucial role in algorithmic trading. This narrative emphasizes the training of a value judgment agent to enhance the profitability of DC-based trend-following strategies. Using the RL framework, we design an optimal training approach that identifies lucrative trading opportunities in the FX market.

Collectively, these chapters examine a progressive sequence of research problems within the DC framework. The first investigates the design and evaluation of DC-based trend-following and counter-trend trading strategies. The second explores the predictability of OS events using machine learning models and the integration of such forecasts into trading decisions. The third studies how reinforcement learning can utilize DC-derived market states to develop adaptive trading policies. Together, these investigations establish a coherent framework that progresses from event identification, to price movement prediction, and ultimately to dynamic trading decision-making in the foreign exchange market.

1. **Trend-following vs Counter trending: a comparative analysis (Chapter 3)**

   In this chapter, we utilize an innovative DC framework to analyze micro-level financial market dynamics, that is, to use the DC framework to reflect the characteristics of financial market price changes. Based on the scaling laws of the DC framework, we develop two trading strategies—DC-TF and DC-CT—and rigorously compare their subtle similarities and differences, evaluating their profitability across an eight-year dataset of eight FX currency pairs. By exploiting DC-based scaling laws, the study provides an event-driven perspective on trading strategy design and examines the conditions under which each strategy is effective. In addition, a DC-based dynamic stop-loss mechanism is introduced and integrated into both strategies, demonstrating how DC event structures can be used to support risk management as well as trade execution. The results provide evidence that DC event structures can be used to support both trading strategy design and risk control.

2. **Machine Learning to Manage Risk in the DC-based Trend-following Strategies (Chapter 4)**
   This chapter extends the DC framework by incorporating machine learning techniques for the prediction of Overshoot (OS) magnitudes. It provides a systematic comparison of five machine learning models—Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and Convolutional Neural Network(CNN)—within a unified DC-based trading framework. By linking OS prediction to trading decisions, the chapter demonstrates how predictive modelling can be integrated with event-based market representations to support risk management and improve the effectiveness of DC-based trend-following strategies.

3. **Reinforcement Learning for a Novel Trading Algorithm (Chapter 5)**
   This chapter extends the DC framework from predictive modelling to adaptive decision-making through reinforcement learning. It proposes a trading framework in which DC-derived market states are used to represent the trading environment, enabling a reinforcement learning agent to learn trading policies directly from market interactions. By integrating event-based market representation with sequential decision-making, the chapter demonstrates the potential of reinforcement learning for developing adaptive DC-based trading strategies.

To bolster the transparency and reproducibility of our study, we have made the associated code openly available. Readers can access and review the full implementation via the provided DOI: [15].

1.2 Thesis Structure

The outline of this thesis is organized as follows: The first chapter serves as an introduction to the full thesis. The second chapter will state the background literature review of my research. All of my work's contributions are distributed in Chapter 3, Chapter 4, and Chapter 5. The last Chapter 6 presents the conclusions and future potentiality of this thesis. The details of each chapter from the next part are summarized as follows:

- Chapter 2 plays a role as the theoretical basis and source of this thesis through reviewing the related work to the research's topic. embarks on an intricate examination of the extant literature that underpins the theoretical foundation of this research endeavor. We begin with a detailed introduction to the primary arena of our investigation, the FX market. A meticulous review of prevailing trading strategies and the methodologies employed in their evaluation is subsequently undertaken. Central to our exploration is the concept of DC, from which the notion of scaling laws and DC-based trading strategies are derived. The latter part of the chapter transitions into an introduction of the foundational mainstream Machine Learning models that form the bedrock of our trading strategy architecture, including Random Forest, SVM, ANN, LSTM, and CNN. The chapter concludes with an exposition of the core concepts of Reinforcement Learning, a crucial methodology employed in the formulation of the proposed trading strategy.

- Chapter 3 presents the application of the DC framework to trading strategy development. The chapter introduces two DC-based trading strategies, namely a trend-following strategy and a counter-trend strategy, and evaluates their performance in the foreign exchange market. It then examines the incorporation of a DC-based dynamic stop-loss mechanism and assesses its impact on trading performance and risk characteristics.

- Chapter 4 This chapter extends the DC framework by incorporating machine learning techniques for the prediction of Overshoot OS magnitudes. It provides a systematic comparison of five machine learning models—Random Forest, Support Vector Machine, Artificial Neural Network, Long Short-Term Memory, and Convolutional Neural Network—within a unified DC-based trading framework. By linking OS prediction to trading decisions, the chapter demonstrates how predictive modelling can be integrated with event-based market representations to support risk management and improve the effectiveness of DC-based trend-following strategies.

- Chapter 5 investigates the application of reinforcement learning to trading within the DC framework. The chapter develops a reinforcement learning environment based on DC-derived market states and designs an agent for making trading decisions. It describes the state representation, action space, reward structure, and training procedure adopted in the learning process. Aided by the RL framework, we design an optimal training approach that unearths lucrative trading opportunities in the FX market. The resulting trading strategy is then evaluated using foreign exchange market data and compared with benchmark DC-based approaches.

- Chapter 6 draws this dissertation to a close by encapsulating the salient points of the discussion and summarizing the contributions made by the research. This final segment of the study posits potential avenues for future research, as our innovative approach forges new pathways for exploring algorithmic trading methodologies in the realm of micro-financial markets.

Chapter 2

Literature Review

Before presenting the main contributions of this thesis, it is essential to review the pertinent background literature. Algorithmic trading strategies, an integral component of financial market transactions, utilize complex computational algorithms for high-speed trade executions. These strategies circumvent human limitations, accounting for factors such as price, timing, volume, and mathematical models. One key approach in devising these strategies is the DC framework, which enables a nuanced understanding of price movements in micro financial markets by identifying significant price shifts or 'events'.

Further enhancing the accuracy and efficiency of trading strategies is the implementation of Machine Learning and Deep Learning algorithms. ML, with models like Random Forest, Support Vector Machine, and Artificial Neural Networks, provides a robust approach to market movement predictions. DL, a subset of ML, employs artificial neural networks to model high-level abstractions in data, with models like Long Short-Term Memory and Convolutional Neural Networks significantly enhancing predictive accuracy. Reinforcement Learning, another type of ML, allows an agent to make decisions to maximize a reward signal, thereby augmenting the profitability of trading strategies. Collectively, these methodologies shape the landscape of algorithmic trading, promising more precise and robust strategies in the ever-evolving micro-financial markets.

In Section 2.1, a comprehensive overview of the FX (Forex) market is provided, including an exposition of trading strategies and their evaluations. Section 2.2 then delves into the central concept of our research, the DC framework, which provides a new perspective on the intricacies of price fluctuations in micro financial markets. Section 2.3 elucidates the machine learning models incorporated in our study, offering insights into their role in enhancing prediction accuracy and trading profitability. Finally, Sections 2.4 furnishes an introduction to the foundational concepts of RL, setting the stage for its implementation in the development of trading strategies.

2.1 Trading Strategies for Financial Markets

In this chapter, we provide a brief introduction to the FX market and concept of trading strategy

2.1.1 The FX Market

The FX market, the world's largest financial market by trading volume, facilitates the trade of currencies at specified rates. A critical component in the global trade ecosystem, it supports worldwide import and export activities by determining exchange rates. The daily turnover of the FX market stood at an average of $7.5 trillion USD as of April 2022, highlighting its substantial economic impact [16].

The unique organization of the FX market as an over-the-counter (OTC) system differentiates it from centralized exchange-based markets. As a decentralized entity, the FX market lacks a unifying, visible price at any given time, unlike the singular market price evident in centralized systems. It embodies the largest example of this type of market. Trading in the FX market does not confine itself to a physical floor within specified hours, as seen in stock markets. Instead, it operates online via computer networks, linking dealers across various trading centers globally.

The FX market has attracted scholarly attention over the last decade, resulting in a rich body of literature exploring diverse aspects of the market. These include studies into its relation with international economics, capital flows, and trade balance,e.g. [17, 18]. The relationship between the recent COVID-19 public health security incident and the FX market [19]. Further research has analysed the impact of central banks' interventions on the FX market, with a particular focus on banks such as the Bank of Japan, Czech National Bank, and the Bank of Canada [20] [21].

Additionally, a significant portion of studies has also been dedicated to the development of profitable trading strategies for FX trading.In 2016, Coakley et al. [22] empirically examined the profitability of over 100,000 technical trading rules (TTR) across 22 currency pairs in the FX market, revealing potential annualised returns of up to 30%. [23–25] They highlighted that technical trading could indeed yield considerable returns that remained robust even after adjusting for transaction costs. Zarrabi et al., in 2017 [23], echoed these findings. By considering 7,650 TTRs across six currencies, they found that technical trading could maintain positive returns even amidst financial crises, such as between January 2007 and December 2009. Their results underlined the necessity for investors to frequently update their portfolios to align with economic changes and reaffirmed that technical trading could deliver attractive risk-adjusted returns after transaction costs were considered.

2.1.2 Essential Terminologies of Trading Strategies and FX Market

In this section, we describe some essential vocabularies related to trading strategy and the FX market. [26–28]

- Currency Pair: It refers to the quotation and pricing structure of currencies traded in the FX market. A currency pair indicates the amount of the second currency (counter or quote currency) needed to buy one unit of the first currency (base currency), e.g., EUR/USD.

- Exchange Rate: This term designates the value of one currency for the purpose of conversion to another. It represents the number of units of one currency that can be exchanged for a unit of another.

- Bid, Ask, and Mid-price: The 'bid' price represents the maximum price that a buyer is willing to pay for a unit of currency, while the 'ask' price is the minimum price that a seller is willing to accept. The 'mid-price' is defined as the average of the bid and ask prices.

- Limit Order Book: This is an electronic list of buy and sell orders for a specific security or financial instrument organized by price level. It records the interest of buyers and sellers, showing the depth of the market.

- FX Market Maker: These are financial institutions or brokerage companies that provide liquidity to the market by actively buying and selling securities. They set both the bid and the ask prices, seeking to profit from the spread.

- Individuals and Retail FX Traders: These are individual or small investors who buy and sell securities for their personal accounts, and not for another company or organization.
- Transaction Costs: These are expenses incurred when buying or selling securities. They include the cost of executing trades, such as broker commissions, and slippage (the difference between the expected price of a trade and the price at which the trade is executed).
- Percentage in Point (Pip): In the FX market, a pip is a standard unit of measure for the smallest change in an exchange rate between two currencies. It represents the smallest incremental move an exchange rate can make. Usually, one pip corresponds to a change in the fourth decimal place of a currency pair and equates to 1/100th of one percent (0.01%), or one basis point. For pairs that include the Japanese yen, a pip is denoted by the second decimal place. Pip values are significant as they are the basis for calculating the profit or loss in forex trading.
- Historical Trading Data: This refers to past data on a security's price and traded volume, often used for back-testing trading strategies and for technical analysis.
- Rate of Return (ROR): This is a measure of the gain or loss made on an investment relative to the amount of money invested. It is usually expressed as a percentage.
- Maximum Drawdown (MDD): This is the maximum observed loss from a peak to a trough of a portfolio before a new peak is attained. It is an indicator of downside risk over a specified time period.
- Profit Factor: This is a measure used in the evaluation of a trading system, calculated as the gross profit divided by the gross loss. A profit factor greater than 1 indicates a profitable system.
- Market Return: This refers to the overall gain or loss experienced in a market. It is the standard against which the performance of individual stocks or investment portfolios is compared.
- Long/Short Trading and Position: A long position is one where an investor buys a security with the expectation that it will increase in value. A short position, on the

other hand, involves borrowing a security and selling it with the expectation that it will decrease in value and can be bought back at a lower price for a profit. Long/short trading involves combining long holdings of securities that are expected to increase in price with short holdings of securities expected to decrease in price.

2.1.3 Common Trading Strategies

Trading strategies serve as the cornerstone within the realm of financial economics, systematically guiding traders and investors on the timing and execution of trades to fulfill predefined investment objectives. These strategies are meticulously crafted tools, aiding in the navigation of the multifaceted landscape of financial markets, and are deeply entrenched in their financial theory, applicable assets, risk profile, and optimal market conditions [29]. These strategies serve a critical role in risk management, facilitating investors' assessment of investment options, prevailing market conditions, and potential risk mitigation strategies [30].

The literature on trading strategies is rich, encompassing studies on quantitative analysis ranging from technical analysis to intricate econometric models [31]. The advent of machine learning and artificial intelligence has further enriched this domain, where these technologies are leveraged for the development of innovative trading strategies. These strategies capitalize on computational power for identifying patterns and forecasting, significantly augmenting the scope and potential of trading strategies [32]. It is pertinent to remember that no strategy ensures a guarantee against losses, and the dynamic nature of financial markets calls for constant strategy review and adjustment [33].

Among the vast landscape of trading strategies, a couple of prevalent and widely recognized approaches stand out. These strategies exemplify the broad diversity of trading methods, each with its unique logic, advantages, and challenges, necessitating the need for comprehensive understanding and appropriate application in various market conditions. Here are some common and basic trading strategies.

The "Buy and Hold" (B&H) trading strategy, also referred to as a "long-term investing" approach, is a historically popular and widely studied investment strategy in which an investor buys a security, such as a stock or mutual fund, and holds it for an extended period of time. The rationale behind this strategy is that, over the long term, financial markets have shown a general upward trajectory, even amidst short-term fluctuations and volatility. Hsieh

(2021) demonstrated the asymptotic log-optimality of the B&H strategy, emphasizing its potential to achieve long-term wealth maximization under specific market conditions [34]. Additionally, Hsieh et al. (2019) compared the performance of the B&H strategy with high-frequency trading strategies, illustrating its robustness and relevance even in modern financial markets [35]. As such, the B&H strategy is often used as a benchmark for evaluating the performance of other trading strategies or financial instruments. While not identical to the concept of market return, the B&H strategy is closely related and is frequently employed to assess the long-term profitability and risk-adjusted returns of alternative strategies.

The MA strategy is a time-honored approach rooted in technical analysis, deploying averages of historical prices to identify potential trends and reversals [36]. There are various forms of MA strategies, such as simple, exponential, and weighted moving averages, each varying in its emphasis on recent price data. Their core purpose remains consistent - to smooth price movements, alleviate short-term fluctuations, and illuminate overarching trends, thus assisting traders in making informed decisions [37]. This strategy has been extensively investigated in literature [36, 38–42].

Contrarily, the B&H strategy derives its origin from a more passive investing perspective. Rather than attempting to time the market or identify short-term price movements, the B&H strategy advocates purchasing a diversified portfolio of assets and maintaining them over a long-term horizon [43]. This approach is premised on the belief in markets' efficiency and the tendency for investments to appreciate over the long run, thus mitigating short-term market volatilities [44]. While MA strategies leverage historical price data to identify trends and market opportunities, B&H strategies emphasize long-term investment horizons and market efficiency. [27, 36, 38, 39]

The mean Reversion strategy is rooted in the concept that price deviations from the mean or average levels are temporary and will eventually revert to their long-term averages [27,45]. This type of strategy involves identifying assets whose prices have significantly diverged from their historical averages, buying assets that have fallen below, and selling those that have risen above their historical means. However, the key challenge lies in determining the appropriate time frame and measure for the average [46].

The counter-trend strategy, a form or subset of mean reversion strategies but often operating on shorter time frames, is an approach where traders seek to capitalize on price reversals or ”pullbacks” in the market. This strategy operates on the premise that significant price movements are often overextended and are likely to reverse or ”correct” to more sustainable levels. While it shares conceptual similarities with the mean reversion strategy in expecting prices to return to some form of equilibrium, the counter-trend strategy is more concerned with short-term price dynamics and the cyclical nature of markets. Traders employing this strategy often rely on technical indicators to identify potential inflection points where the prevailing market trend may start to reverse [46].

Momentum strategy, conversely, posits that assets trending upwards (downwards) will continue to do so in the short run, and thus seeks to profit from buying assets with strong positive momentum and short-selling those with strong negative momentum [27, 47]. This approach essentially rides on the persistence of trends in the market. However, accurately identifying the start and end of trends poses a formidable challenge.

Pair Trading strategy is a market-neutral strategy that involves identifying two historically correlated assets, and buying (short-selling) one when their price ratio diverges significantly from the historical average [48]. The strategy assumes that the price ratio will converge back to its historical average, potentially yielding profits regardless of market direction.

Trading strategies, such as the Moving Average, Buy-and-Hold, Mean Reversion, Momentum, and Pair Trading strategies, underpin the systematic procedures for executing trades, intending to achieve predefined investment goals [27]. These strategies offer unique approaches to navigating financial markets, playing critical roles in risk management, decision-making, and financial conditions evaluation [29,30]. Each strategy carries unique advantages and challenges, emphasizing the need for comprehensive understanding and appropriate application in varied market conditions.

Machine learning, a subset of artificial intelligence, has been increasingly incorporated in the development of trading strategies due to its ability to learn from data and improve its predictions or actions over time. One example of a machine learning trading strategy is the application of reinforcement learning (RL) to optimize trading decisions. RL algorithms learn an optimal policy that maximizes expected profits by interacting with a market environment, with notable examples being Q-learning and policy gradient methods. RL has shown promise in many finance applications, such as portfolio management and order execution [49]. Another common approach is to use supervised learning algorithms, such as Support Vector Machines (SVM) or neural networks, to predict future price movements or other market variables based on historical data. These predictions can then be used to determine whether to buy, sell, or hold a security [50]. Furthermore, unsupervised learning methods, such as clustering or dimensionality reduction, can be used to identify patterns in the data that might not be noticeable otherwise. [4].

Machine learning-based trading strategies, however, are not without their challenges. They require large amounts of data to train effectively and may suffer from overfitting, where the model learns the noise in the training data instead of the underlying trend. There's also the issue of interpretability, as many machine learning models, especially deep learning models, operate as a 'black box', making it hard to understand the decision-making process [49].

2.1.4 Evaluating the performance

In the context of evaluating a trading strategy, historical data can be employed to estimate its potential future performance, a process referred to as backtesting. Harvey and Liu [51] discussed that backtesting involves simulating past trades by applying the strategy's predefined rules to historical market data. This reconstruction yields statistical measures that are crucial for assessing the strategy's effectiveness. A rule-based trading strategy offers several advantages, particularly in its ability to produce consistent, replicable results and minimize emotional decision-making. Rule-based backtesting offers several key advantages. Primarily, it minimizes the influence of human emotions in decision-making, thus enhancing the consistency and objectivity of the process [52]. Moreover, such strategies can be readily tested on historical data, allowing for a thorough evaluation of their potential performance and associated risks before committing real capital. This study employs multiple evaluation metrics to assess the performance of our strategy, including Profit and Loss ($P\&L$), annualized $P\&L$, Maximum Drawdown (MDD), and the Sharpe ratio.

- Profit and Loss: It can be seem as ratio of return. The total (P&L) measures the absolute financial outcome of a strategy over the evaluation period $[T_0, T_t]$. It is defined as the difference between the final portfolio value and the initial capital.

  To allow comparison across different time horizons, we further define the rate of return as:

$$ROR = \frac{C_t + P\&L}{C_0}, \tag{2.1}$$

where C_0 is the initial capital and C_t is the remaining cash at time T_t. The term P&L represents the cumulative realised and unrealised profit from all positions. In this study, transaction costs are not included in this baseline formulation.

Annualised return is used to standardise performance across different sample lengths. It scales the cumulative return to a yearly basis, enabling direct comparison between strategies operating on different time horizons.

- Maximum drawdown: MMD measures the largest peak-to-trough loss during the trading period. It captures downside risk that is not reflected in average returns.

$$MDD = \max_{t_1 > t_2} \frac{R_{t_1} - R_{t_2}}{R_{t_1}}, \tag{2.2}$$

where R_{t_1} and R_{t_2} denote the portfolio value at a local peak and the subsequent trough, respectively. A lower MDD indicates better capital preservation under adverse market conditions.

- Sharpe ratio: The Sharpe ratio provides a measure of risk-adjusted performance. It compares excess returns to their volatility and is widely used in portfolio evaluation [29].

$$SR = \frac{R_p - R_f}{\sigma_p}, \tag{2.3}$$

where R_p is the average portfolio return, R_f is the risk-free rate, and σ_p is the standard deviation of excess returns. A higher Sharpe ratio indicates more efficient risk-adjusted performance.

2.2 Directional Change

The concept of time, traditionally viewed as a continuous flow, is foundational in finance and economics. However, the notion of intrinsic time, rooted in event-based measures, challenges this paradigm by offering an algorithmic framework that more accurately captures the dynamic nature of real-world phenomena. Intrinsic time reveals novel structures and regularities, obscured by traditional equidistant analyses, and suggests that time is observer-dependent, shaped by the intrinsic activities within complex systems [53]. This approach offers new avenues for economic modeling and a deeper understanding of market dynamics, highlighting the limitations of assuming a universal, continuous time 'intrinsic time'.

This section will elucidate the concept of DC and its application in summarizing market prices. The DC framework is an event-driven technique that measures changes in the underlying asset's price using "intrinsic time" as the system clock, rather than conventional fixed time intervals [54]. This approach allows for the examination of financial market data from a perspective that is driven by price movements and irregular time intervals.

2.2.1 Intrinsic Time

Intrinsic Time is an event-driven concept of time, distinct from conventional physical time. Instead of being measured by fixed, equidistant intervals, intrinsic time is defined by key market events, such as shifts in price direction. Unlike linear time, intrinsic time adapts to the level of market activity. The "clock" of intrinsic time is driven by actual market behavior, accelerating during periods of high activity and decelerating when the market is calm.

The concept of intrinsic time can be traced back to research conducted approximately 60 years ago. It was first introduced by Mandelbrot and Taylor in 1967, who proposed an event-based measure of time as an alternative to traditional physical time [55]. Over the following decades, the idea of intrinsic time evolved further. In 1973, P. K. Clark advanced this concept by employing a "subordinated stochastic process model" in his research, suggesting that trading activities, such as volume or price changes, should define time rather than relying on evenly spaced intervals [56]. In 1996, Olsen & Guillaume coined the term "intrinsic time" to describe a method of defining time scales based on the structure of time series data itself [57]. Their research team introduced an algorithm based on price direction changes, clarifying the concept of intrinsic time [57]. Through this algorithm, they defined "Directional Changes" as a core element of intrinsic time, allowing for the distinction between periods of high and low volatility, and effectively capturing market dynamics by analyzing these changes in financial time series. Later, Tsang [58] optimized the proposed approach with a simpler, more formal definition.

Intrinsic Time is closely linked to the concept of DC, serving as a discretized framework that uses actual market events as the fundamental units of analysis. DC is the core mechanism by which intrinsic time is defined and constructed. The algorithm underlying Directional Change operates by setting a specific price movement threshold (e.g., a percentage) to mark shifts in market direction. Time "ticks" only when the price reaches or exceeds this threshold, allowing the passage of time to be recorded at varying speeds depending on the level of market activity, thereby making the analysis more adaptive. Specifically, the "clock" of intrinsic time does not tick at fixed intervals but advances only when significant market changes occur. For instance, when the price declines from a local high by more than a predefined threshold, intrinsic time records this as a change, reflecting the actual volatility of the market.

Intrinsic time possesses the capability for multi-scale analysis. By setting different price movement thresholds, researchers can analyze the market at various levels of granularity. For example, a smaller threshold may be employed to capture minute fluctuations within the market, whereas a larger threshold might be used to analyze more substantial trend shifts. This multi-scale characteristic enables intrinsic time to reveal market structures across different layers, offering a rich perspective for the study of complex systems [53].

The development of the intrinsic time concept represents a paradigm shift in our understanding of time. Intrinsic time emphasizes that time is observer-dependent and determined by the internal interaction processes of a system. This notion draws upon principles from relativity, where time is not absolute but instead varies relative to the observer and the system's internal activities. In financial markets, it is particularly useful because it adapts to market activity, filters out noise, reveals underlying market structures, and provides a natural foundation for trading strategy design, prediction, and RL within the DC framework.

2.2.2 Directional Change Framework Construction Method

As previously discussed, DC is an event-driven algorithmic framework that can be viewed as the operational realization of the Intrinsic Time concept. This section will focus on the algorithmic implementation of DC. In simple terms, DC identifies shifts in market direction by setting a specific price movement threshold (e.g., a percentage). Each time the price reaches or exceeds this threshold, a directional change is recorded, which serves as the "tick" of intrinsic time. The DC framework is inspired by a technical analysis tool known as the ”Zigzag” pattern, which divides the market into alternating upward and downward trends based on predefined thresholds [28].

In financial research, the trajectory of price movements consistently stands as a pivotal area of investigation. Figure 2.1 shows a quintessential time-series representation of such price trends.

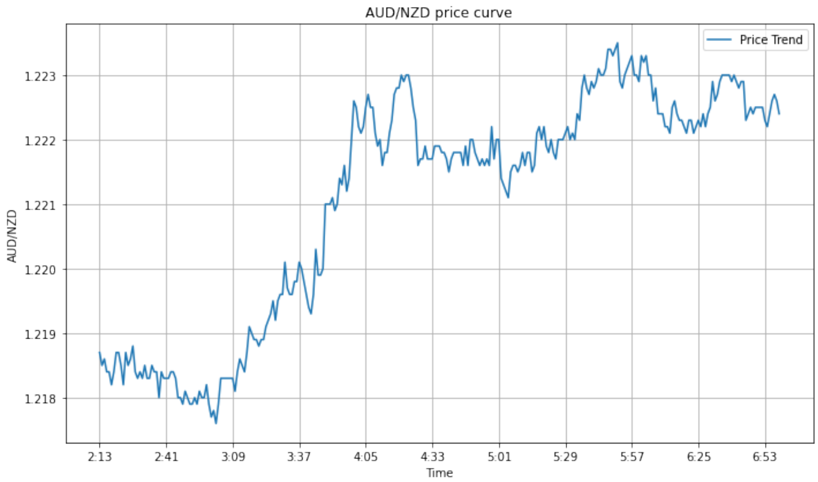

Figure 2.1: The time series price chart for AUN/NZD. The blue line indicates AUD/NZD mid-prices sampled minute by minute.

In the DC paradigm, unlike conventional price fluctuations, it captures price extremes and significant price movements while filtering out intermediate noise. A threshold, denoted as θ, is introduced to express the significance of price changes. This DC structure discretizes the continuous time-price series into alternating instances of price movements. Price trajectories are segmented into alternating upward and downward trends, each composed of a DC event and an Overshoot (OS) event. When the magnitude of a price fluctuation reaches or exceeds θ, it is deemed significant enough to instantiate a new event; otherwise, the current price movement is considered negligible. This section introduces specific symbols based on the DC concept to facilitate a deeper understanding of its principles.

The symbols are defined as follows:

- P_t: Denotes the current price. Throughout the paper, we consider a definition of a price point i given by $p_i = (p_i^{ask} + p_i^{bid})/2$. simpler definition of the price given by

- P_{EXT}: Represents the price extremum nearest to the current DC event.

- P_{EXTh} and P_{EXTl}: Denote the highest and lowest prices respectively, closest to the present DC occurrence. Note that P_{EXTh} is solely used in a declining trend, while P_{EXTl} is specific to an ascending trend.

- $P_{DCC\uparrow}$ and $P_{DCC\downarrow}$: These variables correlate with the confirmation of DC events, where $P_{DCC\uparrow}$ signifies the affirmation of a uptrend DC event, and $P_{DCC\downarrow}$ indicates an downtrend DC event confirmation.

- θ_{DC}: This pre-established value determines the specific shape of the DC curve. Explicitly, the length of a DC event l, which is the absolute difference between the starting price point p_s and the ending price point p_e as well, equates to the absolute disparity between the current price P_t and the subsequent P_{EXT}. The formula for DC is shown in Equation 2.1. Employing diverse θ_{DC} enables researchers to glean insights from variant DC-framework price trajectories derived from an identical foundational price curve.

- OSV: The abbreviation of overshoot value [12]. It measures the length of the overshoot event from the P_{DCC} to the current price. Relative to the set threshold, this indicator can reflect the extent to which the overshoot event exceeds the theoreticalP_{DCC}.

$$OSV = \frac{(P_c - P_{DCC^*})}{P_{DCC^*} \cdot \theta_{DC}} \tag{2.4}$$

- NDC: Number of Directional Change Events. Measures the frequency of DC events during the profiling period, providing unique insights into market price movements.

- $LenC$: Price-Curve Coastline [12]. Measures the length of the price-curve coastline during the profiling period, defined by price changes rather than time intervals.

- TT: Time for Completion of a Trend [12]. This indicator captures the time it takes to complete a trend, from a P_{DCC} to the next P_{DCC}.(from $P_{DCC\uparrow}$ to $P_{DCC\downarrow}$ or from $P_{DCC\downarrow}$ to $P_{DCC\uparrow}$)

$$\left|\frac{P_t - P_{EXT}}{P_{EXT}}\right| \leq \theta_{DC} \tag{2.5}$$

In the context of the DC framework, price movement states are bifurcated into two categories: uptrend and downtrend. Transitions between these price trend states are demarcated by what is termed the DC confirmation points. During an uptrend, the most recent peak, denoted as P_{EXTh}, is continuously updated with any price increment. A DC Event occurs when the price drops from this peak by at least the predefined threshold θ_{DC}, signaling a shift to a downtrend. The current price point, P_t, is recorded as the downtrend DC confirmation point ($P_{DCC\downarrow}$), marking the start of the new downtrend.

Similarly, in a downtrend, the lowest price point or trough, P_{EXTl}, is updated with each price decline. A new DC event is registered when the price rises by a factor of $(1+\theta_{DC})$ from the trough, signaling the beginning of an uptrend. This point is logged as $P_{DCC\uparrow}$, representing the end of the downtrend and the start of the new uptrend. This mechanism allows the DC framework to dynamically capture significant price movements and trend reversals while filtering out minor fluctuations. This heralds the onset of a new uptrend. Follow equation 2.5, a concise and comprehensive definition of this concept is as follows [58]:

$$\begin{cases} a\ downturn\ DC\ confirmation\ point(P_{DCC\downarrow}): & P_t \ \leq \ P_{EXTh} \ \cdot \ (1-\theta_{DC}) \\ an\ upturn\ DC\ confirmation\ point(P_{DCC\uparrow}): & P_t \ \geq \ P_{EXTl} \ \cdot \ (1+\theta_{DC}) \end{cases}$$

Where, P_t is the current price, P_{EXTh} and P_{EXTl} denote the extreme highest point and lowest extreme point respectively. Within the construct of the DC framework, DC events are conceptualized as indicators of trend shifts, while OS events epitomize trend continuations. It is imperative to note that when the extremities of the price range, be it troughs or peaks, coincide with the last P_{DCC}, an OS event is still cataloged, although with a zero-length duration. To elucidate further, DC events encapsulate a time interval characterized by a stable price fluctuation range, delineated by the preordained DC threshold, θ_{DC}; however, their duration remains variable. In contrast, OS events exhibit varied price fluctuation intervals and diverse durations. Following an OS event, a counter-trend DC event is inevitable and post a DC event, a concomitant OS event in the same trend direction is certain.

Based on the theoretical basis of the DC framework, a higher-level pseudocode used to the confirm DC event is shown below:

Algorithm 1 Directional Change Algorithm

Input: Initial price P_0, DC threshold θ_{DC}
Output: Record directional changes based on price movements
Initialize: $trend \leftarrow up$, $P_{next} \leftarrow P_0$
while price data available **do**
    if $trend == up$ **then**
        if current price $> P_{next}$ **then**
            Update P_{next} to new higher price
        end if
        if current price drops by $\theta\%$ from P_{next} **then**
            Record directional change
            $trend \leftarrow down$
            Update P_{next} to current price
        end if
    else
        if current price $< P_{next}$ **then**
            Update P_{next} to new lower price
        end if
        if current price rises by $\theta\%$ from P_{next} **then**
            Record directional change
            $trend \leftarrow up$
            Update P_{next} to current price
        end if
    end if
end while

This DC algorithm tracks the current market trend and continuously updates the latest price extreme (peak or trough). A DC event is recorded whenever the price reverses by more than a predefined threshold θ_{DC} from the most recent extreme. After a DC event is detected, the trend direction is switched and a new extreme is tracked. This process converts the original price series into an event-based representation of market activity, providing the foundation for the intrinsic-time framework.

The inception of the DC concept has remarkably transformed the way we interpret price curves in a fixed time series framework. With the DC concept, these price curves can now be envisaged as event-driven curves, facilitating a richer, dynamic analysis of price trends. To better illustrate this, consider the rudimentary example portrayed in the accompanying figure 2.2.

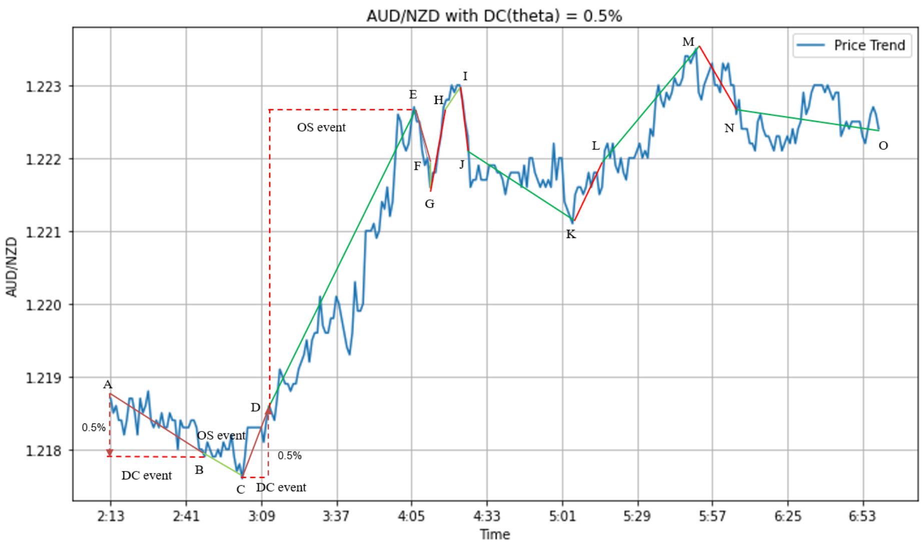

Figure 2.2: An example of a DC-based summary of the price series from Figure 2.1. $\theta_{DC} = 0.5\%$. The blue line indicates AUD/NZD mid-prices sampled minute by minute. Solid red lines represent DC events, and Solid green lines represent OS events. This is to say that the market follows an event cycle of 4 events: ... $\rightarrow$ downtrend DC event $\rightarrow$ downtrend OS event $\rightarrow$ uptrend DC event $\rightarrow$uptrend OS event $\rightarrow$ downtrend DC event $\rightarrow$...

In Figure 2.2, we elucidate the transformation of the time-series price chart from Figure 2.1, into a DC-based event series. This transformation delineates the constitution of DC and OS event sequences and demonstrates the identification process for P_{EXT} and P_{DCC} points. By convention, we initiate the price curve on an uptrend, a preference that doesn't perturb the resultant DC curve but merely standardizes the commencement pattern of the DC. Point A serves as the arbitrary starting point without bearing specific implications, acknowledging the DC framework's flexibility to initiate from any given point.

Within the ambit of the DC paradigm, the price curve decomposes into distinct DC and OS events. The DC event embarks at the P_{EXT} point and culminates at the P_{DCC} point. For instance, in Figure 2.2, a DC event unfolds from point A and concludes at point B, denoted as [AB]. Conversely, the OS event is initiated at the P_{DCC} point and terminates at the ensuing P_{EXT} point. Following a similar paradigm in Figure 2.2, an OS event begins at point D and culminates at point E, represented as [DE].

In the depiction within Figure 2.2, the original price trajectory is portrayed by the blue line. Concurrently, the red solid line delineates the DC events, and the green solid line exemplifies the OS events. Points B, D, F, H, J, L, and N are designated as P_{DCC} points, while points C, E, G, I, K, and M symbolize P_{EXT} points. Point O embodies the terminal price point in this illustration, which can also be perceived as the current price P_t, although it isn't the proximate extremum. This nearest extremity undergoes updates in alignment with fluctuations in the current price P_t. Only upon a trend inversion, signaling the confirmation of a new P_{DCC} point is the P_{EXT} of the current trend ascertained. Concurrently, the establishment and updates of a new P_{EXT} point commence. This graphical representation illuminates the intricate interplay of DC and OS events, underscoring the novel insights afforded by the DC framework in deciphering price movements.

In general, a time period T consists of n points that include time and price information. We have adopted this time period using the DC framework to create a vector composed of m alternating DC and OS events, with each event containing the starting and ending time, price, and trend direction. The vector representation of event e is

$$e = \left[e_{type}, \quad d, \quad p_s, \quad t_s, \quad p_e, \quad t_e, \right] \tag{2.6}$$

Where e_{type} means the type of the event, it has only two values, 0, 1. A value of 1 indicates that the current DC event is an upward trend DC event, while a value of 1 demonstrates that the current DC event is a downward trend OS event. the type of the OS and DC and d is the direction of this event (1 for the uptrend event, -1 for the downtrend event). t_s and t_e

are the start and the end time of the event. p_s and p_e are the start and the end price of the event. Furthermore, based on the vector properties of the events, we have added the length l and duration T of each event as additional features. We then divide the l of the event by its T to obtain a new value, the rate of event ROE.

Then the vector representation of event e is

$$e = \left[e_{type}, \quad d, \quad p_s, \quad t_s, \quad p_e, \quad t_e, \quad l, \quad T \quad ROE \right] \tag{2.7}$$

Where l means $length$ of this event (obtained from $\mid p_s - p_e \mid$) and T is the duration (obtained from$\mid t_s - t_e \mid$). The price movement of a period can be represented by a set of m event vectors grouped into a vector set E.

$$E = \left[e_1 \mid \quad e_2 \mid \quad e_3 \mid \quad ... \mid \quad e_m \right] \tag{2.8}$$

Table 2.1 provides a summary of the various DC-related variables discussed in the previous equations.

Table 2.1: Explanation of DC-related variables

Variable	**Description**
e	An event
E	A collection of events formed by a sequence of events arranged in order.
e_{type}	The type of the event. It takes values $\{0, 1\}$. 0 indicates an upward DC event, while 1 indicates a downward OS event.
d	The direction of the event, indicating whether the event is upward or downward. It takes values $\{$-1, 1$\}$
t_s	The start time of the event.
t_e	The end time of the event.
p_s	The start price of the event.
p_e	The end price of the event.
l	The length of the event, equal to $p_e - p_s$.
T	The duration of the event, equal to $t_e - t_s$.
ROE	the rate of event, equal to l/T.

2.2.3 The Scaling Laws of the Directional Change Framework

Scaling laws represent a fundamental principle in nature, describing proportional relationships between input and output, where changes in the input result in proportional changes in the output. This relationship exhibits scale invariance, meaning it remains consistent across different scales. Scaling laws are ubiquitous in complex systems, observable across diverse fields such as biology, physics, and the social sciences. For instance, the distribution of resources in cities or the structure of ecosystems in relation to species abundance both adhere to scaling laws. In financial markets, the relationship between price changes and time intervals also follows scaling laws. The importance of scaling laws in depicting the behavior of complex systems cannot be overstated. This section outlines the application of scaling laws within the DC framework, which is a key finding in the study of the DC framework.

The study of scaling laws can be traced back to Galileo Galilei, whose work, while not explicitly coining the term ”scaling laws,” introduced concepts related to scaling effects. By investigating how the size of objects influences their motion and properties, Galileo was the first to reveal the principles of scaling [59]. In the 20th century, scaling laws became a systematic part of research and were used to describe scale invariance in complex systems. For example, in the 1950s, linguist ”George Kingsley Zipf” studied word frequency distributions and proposed ”Zipf's Law”, an early application of scaling laws [60]. Similarly, economist ”Vilfredo Pareto” introduced the renowned ”Pareto distribution”, or the ”80/20 rule,” in the late 19th century, which described scaling effects in wealth distribution [61]. In modern science, the application of scaling laws spans various domains, including complex networks, social structures, ecosystems, statistical physics, and financial markets. For instance, price fluctuations and trading volumes in financial markets exhibit characteristics consistent with scaling laws.

A notable feature of scaling laws is their invariance across multiple scales, meaning that the behavior of a system can be described by the same scaling law regardless of the scale of analysis. This implies that certain key patterns within the system persist across different temporal or spatial scales. In the study of scaling laws in financial markets, adjusting the threshold for directional changes allows for the capture of market behavior across varying time scales. This multi-scale characteristic provides flexibility in the analysis of financial data, enabling researchers to uncover hidden market structures across different time horizons. Under the framework of Intrinsic Time, the multi-scale nature of scaling laws is further developed. Figure 2.3 illustrates how fixed-time price charts are transformed into a multi-scale analytical framework under the DC framework. Using different θ_{DC} can get different lengths of *LenC*.The integration of scaling laws' multi-scale properties with Intrinsic Time within the DC framework allows researchers to more accurately capture key events in both high- and low-volatility market environments, while avoiding analytical breakdowns caused by periods of market inactivity.

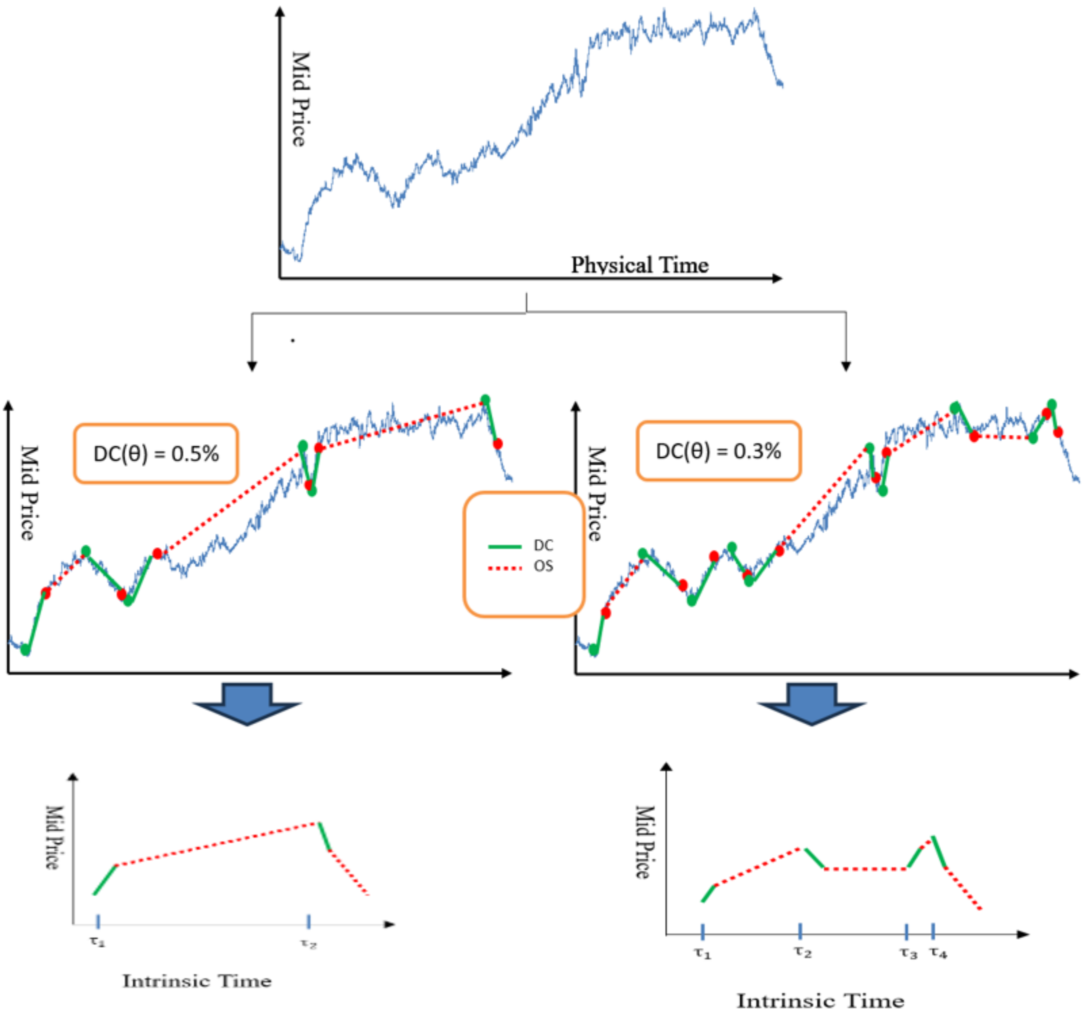

Figure 2.3: Intrinsic time transformations: According to the procedure defined in the main text, the original time series is decomposed into DC and Overshoot (OS) events. Two DC thresholds (0.5% and 0.3%) are used to measure intrinsic time events. In practice, intrinsic time defined by different resolution thresholds results in a transformation from a tick-by-tick time series representation to an event-based price curve, evolving according to the intrinsic time increments τ_i. Thus, the dependence on physical time is replaced by a dependence on intrinsic market activity, which is defined at the selected granularity level, revealing the underlying coastlines.

An intriguing observation from comprehensive studies is the conformity of FX market price movement, within the event-based framework, to scaling-law distributions. These exceptional scaling laws embody the associations between diverse measured quantities. It was

Muller [54] who pioneered the publishing of a scaling law, which described that the average absolute change in price is proportional to the power of the time interval:

$$\langle|\Delta X|\rangle_p = \left(\frac{\Delta t}{C_X(p)}\right)^{E_x(p)} \tag{2.9}$$

where ΔX is the average absolute change in the logarithmic price. it equal to $X_s - X_e$, $X_s = (\ln X_s^{ask} + \ln X_s^{bid})/2$. Δt is the duration of price movement form X_s to X_e, and $E_x(p)$ and $C_x(p)$ are the scaling law parameters. The averaging operator is $\langle \Delta X \rangle_p = (1/n\sum_{i=1}^{n} X_i^p)^{1/p}$, usually with $p \in \{1,2\}$, and p is omitted if equal to one. Note that the data is sampled at fixed time intervals. Although Muller [54] did not use the DC framework at the time, it gave a direction for studying scaling law under the DC framework.

Subsequently followed by the introduction of a second scaling law by Guillaume [57]. He described the relationship between the size of the DC event and the number of DC events within a certain sample time after introducing the DC framework.

$$N(\Delta X_{DC}) = \left(\frac{\Delta X_{DC}}{C_{N,DC}}\right)^{E_{N,DC}} \tag{2.10}$$

where ΔX_{DC} represents the magnitude of price changes, while $N(\Delta X_{DC})$ denotes the count of DC events within a sampled time frame. In this seminal contribution, Guillaume [57] integrated the DC framework into the research paradigm of scaling laws. Eschewing a fixed temporal scale, he delved into the scaling behaviors of financial data through event sequences. This innovative approach delineates a time scale depending on activity, termed 'intrinsic time'. The expansion of this event-driven paradigm holds the potential to uncover distinct and robust scaling laws, mitigating the inherent intricacy of real-world time series data. However, it is pertinent to note that Guillaume [57] did not distinctly differentiate between OS and DC events. He solely employed the DC framework to ascertain uptrends and downtrends. As such, ΔX_{DC} essentially captures the length of either a singular uptrend or downtrend, which is the summation of lengths of both OS and DC events.

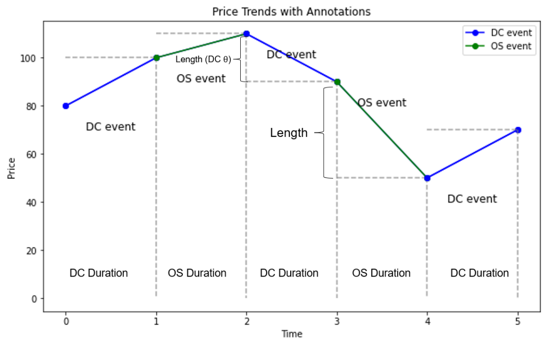

Figure 2.4: Statistical metrics for Directional-change and Overshoot Events.

Inspired by the seminal works of [57] and [54], Glattfelder [1] embarked on a deeper exploration of the DC framework. He not only distinctly differentiated between DC and OS events, providing their algorithmic representations and succinct definitions, but also shed light on a series of prominent statistical scaling laws in financial markets. Building upon the foundational work of [57], Glattfelder [1] employed statistical techniques to validate novel statistical scaling laws characteristic of FX market price trajectories. Specifically, within the DC framework, he examined the relationship between the number of price trend events $N(X_{DC})$in a fixed sampling time and their θ_{DC}.

$$N(X_{DC}) = \left(\frac{\theta_{DC}}{C_{N,X_{DC}}}\right)^{E_{N,X_{DC}}} \tag{2.11}$$

Where, $E(C_{N,X_{DC}})$ and $C(C_{N,X_{DC}})$ are the scaling law parameters in case of θ_{DC}. The correlation between the average maximal price range $max(\Delta X_{DC})$—and the sampling timeΔt:

$$max(|\Delta X_{DC}|) = \left(\frac{\Delta t}{C_{max}}\right)^{E_{max}} \tag{2.12}$$

The relationship between the average duration of a price trend—calculated as the sum of durations from both DC and OS events within a singular price trend—and θ_{DC}.

$$\Delta t = \left(\frac{\theta_{DC}}{C_{t,\theta_{DC}}}\right)^{E_{t,\theta_{DC}}} \tag{2.13}$$

The correlation between the average duration of a price trend Δt encompassing the combined durations of both DC and OS events within a single price trend—and the average length of the price trend X_{DC}.

$$\Delta t = \left(\frac{X_{DC}}{C_{t,X}}\right)^{E_{t,X}} \tag{2.14}$$

Subsequently, by categorizing price moves explicitly into DC and OS events, the total price move, waiting time, and the number of price trends can be written as:

$$|X_{DC}| = |DC_l| + |OS_l| \tag{2.15}$$

$$\Delta t = \Delta t_{DC} + \Delta t_{OS} \tag{2.16}$$

where DC_l and OS_l are the length of DC and OS events. Δt_{DC} and Δt_{OS} are the duration time of DC and OS events. One of the prominent regularities underscores that, on average, a Directional Change event's span is succeeded by an Overshoot event of equivalent magnitude:

$$DC_l \approx C_l \cdot OS_l \tag{2.17}$$

where, in the FX market, the constant number C_l is equal to 1. Additionally, a similar peculiar feature was displayed: on average, the duration of the DC event is half of the duration of the OS event.

$$\Delta t_{OS} \approx C_t \cdot \Delta t_{DC} \tag{2.18}$$

where, in the FX market, the constant number C_t is equal to 2. The key statistical metrics associated with each event, exhibiting potent scaling-law relationships, are illustrated in Figure 2.4. Given these scaling observations, numerous directional-change (DC) based forecasting models and trading strategies have been proposed. Most of the DC models anchor their specific trading rules on one of these significant scaling-law associations.

By applying scaling laws to financial market data, researchers have discovered many statistical regularities associated with price changes. These regularities appear as linear relationships in log-log plots, indicating that price changes in the market follow a power-law distribution. This statistically robust relationship validates the process of segmenting the price curve into uniform directional change and overshoot segments [62]. A plethora of literature (e.g., [58, 62–68]), underpinned by these significant statistical findings of scaling laws, suggests that DC-based trading strategies yield high profitability. The revelations brought about by these statistical characteristics not only present fresh avenues for traders but also unearth an entirely novel research territory [64]. By adopting an event-driven temporal framework, researchers can conduct more flexible market analyses at different time granularities, bypassing the limitations of traditional clock time.

Certainly, the DC framework, as an emergent tool in financial research, brings its own set of limitations and incurs additional overheads. Primarily, the transformation requires extra computational steps, including the identification of key points and OS events within price data, which may necessitate increased computational resources and processing time. Moreover, this procedure generates supplementary datasets as it not only preserves the original price data but also produces related DC and OS event data, thereby augmenting storage requirements. A potential drawback is the loss of data precision; particularly when larger values of θ_{DC} are chosen, this might lead to decreased responsiveness during significant market fluctuations. For applications demanding real-time or near-real-time analysis, the transformation could introduce delays, affecting the speed of decision-making. Additionally, the transformed data may require specific types of analytical models or trading strategies, implying that existing models might need adjustments or redevelopment to accommodate the new data structure. This shift could also present a learning curve for analysts or traders accustomed to standard time series data. Overall, while the DC framework offers a valuable method for reorganizing and analyzing time series data, it also carries additional challenges and costs.

2.2.4 The DC-based Trading strategies

The crux of research initiatives that utilize the Directional-Change Framework fundamentally lies in the development of inventive algorithmic trading strategies. These strategies aim to meticulously assess and question the effectiveness of current forecasting methods and predictive pricing tactics. Through such critical examination, these studies illuminate opportunities for improvement and innovation within the financial markets. The DC Framework, with its unique approach to market data, provides a fertile platform for devising successful trading strategies. These strategies, inspired by the novel observation of regularities within financial markets, have the potential to deliver substantial benefits to traders and investors. Essentially, the DC Framework acts as a springboard for identifying profitable trading patterns and shaping them into actionable strategies.

Kablan [69] introduces a sophisticated adaptive neuro-fuzzy inference system (ANFIS) tailored for high-frequency financial trading. Utilizing five-minute FX market data for training, the system employs an innovative event-driven approach to capture volatility based on DC within defined thresholds. Through optimization techniques such as back-testing and varying epoch numbers, the results clearly indicate that the proposed model surpasses traditional trading strategies like buy-and-hold and linear forecasting in performance.

Dupuis [64] proposed a DC-based contrarian trading strategy that harnesses the power of scaling laws within the FX market. Dupuis' strategy elegantly leverages the relationship between different variables and functions within complex systems, revealing significant opportunities within the dynamics of the FX market. This strategy marks a critical step forward in the application of scaling laws, offering a novel approach to understanding and predicting market trends.

In a similar vein, Gypteau [70] pioneered a trading strategy focusing on forecasting the daily closing price of a financial market. The strategy, conceived within the DC framework, represents a significant contribution to the field of financial prediction and price trend analysis. It demonstrates the practicality and potential of applying the DC framework to daily market analysis and decision-making. The model was tested on four markets: Barclays and Marks & Spencer from the UK FTSE100 market, and the NASDAQ and NYSE indices, with returns reportedly less than 10% over a trading period of 500 days

Bakhach et al. [71] developed a DC-based model with a specific focus on forecasting the future direction of price trends. They introduced an inventive contrarian trading strategy named "TSFDC", further exemplifying the DC framework's ability to predict and profit from future market trends. This strategy, designed to oppose the prevalent market trend, demonstrates the versatility of the DC framework and its potential for facilitating unconventional yet effective trading strategies.

In 2012, Aloud et al., [58] presented a DC-based trading strategy named Zero Intelligence Directional Change Trading (ZI-DCT0). And they (2015) [15] introduced the trading strategy 'DCT1', an updated rendition of ZI-DCT0. Despite adhering to the same trading rules as ZI-DCT0, DCT1 is uniquely designed to autonomously compute parameters such as the DC threshold $\delta_x DC$ and the type of trade (Counter-trend or Trend-following). Aloud's work further revealed DCT1's ability to yield a 6.2% return rate over a one-year testing period using high-frequency data from the EUR/USD currency pair.

In 2017, Kampouridis and Otero [72] introduced an innovative Directional-Change (DC) based trading strategy, designated as 'DC+GA'. This strategy is unique in its approach of operating multiple DC summaries simultaneously, using a multitude of thresholds. For each threshold, DC+GA calculates the average duration of each DC and Overshoot (OS) event throughout every DC trend during a designated training or in-sample period. The fundamental objective of this strategy is to anticipate, with approximate accuracy, the culmination of an uptrend or downtrend.

Golub et al. [62] propose the "Alpha Engine", an agent-based algorithmic trading framework built on the DC paradigm. The core idea is to use DC-based event structures and their associated scaling laws to define market states and generate trading signals. Instead of relying on ad hoc rule sets, the Alpha Engine embeds trading logic within a systematic position management mechanism that governs entry, exit, and exposure in a consistent way. In this framework, trading decisions are driven by intrinsic-time DC events, while risk and position sizing are controlled through a unified agent-based structure. This design allows the strategy to operate in a parsimonious and scalable manner, while maintaining internal consistency between signal generation and portfolio management. Part of the trading rules in our second chapter are derived from this model.

Alkhamees [68] proposes an adaptive intelligent DC pattern recognition model, termed the DCRL model, which integrates RL. By incorporating the environment's behavior into the RL process, the model automates the identification of directional price changes. It adaptively selects varying price features based on the current state, providing a nuanced representation of price time series. The model's efficacy is evaluated on Saudi stock market data with diverse price trends, with analyses demonstrating successful price change detection and the DCRL model's broad applicability.

In essence, the DC event method stands as a formidable tool in the realm of financial trading and market analysis. Conceived as a response to discontinuities in price time series, it offers an innovative approach to sample such data. This methodology has empowered the genesis of a myriad of trading strategies, all demonstrating tangible benefits to market participants. Concurrently, it contributes meaningfully to a diverse array of research applications beyond trading strategies, such as pattern recognition [1], regime change detection [65], event detection [73], forecasting models [71, 74], and most pertinently, designing trading strategies [63, 75–83]. The versatility of the DC method undeniably sustains its substantial influence across these domains, propelling the progression of associated research fields. Furthermore, the DC Framework's unique perspective on financial data enables the forecasting of future price movements and the effective use of scaling laws, thus enhancing its value as an influential tool in financial trading and market analysis.

2.3 Machine Learning Techniques

Machine learning, a cornerstone of artificial intelligence, involves the use of computational algorithms to enable systems to learn from and make decisions based on data. The essence of machine learning lies in its ability to evolve independently, continuously improving its performance without explicit human programming.

A fundamental paradigm within this field is Supervised Learning, a method in which algorithms are trained using labeled data, a dataset where each example is paired with a correct output or label. In the learning phase, the algorithm creates a model by analyzing the training set and identifying patterns that correlate inputs with outputs. Once the model is trained, it can predict labels for new, unseen data. This approach finds vast application in fields such as image classification, spam detection, and predicting disease progression. Mitchell's seminal work provides an excellent starting point for understanding supervised learning, detailing its framework, applications, and challenges [84].

Deep Learning, a subset of Machine Learning, represents cutting-edge of computational modeling. It employs artificial neural networks with multiple layers (hence "deep") to model complex patterns in large volumes of data. These layered networks can learn and represent data with a level of abstraction that makes them capable of handling intricate tasks, including image and speech recognition, language translation, and playing video games.

A significant contribution to Deep Learning was made by Krizhevsky et al [85]. in their work on ImageNet classification with deep convolutional neural networks. Their study laid the groundwork for subsequent applications of deep learning in numerous domains, including finance, where the technology is used for tasks such as credit scoring, algorithmic trading, and fraud detection.

ML has emerged as a powerful tool for designing and optimizing trading strategies due to its ability to model complex, non-linear relationships and extract valuable patterns from large datasets. Its adoption in financial markets stems from the need to handle the dynamic, stochastic nature of price movements, enabling adaptive and data-driven decision-making. In the context of DC trading strategies, ML provides a systematic approach to identify and leverage patterns embedded in price dynamics, enhancing strategy robustness and profitability.

Previous studies have demonstrated that ML techniques excel in analyzing financial time-series data by addressing challenges such as noise reduction, feature extraction, and real-time adaptability [49, 86]. In particular, supervised learning models have been effectively employed to predict market directions, while RL frameworks have been used to optimize trading policies through iterative exploration and exploitation [87, 88]. These approaches have advanced the development of trading strategies by offering data-driven insights that surpass traditional heuristic methods.

By integrating ML techniques with the DC framework, this study investigates whether the characteristics of OS events can be predicted to improve DC-based trading strategies. In particular, the length of an OS event is treated as the prediction target, as it provides information about the potential continuation of price movements after a directional change event. Accurate estimation of OS length enables the strategy to better identify trading opportunities and adjust decisions according to expected market movements.

To achieve this objective, five representative ML models are investigated in Chapter 4: Random Forest, Support Vector Machine, Artificial Neural Network , Long Short-Term Memory network, and Convolutional Neural Network. These models are selected to examine different approaches to OS prediction, including tree-based learning, kernel-based learning, feed-forward neural networks, sequential modelling, and feature extraction from local patterns. Specifically, RF provides a robust non-linear prediction framework based on ensemble decision trees, while SVM offers an effective approach for modelling complex relationships under limited samples. ANN is employed to capture non-linear dependencies among DC-related features, whereas LSTM is considered for its ability to model sequential dependencies in event-based financial data. CNN is further explored for its capability to extract local patterns from structured input representations.

The following sections introduce these ML methods and explain their relevance to predicting OS event length within the DC framework. Their predictive performance and contribution to the construction of DC-based trading strategies are evaluated empirically in Chapter 4.

2.3.1 Random Forest

The exploration of Random Forest models necessitates an understanding of decision tree learning - a cornerstone of classification techniques within machine learning and data mining. Classification models grounded in decision tree algorithms are characterized by their ability to map relationships between object attributes and values, creating a graphical representation akin to a tree. Each tree node signifies an object, every branching point embodies a possible attribute value, and each leaf represents the pathway from root to leaf. Nevertheless, singular decision tree models can become overly complex, particularly for trees with significant growth depth, leading to the potential for overfitting and overlearning. These irregular patterns could contribute to reduced variance but at the cost of increased deviation within the training set.

Random Forest (RF), introduced by Breiman [89], has become a powerful tool in trading strategies due to its ability to handle nonlinear relationships and high-dimensional data. An ensemble learning algorithm, Random Forest (RF) , effectively mitigates the high variance of the decision tree model without amplifying bias, thereby enhancing its performance [89]. Originating from the work of Tin Kam Ho in 1955, Random Forest constructs an array of decision trees during the training phase for classification, regression, and other tasks. It harnesses the concept of bagging in decision tree learning, recognizing that single classifiers may not suffice in accurately defining the class of test data due to noise interference. Consequently, bagging creates a multitude of varied classifiers from decision tree models, which learn from the training data. The collective output, adjudged as the majority consensus among trees, constitutes the final result, thereby largely circumventing the issue of overfitting. Figure 2.5 shows the structure of RF.

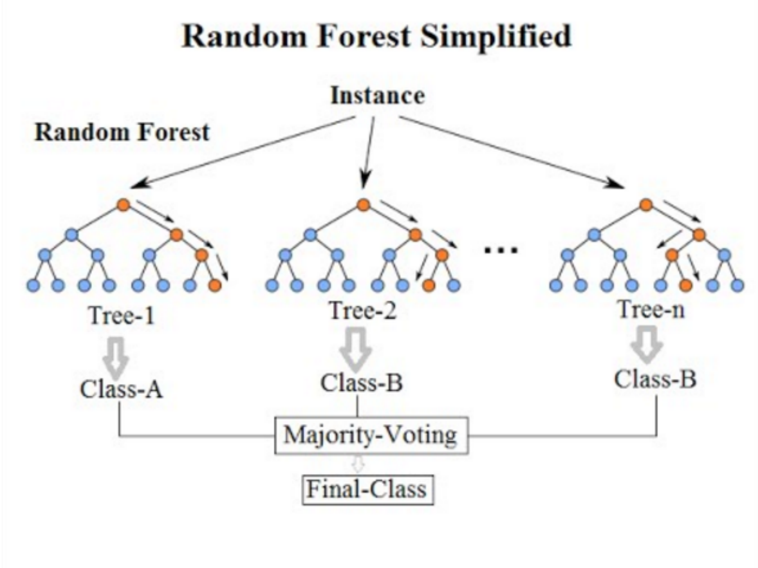

Figure 2.5: Random Forest

Given a training set $X = x_1, \ldots, x_n$ with responses $Y = y_1, \ldots, y_n$, the Random Forest algorithm in its simplest form proceeds as follows:

1. For each iteration $b = 1, \ldots, B$ (B being the number of trees in the forest):
   (a) Draw a bootstrap sample Z^* of size n from the training data.
   (b) Grow a decision tree T_b to the bootstrapped data by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{\min}$ is reached:
      i. Select m variables at random from all p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}$.

To make a prediction at a new point x:

- For regression, it averages the predictions from all the decision trees:

$$\hat{f}_{rf}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(x) \tag{2.19}$$

- For classification, it uses majority votes from all the decision trees.

The parameters B, m and $n_{\min}$ are typically chosen via cross-validation.

The power of RF lies in the fact that as an ensemble method, it leverages the power of multiple ”weak” decision trees to build a more robust and accurate model. By subsampling the input data and variables for each tree, RF ensures a diverse set of models that can capture different patterns and trends in the data, thereby enhancing its generalization capabilities.

In the field of financial transactions, Early applications focused on predicting price trends using technical indicators, where RF demonstrated superior generalization compared to regression-based models. Over time, RF found applications in signal generation, portfolio optimization, and event-driven strategies [90]. For example, RF has been used to analyze order book dynamics and news sentiment, providing buy/sell signals and risk forecasts. Recent advancements integrate RF with other methods, such as deep learning, to improve adaptability in dynamic markets [91]. RF identifies short-term arbitrage opportunities in high-frequency trading by analyzing bid-ask spreads and microstructural data. However, challenges remain, including limited interpretability and difficulties in adapting to sudden market shifts. Looking forward, RF is expected to benefit from alternative data sources (e.g., satellite imagery) and explainable AI techniques [90]. Its robust performance and versatility make it a valuable tool for modern financial strategies, despite inherent challenges in dynamic market conditions.

2.3.2 Support Vector Machines

Support Vector Machines (SVMs), introduced by Vapnik and Cortes in 1995, are highly versatile machine learning models capable of performing linear or nonlinear classification, regression, and even outlier detection [92]. SVMs fundamentally construct a hyperplane or a set of hyperplanes in a high or even infinite dimensional feature space, which can be utilized for classification and regression purposes. The aim is to determine the hyperplane with the maximum margin, leading to optimal separability and generalizability.

Support Vector Machine, introduced by Cortes and Vapnik, has been extensively applied in trading strategies due to its ability to handle nonlinear relationships and high-dimensional data. Early applications focused on binary classification, such as predicting price movements and detecting market events using technical indicators like moving averages and Bollinger

Bands. Over time, SVM was adapted for multiclass classification (e.g., bull and bear markets), portfolio optimization, and high-frequency trading [93]. To improve performance, SVM has been combined with time-series models like ARIMA and feature selection techniques such as genetic algorithms [94]. However, challenges remain, including high computational costs for large datasets and sensitivity to hyperparameter tuning. Recent advancements integrate explainable AI methods and alternative data sources, such as social media sentiment and macroeconomic indicators, enhancing SVM's interpretability and adaptability in financial markets [95].

Depending upon the specific distribution of samples, SVM algorithms can be primarily divided into three categories: Hard-margin, Soft-margin, and Kernel SVM. The Hard-margin SVM aims to find a hyperplane in the feature space that separates all the positive and negative examples perfectly, which is generally suitable for linearly separable data. In cases where perfect separation isn't feasible, the Soft-margin SVM extends the capability by allowing some training examples to be misclassified for the sake of a wider margin. For non-linearly separable data, Kernel SVMs project the input data into higher dimensions where a hyperplane can separate the classes.

In the context of this research, we initially employed the original hard-margin SVM, also known as the linear kernel, as well as three widely used kernel SVMs to establish our learning model. Following an evaluation of the accuracy across each variant of the SVM model, we ultimately chose to employ the Sigmoid kernel of the SVM model due to its superior performance. The function expressions of this kernel, as defined by the Sklearn API Reference, are presented below [96].

$$\kappa(x,x') = \tanh(\gamma \cdot x^T x' + r) \tag{2.20}$$

The specific selection of the SVM kernel significantly influences the effectiveness of the model, and it is thus critical to ensure that the chosen kernel is most suited to the particular structure and distribution of the data.

2.3.3 Artificial Neural Networks

In financial forecasting, Artificial Neural Networks (ANNs) are among the most widely applied AI techniques, gaining popularity for their ability to model complex, non-linear relationships in data. Numerous studies have explored the potential of ANNs in the financial domain, such as those by [97–100], which highlight their versatility and predictive power. These models excel in capturing intricate patterns and trends within financial time-series data, making them particularly suitable for applications such as stock price prediction, risk assessment, and portfolio optimization. ANNs have been employed across various financial fields, demonstrating their efficacy in tasks ranging from credit scoring and fraud detection to high-frequency trading. Their ability to process large datasets and self-learn from data without explicit programming has positioned them as a cornerstone in modern financial analytics. Despite their success, challenges such as overfitting, interpretability, and computational demands remain areas of active research to further optimize their application in finance.

Artificial Neural Networks, also referred to as Neural Networks (NNs), draw inspiration from the biological neural networks found in animal brains. These models serve as powerful tools for non-linear statistical data modeling and analysis. Introduced by McCulloch and Pitts (1943), ANNs consist of interconnected nodes or units (also termed neurons), embodying a multilayer structure comprising an input layer, output layer, and several hidden layers [101].Figure 2.6 shows the structure of ANNs.

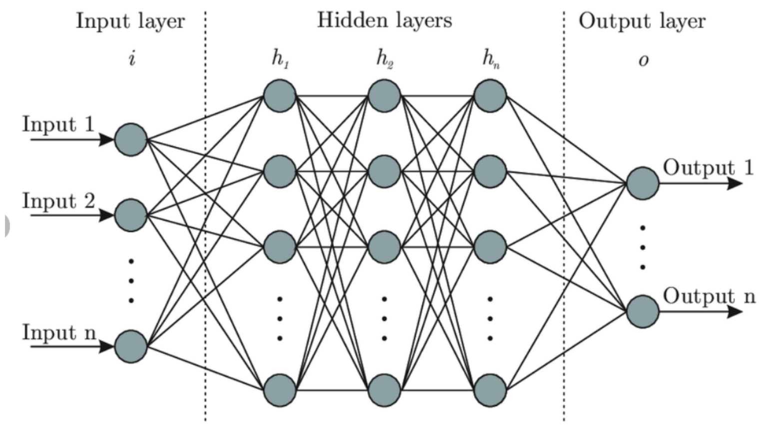

Figure 2.6: ANNs

This composition mirrors the functioning of biological neural networks, wherein neurons receive, process, and transmit information. A neural network's capacity to learn and make sophisticated decisions hinges on the strength of the connections, or weights, between these neurons. The mathematical representation of a neuron's output is given by:

$$y = f\left(\sum_{i=1}^{n} w_i x_i - b\right) \tag{2.21}$$

where x_i are the inputs, w_i are the weights, b is the bias, and f is the activation function.

2.3.4 Long Short-Term Memory

Artificial neural networks have significantly advanced with the inception of Long Short-Term Memory (LSTM), an improved architecture tailored for the analysis and processing of time series data. First proposed by Hochreiter and Schmidhuber in 1997 [101], LSTM has been instrumental in transcending the boundaries of conventional artificial neural networks by integrating feedback connections, thereby allowing the storage and processing of long-term dependencies.

A typical LSTM unit constitutes a cell and three gates: an input gate, an output gate, and a forget gate. The cell acts as the memory unit, preserving values across arbitrary time intervals, while the gates collectively regulate the flow of information into and out of the cell [102].

Mathematically, the current state of the cell at time t, denoted as C_t, is determined by the preceding state C_{t-1} with the intervention of the forget gate f_t. The forget gate, adopting either Sigmoid or ReLu as its activation function, filters specific features from C_{t-1} to compute C_t. Figure 2.7 shows the structure of LSTM.

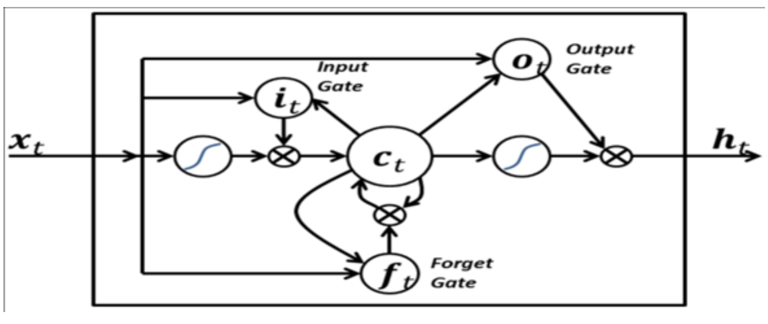

Figure 2.7: LSTM

LSTM, a special kind of Recurrent Neural Network (RNN), have demonstrated considerable promise in financial time-series forecasting due to its ability to capture long-term dependencies. Fischer and Krauss [103] used LSTM networks for stock return predictions and reported substantial outperformance against standard methods. Bao et al. [104] applied LSTMs to predict the directional movements of the Chinese stock market, with compelling results. Moreover, LSTM networks were employed by Kraus and Feuerriegel [105] to forecast intraday stock prices and volatility, demonstrating LSTM's robust performance in capturing complex patterns. These studies highlight LSTM's remarkable potential in revolutionizing financial forecasting and decision-making.

The LSTM model, owing to its unique architecture, has garnered significant attention in the field of financial trading. Studies such as [106–108] have demonstrated its diverse applications in trading strategies, including asset price prediction, intraday trading direction forecasting, and large order execution optimization, all of which exhibit superior performance compared to traditional methods.

In our research, we adopt a three-layer LSTM model where 'ReLu' is the designated activation function for the LSTM layers and 'Nadam' is the employed optimizer. This configuration of LSTM aligns with our research requirements and offers promising prospects for achieving our objectives.

2.3.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs), a distinctive class of Artificial Neural Networks (ANNs), have been predominantly employed to analyze visual data since their inception [109]. Also termed as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), CNNs boast a shared-weight architecture that employs convolution kernels or filters to traverse input features. This methodology yields translation-equivariant responses, which are also known as feature maps. The distinctive feature of CNNs is their ability to autonomously and adaptively learn spatial hierarchies of features, thus facilitating the extraction of patterns from lower to higher levels of complexity.

Convolutional Neural Networks (CNNs), initially designed for image processing tasks [110], have been increasingly applied to financial trading strategies due to their ability to capture spatial and temporal patterns in structured data. Early research focused on using CNNs to analyze candlestick charts and identify trading patterns, demonstrating that CNNs could outperform traditional technical analysis methods [111]. Beyond chart analysis, CNNs have been used to process high-frequency financial data, such as order book dynamics, to predict price movements in real-time trading [112]. Additionally, hybrid models combining

CNNs with LSTMs have been developed to leverage CNNs' feature extraction capabilities and LSTMs' strength in handling sequential dependencies [113].

CNNs operate by carrying out a convolution operation, which is an element-wise multiplication of the filter (kernel) and a portion of the input followed by a sum of all elements in the resulting matrix. Mathematically, if f is the input feature map and g is the kernel, the convolution operation $(f * g)$ at a particular point (x, y) is expressed as:

$$(f * g)(x,y) = \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} f(x - dx, y - dy) g(dx, dy) \tag{2.22}$$

where a and b are the extents of the kernel in the x and y directions respectively. This equation provides a succinct representation of the convolution operation as applied in CNNs [114].

Convolutional Neural Networks (CNNs) and their variants have been increasingly utilized in the financial sector due to their capacity to model complex and high-dimensional data. Particularly, their ability to process temporal sequences and spatial structures in the data has opened new avenues in financial forecasting and trading strategies.

For instance, Zhang et al. [115] proposed a CNN-based model for predicting stock market movements, leveraging temporal close price patterns, market sentiment from news articles, and macroeconomic factors. Their model outperformed traditional methods, highlighting the potential of CNNs in financial forecasting.

Additionally, Cavalcante et al. [116] employed a CNN for forecasting future commodities price movements based on past price changes and volumes. Their CNN model outperformed several traditional forecasting models, further demonstrating the versatility and efficacy of CNNs in financial forecasting tasks.

Despite their success, CNNs face challenges, such as the need for large labeled datasets and interpretability issues. Recent advances, including explainable AI techniques and data augmentation methods, aim to address these limitations and enhance CNN-based trading strategies.

Interestingly, the inception of Residual Networks (ResNet), a CNN variant, has spurred further advancements in the field. ResNet, introduced by He et al. [117], has been shown to significantly improve the performance of CNNs by mitigating the problem of vanishing gradients, thus allowing the training of deeper networks. Its impact has extended to financial applications, further improving the performance and robustness of predictive models.

2.3.6 Residual Convolutional Neural Network

The Residual Convolutional Neural Network (ResCNN) is a deep learning architecture utilized in image processing and computer vision tasks. It draws its inspiration from residual learning and was initially proposed by Kaiming He et al. [117].

The Residual Convolutional Neural Network (ResCNN) is, in fact, a specific variant of Convolutional Neural Network (CNN) designed to address the issue of vanishing gradients that arises with the increasing depth of the network. ResCNN introduces residual blocks, which incorporate what are known as ”shortcut connections” or ”skip connections.” These connections allow the direct addition of the output of one layer to the output of a subsequent layer, thereby creating a ”residual” term. This residual term captures the difference between the input and the expected output, enabling the network to learn the residual rather than the entire transformation process. A residual block consists of two or more convolutional layers, and these shortcut connections enable the output of one layer to be directly added to the output of a subsequent layer. This creates a ”residual” that captures the difference between the layer's input and desired output. Mathematically, this can be represented as:

$$F(x) = H(x) + x \tag{2.23}$$

Where: $F(x)$ represents the output of the residual block. $H(x)$ represents the internal mapping within the block. x represents the input. These shortcut connections have transformative implications. They allow the network to learn residuals directly, rather than trying to learn the entire transformation. If the network finds that the optimal transformation is close to the identity function, it can simply set the weights of the additional layers to near-zero values, effectively skipping them. This means the network can easily learn the identity mapping, and deeper layers can learn more complex transformations on top of that. This mitigates the vanishing gradient problem because even if a deep layer does not contribute much, its residual can still contribute positively to the learning process.

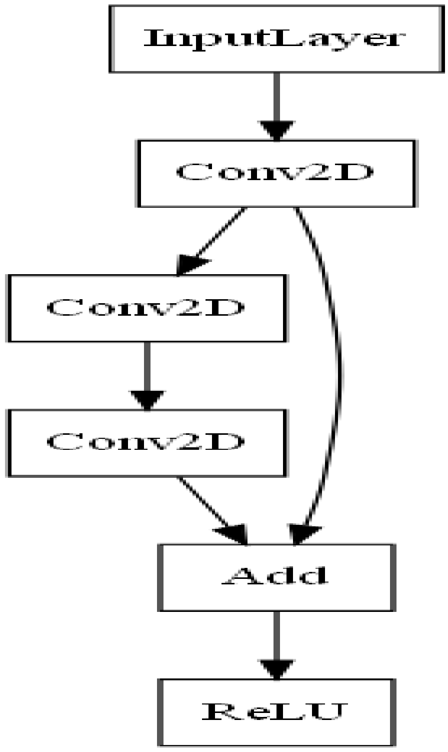

Figure 2.8: Residul block

This architecture results in faster convergence during training and enhances the overall performance of the network. As a result, ResCNNs can be significantly deeper while still maintaining good training dynamics. The use of residual blocks also naturally leads to ”shortcut” paths for gradients, addressing the vanishing gradient issue. Additionally, the direct learning of residuals helps in preserving and even enhancing the information learned in earlier layers, minimizing information loss.

ResCNN’s ability to handle increased depth with shortcut connections makes it highly suitable for constructing very deep networks [118]. The ResNet architecture, an influential variant of ResCNN, has achieved remarkable success in various image processing tasks. It was initially introduced in the context of image classification, where it surpassed conventional architectures and led to substantial performance improvements.

In conclusion, the structure of ResCNN, with its innovative residual blocks and shortcut connections, has revolutionized deep neural network design. By addressing the vanishing gradient problem and allowing the direct learning of residuals, ResCNN achieves faster convergence rates, enhances optimization capabilities, and ultimately delivers superior performance in various image processing tasks.

2.4 Reinforcement Learning

Reinforcement Learning extends machine learning from prediction tasks to sequential decision-making problems, where an agent learns optimal actions through interactions with an environment and aims to maximize cumulative rewards [119]. The success of RL in complex decision-making tasks, such as the game-playing systems developed by DeepMind, demonstrates its ability to learn adaptive strategies in dynamic environments [120].

In financial markets, RL has attracted increasing attention due to its potential for developing adaptive trading strategies. Unlike traditional rule-based approaches, RL-based methods allow agents to continuously evaluate actions, receive feedback, and adjust their decisions according to changing market conditions. These characteristics make RL suitable for applications such as portfolio management, algorithmic trading, and financial decision-making [121, 122].

This section provides the theoretical foundation for the RL-based trading model developed in Chapter 5. In that chapter, a reinforcement learning framework is integrated with the DC framework to construct an adaptive trading agent. The objective is to investigate whether DC-derived market states can improve RL-based decision-making and to evaluate the potential of combining RL with DC for developing dynamic trading strategies.lexity of financial markets and the proliferation of high-frequency trading.

2.4.1 Basic Principles and Components of Reinforcement Learning

RL is a computational approach to learning, characterized by an agent learning from interaction with an environment to achieve a goal. It follows a ”trial-and-error” approach, where the agent makes decisions, observes the outcomes, and adjusts its behavior to maximize a certain reward signal [121].

An RL system is comprised of several key components:

- Agent: This is the decision-making unit that learns from trial-and-error to choose actions that lead to the highest cumulative reward over time.
- Environment: This is the world through which the agent moves and interacts, producing outcomes in response to the agent's actions.

- States: These are the particular conditions or positions in which the agent finds itself within the environment.

- Actions: These are the steps or maneuvers that the agent can take in a given state.

- Rewards: These are the signals that the agent tries to maximize over time. It is the feedback on the agent's actions, indicating how well it is doing.

The fundamental theory behind RL is the Markov Decision Process (MDP), a mathematical framework for modeling decision-making situations where outcomes are partly random and partly under the control of a decision-maker [123]. MDPs are characterized by the tuple (S, A, P, R), where S is a set of states, A is a set of actions, P is the state transition probability matrix, and R is the reward function.

One of the most important concepts in RL is the policy, denoted by $\pi(a|s)$, which describes the probability of taking action a in state s. The goal of RL is to find the optimal policy π^* that maximizes the expected cumulative reward from all states.

To measure the value of each state and action under a specific policy, two functions are defined, the state-value function $V^\pi(s)$ and the action-value function $Q^\pi(s,a)$:

$$V^\pi(s) = \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \middle| S_t = s \right] \tag{2.24}$$

$$Q^\pi(s,a) = \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \middle| S_t = s, A_t = a \right] \tag{2.25}$$

Here, R_{t+1} is the reward after the t-th action, γ is the discount factor ($0 \leq \gamma < 1$), which determines the present value of future rewards [121]. These functions allow the agent to evaluate its policy and adjust it towards optimality.

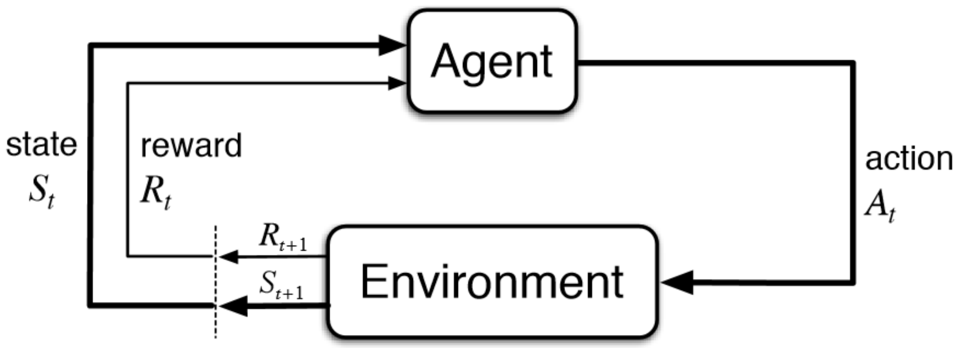

Figure 2.9: Reinforcement Learning Algorithm

[124]. RL has a rich suite of algorithms, with two primary categories: model-based and model-free methods. Model-based methods, such as Dynamic Programming and Monte Carlo Tree Search, build and use a model of the environment, such as the application in [120, 125]. Model-free methods, including Temporal Difference Learning, Q-learning, and Policy Gradient methods, learn directly from experience without a model of the environment [126–129].

2.4.2 Reinforcement Learning in Financial Markets and Trading

This section focuses on the application of RL in financial markets and algorithmic trading. Unlike conventional machine learning approaches, which primarily learn patterns from historical data for prediction or classification tasks, RL aims to learn optimal decision-making policies through continuous interaction with an environment. By receiving feedback from previous actions and optimising cumulative rewards, RL provides a natural framework for sequential decision-making problems.

This characteristic makes RL particularly suitable for financial markets, where trading decisions are dynamic, interdependent, and influenced by continuously changing market conditions. Instead of predicting individual market variables only, RL-based trading systems can directly learn how to select actions, such as entering, holding, or exiting positions, to maximise long-term trading performance. Therefore, this section reviews the fundamental principles of RL and its applications in financial trading, providing the theoretical foundation for the RL-based trading model developed in Chapter 5. In that chapter, an RL framework is combined with the DC framework to train an adaptive trading agent and evaluate the potential of integrating RL with DC-based market representations.

RL has increasingly garnered attention within the financial markets and trading arena due to its significant potential in optimizing decision-making processes and predicting market trends. The foundational concepts of RL, including its capabilities for learning optimal policies through interactions with an environment and leveraging rewards or punishments to inform future actions, prove particularly appealing in financial markets where the state is dynamic and complex [121].

One of the earliest applications of RL in financial markets was to optimize trading strategies. With the rapid advancements in technology, RL's data-driven approach and ability to incorporate risk management in its decision-making process were considered well-suited for market trend prediction and portfolio management [122]. RL provides a mechanism for learning strategies directly from the financial market's interaction experience, which could provide significant advantages in a complex, non-stationary environment [130].

Moody et al. [121] demonstrated the application of RL to the financial markets for the first time. They employed Q-Learning to develop trading systems for the S&P 500 index futures market, indicating the potential of RL for real-world trading applications. Similarly, Deng et al. [49] employed Deep Direct Reinforcement Learning (DDRL) for financial signal representation and trading and found it outperformed other strategies in terms of profitability and risk-adjusted return.

Another critical application of RL in finance has been in portfolio management [131–136]. RL's ability to handle complex decision-making environments makes it an ideal candidate for asset allocation tasks, where it can learn to balance risks and rewards. For instance, Li and Hoi [137] employed RL for portfolio management and showed it could optimize portfolio performance better than traditional methods.

Bertoluzzo et al. [132] also applied RL to the Forex market, one of the most challenging markets due to its non-stationary and high-frequency nature. Their results suggested that RL could handle this complexity and volatility, and yield profitable trading strategies. Meanwhile, RL has been utilized in high-frequency trading (HFT) due to its capability to make rapid decisions in dynamic environments. Nevmyvaka et al. [138] utilized RL for optimal execution in HFT, providing further evidence of RL's robustness and flexibility in finance.

However, the application of RL in financial markets is not without challenges. It often involves dealing with complex and noisy financial data, dealing with a high-dimensional action space, and managing the trade-off between exploration and exploitation [139]. Despite these challenges, the potential of RL to revolutionize financial markets and trading is undeniable.

Conclusively, the emerging applications of RL in financial markets and trading demonstrate the significant potential of this technique for complex, dynamic, and uncertain environments. Its capability to learn from experience, adapt to changing circumstances, and make optimal decisions under uncertainty makes it an ideal choice for financial markets. Further research will continue to explore and develop RL's potential, contributing to a more efficient and adaptive financial industry.

Despite the demonstrated effectiveness of RL in financial trading, several challenges remain. Notably, the application of RL requires substantial computational resources, which can be prohibitive. [140] Furthermore, financial markets are notoriously stochastic and non-stationary, presenting difficulties for the stable and effective deployment of RL methods [141]. Additionally, there is also the need for comprehensive backtesting methodologies to validate the performance of RL algorithms in trading. Looking ahead, the potential integration of advanced RL techniques, such as deep RL [142], and the exploration of more robust, adaptive strategies represent exciting directions for future research. The ongoing development of computational capabilities and novel methodologies may further enhance the application of RL in the financial sector.

2.5 Research Dataset

The selection of an appropriate dataset is critical to ensuring the validity of backtesting results. A well-curated dataset aids in evaluating strategy performance under diverse market conditions and mitigates the risk of overfitting [51]. This section introduces the dataset used to validate the effectiveness of our trading strategy.

Given the demonstrated statistical significance of the DC framework in the FX market [53], we selected the Forex market for backtesting. A detailed introduction to the Forex market and the benefits of its data for DC-based trading strategy research has been discussed extensively in 2.1.

This study relies on the data collected from eight different currency pairs—AUD/NZD, EUR/GBP, EUR/JPY, EUR/USD, USD/CAD, USD/CHF, USD/JPY, and NZD/JPY—spanning the period from January 2006 to December 2020. The selection of these currency pairs was deliberate, encompassing both highly liquid and actively traded pairs as well as less commonly traded, less liquid pairs. Such a selection highlights the comprehensive applicability and robustness of the DC framework. The chosen time frame includes the 2008 global financial crisis as well as more recent periods, ensuring that the model is tested for both its robustness under extreme conditions and its adaptability to contemporary market dynamics.

The dataset spans a substantial time range and provides high-frequency observations, enabling precise analysis of market dynamics. All data used in this study were obtained from "https://www.histdata.com/" and consisted of minute-level high-frequency data. The raw dataset, as shown in Table 2.2, includes the following components: *Date* (the calendar date of the observation), *Minute Time* (the specific minute within the trading day), *Opening Price* (the price at the beginning of the minute), and *Closing Price* (the price at the end of the minute).

Table 2.2: An example of the dataset showing minute-level Forex price data. The minute-by-minute period displayed is from 20:01 to 20:04 on January 1, 2007 for NZD/JPY

Date	**Time**	**Opening Price**	**Closing Price**
2007/1/1	20:00:00	83.76	83.74
2007/1/1	20:01:00	83.73	83.73
2007/1/1	20:02:00	83.73	83.73
2007/1/1	20:03:00	83.73	83.73
2007/1/1	20:04:00	83.72	83.74

For this study, we use the midprice, which is calculated as the average of the opening and closing prices, i.e., $(\text{Opening Price} + \text{Closing Price})/2$—to represent the price at each minute. We identify all DC and OS events in the initial dataset using the DC framework (see Table 2.3 below).

The use of minute-level data enhances the ability of the DC framework to capture scaling laws, allowing for more precise representation of intra-time price changes and capturing subtle movements in the Forex market. Additionally, it enables the strategy's trading agent to better identify and exploit trading opportunities. By leveraging this high-resolution dataset, we ensure the DC framework's applicability and validate its potential in capturing significant market events. The choice of minute-level data allows the strategy to be tested against dynamic market conditions, further solidifying its adaptability and robustness.

Table 2.3: An example of a DC summary using USD/JPY mid-prices sampled minute-by-minute from 17:00:00 to 17:21:00 on 2012-02-05

Date Time	Mid-price	Event Type
2012/2/5 17:00	76.5900	$P_{DCC\downarrow}$
2012/2/5 17:01	76.5805	OS event (DOWNTREND) start
2012/2/5 17:02	76.5850	
2012/2/5 17:03	76.5820	
2012/2/5 17:04	76.5845	
2012/2/5 17:05	76.5840	
2012/2/5 17:06	76.5815	
2012/2/5 17:07	76.5775	
2012/2/5 17:09	76.5705	DC event (DOWNTREND) start
2012/2/5 17:10	76.5775	
2012/2/5 17:11	76.5825	
2012/2/5 17:12	76.5945	
2012/2/5 17:13	76.5840	
2012/2/5 17:14	76.5875	
2012/2/5 17:15	76.5960	
2012/2/5 17:16	76.5780	
2012/2/5 17:17	76.5865	
2012/2/5 17:18	76.5825	
2012/2/5 17:20	76.5885	$P_{DCC\uparrow}$
2012/2/5 17:21	76.5920	OS event (up) start

Chapter 3

Trend-following vs Counter-trending: a comparative analysis

This chapter investigates two research questions within the Directional Change (DC) framework. First, we examine whether a DC-based trend-following strategy (DC-TF) can outperform a DC-based counter-trend strategy (DC-CT) in the foreign exchange market. Second, we evaluate whether a DC-based dynamic stop-loss mechanism can improve the risk-return characteristics of these trading strategies.

To address these questions, two trading strategies derived from the scaling laws of the DC framework are developed and systematically compared. Their profitability, drawdown characteristics, and risk-adjusted performance are evaluated across a range of DC thresholds. Furthermore, a DC-based stop-loss system is incorporated into both strategies to assess its impact on trading performance and risk management.

The empirical analysis is conducted using eight major currency pairs over the period from 2006 to 2014. This relatively long historical window includes different market regimes, including the 2007–2009 global financial crisis, allowing the strategies to be evaluated under varying market conditions. Nevertheless, the results are obtained from a single historical sample period and a specific set of foreign exchange instruments. Therefore, while the findings provide evidence regarding the effectiveness of the proposed strategies, their robustness and generalizability to other asset classes, market environments, or future periods should be interpreted with appropriate caution.

3.1 Introduction

In the context of our research domain focused on applying the DC framework in financial trading strategies, it becomes evident that this framework presents a promising alternative to the traditional time series framework. The Directional Change framework, as a response to the limitations of the time series framework, introduces the concept of event-based analysis, wherein significant price changes are identified as "DC events." This framework inherently addresses the irregularity of financial markets and the inadequacies of fixed time interval-based sampling.

A significant characteristic of research based on the Directional Change framework is its categorization of trading strategies into two distinct types: trend-following and counter-trend. These strategies are not merely the result of arbitrary choices, but rather they emerge from the underlying nature scaling law of DC events. The trend-following strategies capitalize on sustained price movements in the same direction, while Counter-trend strategies exploit the reversal of trends. This dichotomy aligns with the inherent nature of financial markets, wherein trends and reversals are integral components.

The pivotal motivation behind this chapter is to conduct an in-depth comparative analysis of these two distinct types of strategies within the DC framework. By critically evaluating their advantages and limitations, our research aims to shed light on which approach is more suitable for specific market conditions and objectives. Furthermore, our study delves into the effectiveness of incorporating stop-loss mechanisms within DC-based trading strategies. These mechanisms, acting as safeguards against excessive losses, are crucial in maintaining risk management within trading activities. Grossman's work [143] and Kaminski & Lo's study [144] have notably validated the effectiveness of stop-loss strategies across diverse financial scenarios. Their research provides a solid foundation for our exploration into the risk and reward dynamics inherent in DC-based trading models.

It's noteworthy that the application of the Directional Change framework often finds resonance within the FX market due to its unique attributes, such as high liquidity and continuous operation. Existing research that opts for the DC framework generally prefers to utilize data from the FX market, and our study is no exception. With its unique properties, the FX market provides an ideal data source for DC framework research. The daily trading volume of the FX market is approximately 5 trillion USD, far surpassing any futures or equity markets. This immense trading volume translates into rapid price changes, virtually every second. This high liquidity state is not bound by any specific exchange rules, rendering it highly compatible with DC-based analysis. Furthermore, the FX market is a long/short symmetric market, operating 24/7. These trading rules lead to a more balanced price curve and fewer interruptions during any selected period, enhancing the applicability of any DC framework.These attributes, coupled with the framework's event-based approach, provide an ideal setting for assessing the framework's potential effectiveness.

In the following sections of this chapter, we introduce and compare two novel trading strategies grounded in the Directional Change framework. Through our comparative analysis, we strive to provide valuable insights into the nuances of trend-following and Counter-trend strategies, facilitating informed decision-making for traders and investors seeking optimal trading approaches in the financial markets. Further, by incorporating a DC based dynamic stop-loss system, we delved into the potential profit and associated risks inherent in DC strategies.

The outline of this chapter is organized as follows: Section 3.2 gives an overview of the Directional Change framework and introduces the two types of DC-based strategies: Trend-following (TF) and Counter-trend (CT). The stop-loss system is still introduced in this section. It still proposed two trading models representing DC-based TF and CT trading strategies and introduced the DC-based dynamic Stop-loss Section 3.3 presents our experimental results and the analysis of the results. Finally, conclusions and future work are drawn in Section 3.4.

3.2 Related Work

The concept of DC has garnered substantial attention in the realm of financial analysis, offering a distinct departure from conventional methodologies. DC, as an event-based approach, has precipitated a paradigm shift in the understanding and modeling of price dynamics. In previours chapter, we survey the landscape of DC-related research to illuminate the advancements and insights that have emerged.

DC's inception finds its origins in the 'Zigzag' indicator of technical analysis, as cited by Colby and Meyers in their seminal work [145]. Building upon this foundation, subsequent scholars, including Guillaume et al. [146] and Tsang [147], propelled the concept by formalizing its definition and applying it to financial markets. These early works laid the groundwork for comprehending price movements in terms of discrete events, distinct from the uniform time intervals of classical time series analysis.

In summation, the DC framework, driven by its event-centric methodology, has redefined the financial analysis landscape. Its application spans diverse areas, ranging from anomaly detection to trading strategy formulation. The convergence of DC with trend-following and Counter-trend strategies has further enriched its practicality. This review underscores the burgeoning significance of DC in financial analysis and its potential to reshape decision-making paradigms.

3.2.1 The Directional Change framework: A brief overview

The DC framework offers a unique method for describing price movements. While Section 2.2 provides a comprehensive discussion of this approach, this subsection presents a brief summary to introduce the subsequent research. The DC framework transcends the limitations of interval-based summaries by incorporating the concept of "intrinsic time," defined by significant events rather than fixed time intervals. In this context, "events" refer to price changes deemed critical by the observer. The market can be perceived within the DC framework as a series of alternating events, as illustrated in Figure 3.1. The purpose of adopting this event-based summary for time series is to filter out irrelevant details in price evolution and to capture significant market price movements.

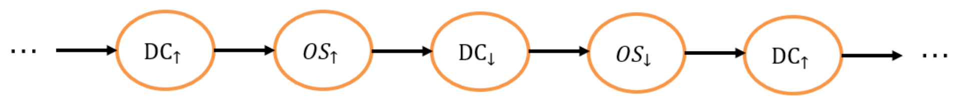

Figure 3.1: chain of alternating DC and OS events

In the DC Framework, the unit ticks of the analysis system are not the fixed intrinsic time, but the alternating price movement events. The price curves are divided into two types of tick events: DC Event δ and OS Event ω. Price changes are recorded as a tick event if, and only if the range of this change is at least equal to a pre-defined DC threshold θ (shown as a percentage in general). By adjusting θ, analysts can observe different price curves from the same original data. During an uptrend, the price continues to be updated at the peak, and a DC event is recorded if the price drops by at least θ from the peak. Conversely, during a downtrend, the lowest point is adjusted until the price increases by θ from the trough, signaling an uptrend. Tsang gives a simple and formal definition [58]:

$$\begin{cases} a\ downturn\ DC\ confirmation\ point(P_{DCC\downarrow}): & P_t \ \leq \ P_{EXTh} \ \cdot \ (1-\theta_{DC}) \\ an\ upturn\ DC\ confirmation\ point(P_{DCC\uparrow}): & P_t \ \geq \ P_{EXTl} \ \cdot \ (1+\theta_{DC}) \end{cases}$$

Where P_t is the current price, and P_{EXTh} and P_{EXTl} are the extreme high and low points. During an upward run, the recent high is updated to the maximum of the current price and recent high, while during a downward run, the recent low is updated to the minimum of the current price and recent low; an upturn DC event occurs when the price rise from the recent low exceeds the threshold θ_{DC}, and a downturn DC event occurs when the price fall from the recent high surpasses θ_{DC}.

The DC framework introduces a distinct asset price portrayal paradigm, diverging from the conventional time-based approach. By adjusting θ_{DC}, it reveals multiple layers of market trends, adaptable to different observation windows, such as daily or hourly movements. This adaptability harmonizes with the diverse temporal preferences of market participants, spanning from daily to hourly trends. By moving away from a time-centered approach, it offers new insights into asset price dynamics and provides a broader perspective on market behavior.

3.2.2 Scaling laws

One of the most intriguing scaling laws discovered by Glattfelder et al. (2011) serves as a fundamental basis for constructing DC-based trading strategies. Specifically, the study by [1] statistically revealed that within price curves constructed using the DC framework, there exists a proportional relationship between the average lengths and durations of DC events and OS events. In the FX market, a DC event with a θ is typically followed by an OS event with the same threshold θ. Furthermore, if a DC event takes $\hat{u}$nits of physical time to complete, the subsequent OS event will take approximately 2$\hat{u}$nits of time. Figure 3.2 summarizes this observation, which holds true only under the DC-based price summary and not under the physical time summary. This finding was consistent across all 13 different currency exchange rates examined in the study by [1]. Consequently, we hypothesize that if these statistical properties are appropriately leveraged, they could yield profitable trading strategies, primarily because these properties are not yet widely known among traders. Thus, the DC framework represents a rich field of study with the potential for significant discoveries.

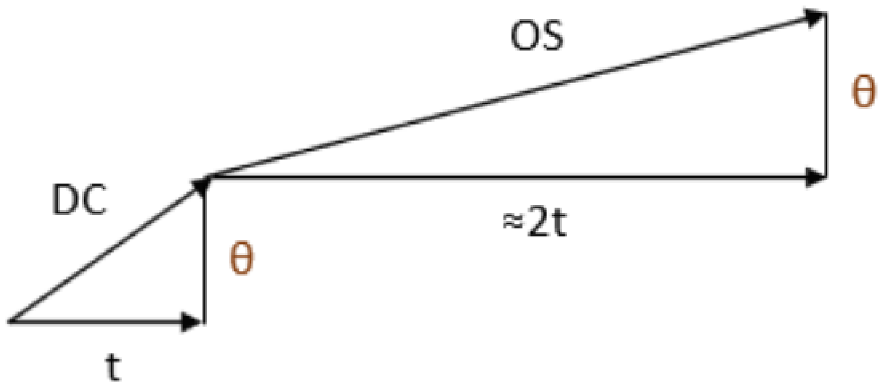

Figure 3.2: An example of a scaling law presented in [1], that shows (a) price changes in DC and followed OS is approximately same and the average length both equal to θ_{DC}, (b) the OS event lasts about the double amount of time that it took for the DC event to take place

.

To be specific, [1] discovered an empirical consistency between numerous foreign currency data that the length of DC (i.e. price changes of DC) and followed OS is approximately the same. It can be formulated as:

$$DC_l \approx C \cdot OS_l \tag{3.1}$$

where the constant number C is equal to 1 [1].

Another important regularity in Figure 3.2 is that the duration of an OS event is approximately twice the duration of its DC event. We can still formulate it as:

$$\triangledown OS_T \approx 2 \cdot \triangledown DC_T \tag{3.2}$$

where $\triangledown OS_T$ and $\triangledown DC_T$ represent the average durations of OS events and DC events, respectively.

3.2.3 The Directional Change based trading strategies

An important research focus utilizing the DC framework centers on the development of inventive algorithmic trading strategies. Numerous strategies grounded on the DC framework currently exist, which are committed to the meticulous examination and scrutiny of the effectiveness of prevailing forecasting methods and pricing strategies. Through rigorous inspection, these investigations shed light on opportunities for refinement and innovation within the financial market domain. The DC Framework, with its distinctive approach to market data, provides a rich platform for devising successful trading strategies. These strategies, galvanized by fresh observations on the regularities of financial markets, hold the potential to yield substantial benefits for traders and investors. Fundamentally, the DC framework acts as a springboard for identifying profitable trading patterns and molding them into actionable strategies.

DC-based Trend-following: An investment strategy widely adopted for its lucid underpinnings, TF operates on a rule-based framework that takes cues from the directionality of market price trends. Anchoring on the tenet that prices manifest consistent trajectories, it posits that select traders, privy to information ahead of the masses, set these directional tones [148]. Consequently, upward price surges see TF aficionados executing buy orders, and vice versa for downturns. Yet, the strategy's broader adoption grapples with complexities, namely defining the essence of a 'trend' and anticipating its cessation. Here, the DC framework emerges as an invaluable asset. As an event-based tool, not only does it demystify 'trend' contours, but its inherent statistical property, the 'scaling law', proffers discernible trend duration expectations. Financial literature [62, 67, 72, 75, 81, 83, 149–154] vouches for the potency of DC-TF, evidencing successes across stock, currency, and commodity futures markets. Figure 3.3 describes the opening and closing logic of the DC-TF strategy.

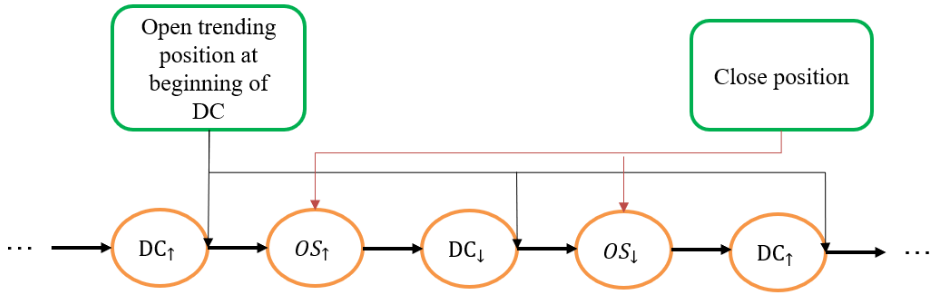

Figure 3.3: The DC-based Trend-following strategy, which capitalizes on price trends using the DC framework. The strategy opens a position at the start of a confirmed DC event, following the trend direction. And expects to close the position before the next DC event occurs.

We now provide a detailed description of the DC-TF strategy's specific paradigm. The core idea of the TF strategy is to initiate positions in the direction of the prevailing price trend. A key aspect of conventional TF strategies is determining the direction and existence of trends. Defining a clear upward or downward trend in traditional time series price charts can be challenging. However, this issue is readily addressed within the DC framework. Here, price curves are understood as a sequence of alternating directional events, forming a price event curve. An upward trend typically consists of an upward DC event followed by an upward OS event. A change in the current trend is recognized only when a new DC event is reported, establishing a new trend. Therefore, in the context of the DC-TF strategy, we can consider that a new upward or downward trend commences at the confirmation point of a DC event, allowing the strategy to establish a position in the same direction as the current trend.

The subsequent question concerns the exit strategy. There are generally three approaches for exiting the DC-TF strategy. The first involves closing the position when a subsequent trend is established; in the DC framework, the emergence of a new DC event signifies a trend shift, thus one exit method is to close the position at the confirmation point of the new DC event. The second method leverages the scaling laws inherent in the DC framework. Statistically, it is known that the lengths of DC events and OS events in the Forex market are equivalent, with the duration of DC events being half that of OS events. Utilizing this insight, the DC-TF strategy may exit when the price curve continues to extend in the direction of the entry for a duration equivalent to the length of the DC event (determined by θ_{DC}). Alternatively, the DC-TF strategy may exit after twice the average duration of the DC events following the entry.

In summary, the DC-TF strategy initiates a position in the direction of the trend at the endpoint of a DC event. It can be closed in three cases: when a new DC event is confirmed, when the price has advanced the length of the DC event in the direction of the trend, or when the elapsed time reaches twice the average duration of the DC events.

DC-based Counter-trend : A strategy echoing CT's nuances but predicated on antithetical principles. CT hinges on the belief that prices oscillate around their intrinsic values. An excessive deviation from this value—either an overvaluation or undervaluation—propels the market to course-correct, engendering counter-trends. Thus, when prices soar excessively, CT practitioners place sell orders, and conversely, they buy during sharp declines. The genius of the DC framework again shines here, as its 'scaling law' furnishes clarity on trend longevity, ushering in an era of refined DC-CT strategies. Revelations from financial literature [62, 64, 67, 72, 75, 81, 83, 149–156] bear testament to the prowess of DC-endorsed CT approaches.

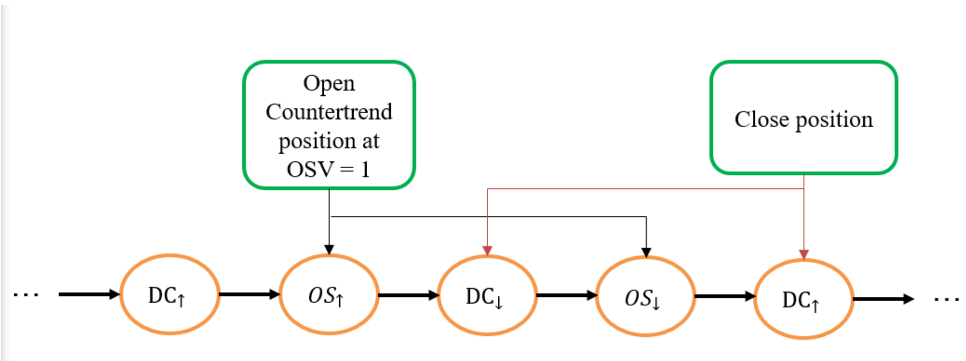

Figure 3.4: The DC-based Counter-trend strategy. The strategy opens a countertrend position when OSV equals 1. And expects to close the position after the next DC event has been reported.

Now we will describe the specific paradigm of the DC-CT strategy in detail. The core idea of the CT strategy is based on the expectation that significant deviations in price from its intrinsic value will generate a counter-trend. Therefore, positions are typically opened when the current price trend is anticipated to end. In conventional time-series price charts, identifying the precise moment when a trend ends is challenging. However, within the DC framework, the boundary between upward and downward trends is clearly defined. Furthermore, using the scaling laws within the DC framework, we expect the length of an OS event to be similar to that of a DC event. Thus, the DC-CT strategy hypothesizes that most price trends will reverse once the OS event extends to the length of the DC event (i.e., when *OSV* equals 1, the definition of *OSV* from 2.4). Based on this hypothesis, the DC-CT strategy initiates a position opposite to the current trend when OSV equals 1.

Next is the issue of deciding when to close the position. The CT strategy's closing logic is based on confirming that the price has returned to its normal value, signaling the end of the arbitrage opportunity. Within the DC framework, there are typically two scenarios in which the DC-CT strategy generates a sell signal. The first occurs at the confirmation of the next DC event. This approach is supported by the scaling laws of the DC framework, as the DC-CT strategy assumes that the entry point roughly coincides with the timing of a new DC event, which has a fixed length. Therefore, take-profit is set at the confirmation of the new DC event or upon reaching the predefined profit target. However, in some cases, OS events may extend in the direction of the trend for an extended period. In such cases, the first method of closing the position may result in insufficient returns. Hence, a second method involves setting a take-profit target equal to the length of the DC event.

In summary, the DC-CT strategy opens a position in the direction of the trend at the end of an OS event when OSV equals 1. The position can be closed under two conditions: upon confirming a new DC event or when the take-profit target(θ_{DC}) is reached. Figure 3.4 describes the opening and closing logic of the DC-CT strategy.

These strategies, enhanced by the DC paradigm, herald a new frontier in financial trading, one steeped in precision and insight. Pioneering research has demonstrated the potential of the DC framework for creating effective trading strategies within the FX market and other financial markets. Kaplan [69] harnessed DC's inherent adaptability to create an innovative trading strategy, while Tsang [157] expanded its application to anomaly detection. Additionally, dynamic threshold methods [157] have emerged, optimizing DC's thresholds in response to market dynamics. Dupuis [64] introduced a contrarian trading strategy rooted in DC, while Bakhach et al. [67] ventured into forecasting price trend directions. These endeavors underscore the versatility of DC as a platform for devising trading strategies that align with prevailing market conditions. Dupuis [64] applied scaling laws to a DC-based contrarian strategy, offering a unique perspective on market trend prediction. Gypteau [70] developed a daily market prediction model within the DC framework, while Bakhach et al. [153] designed a forecasting model to predict future price trends. These models exhibit the applicability of the DC framework in both daily market analysis and future trend prediction. Aloud et al. [63] contributed to the field with the Zero Intelligence Directional Change Trading (ZI-DCT0) strategy and its updated version, DCT1, both grounded in the DC framework. Their work emphasized the DCT1's promising return rates, indicating the strategy's effectiveness. Kampouridis and Otero [72] introduced the 'DC+GA' strategy, operating multiple DC summaries simultaneously, showcasing the framework's versatility. The 'Alpha Engine' by Golub et al. [62] integrated a position management system within the DC framework, enhancing rule regulation and contributing to robust research foundations. Alkhamees [68] proposed the adaptive DCRL model that fuses Reinforcement Learning with DC, showing promising results on Saudi stock market data.

In summary, these studies demonstrate the DC framework's potential for trading strategy development, trend forecasting, and future market analysis. In this section, we commence by revisiting the concept of Directional Changes, which forms the bedrock of our discussion. Following this, we delineate two trading strategies that have been extrapolated from the DC framework, further expanding on our central theme. We initiate this section by revisiting the concept of Directional Change, which serves as the bedrock of our discussion. Subsequently, we examine an array of trading strategies rooted in the DC paradigm. Notably, many of these strategies predominantly derive from the scaling law inherent in the DC framework, giving rise to DC-based trend-following and Counter-trend trading strategies. This thesis categorizes these DC-centric models into two primary classifications based on their affinity for either trend-following or Counter-trend approaches, as delineated in Table 3.1.

3.2.4 The Simple DC-TF Trading Strategy

In the derivation of our illustrative DC-TF strategy, the intrinsic scaling laws of the DC framework intersect profoundly with the foundational principles of the TF strategy. The operational premise behind the TF approach rests on the assumption of trend persistence. The

	Trend-following	Counter-trend
[69]	Kablan et al.	
[64]		Dupuis
[149]	ZI-DCT0	ZI-DCT0
[150]	DCT1	DCT1
[151]	DCT2	DCT2
[152]		BA
[153]	DBA	
[75]	Alkhamees et al.	Alkhamees et al.
[154]	GP-DC	
[72]		DC+GA
[67]		TSFDC
[62]		Alpha Engine
[83]	Ye et al.	
[81]	TA	
[155]		DC+CSFLA
[155]		DC+PSO
[156]		Salman et al.

Table 3.1: The DC-based trading strategy employing trend-following or Counter-trend trading rules.

DC paradigm captures price trajectories as a sequence of alternating uptrends and downtrends, partitioning each trend—whether uptrend or downtrend—into distinct DC and OS events. From a statistical vantage point, the scaling laws of the DC framework suggest that within the FX market, the duration of a DC event closely mirrors the length of an OS event. This leads us to a plausible hypothesis that the length of an OS event approximately aligns with that of a DC event. Thus, upon the confirmation of a trend—specifically at the point of DC confirmation—it is highly probable that a subsequent OS event, roughly equating to the length of the DC event, will ensue. Given that the length of a DC event is denoted as θ_{DC}, our DC-TF strategy entails initiating a position at the DC confirmation point that aligns with the prevailing trend direction—going long in an uptrend and short in a downtrend. Concurrently, the profit target PT for this position is set equal to θ_{DC}.The reason why the DC-based strategies set the θ_{DC} as profit target is because of the scaling laws of the DC framework. "' This study incorporates the position management mechanism proposed in the "Alpha Engine" framework [62]. Unlike many DC-based trading strategies that allocate the entire available capital at each DC event, the Alpha Engine adopts a more realistic capital allocation approach by gradually building positions through multiple small trades.

Golub et al. [62] introduced the ”Alpha Engine” as an agent-based DC trading framework that constructs trading decisions based on intrinsic-time events and DC scaling laws. A key component of this framework is its position management system. Instead of opening a full position at every DC signal, the Alpha Engine divides the available capital into multiple smaller units. Initially, each DC entry point only allocates approximately 1% of the total capital to a new trading position. These positions are treated as independent trading agents and are accumulated over time when new trading opportunities emerge. Such a mechanism reduces the exposure to individual entry points and better reflects the gradual position-building behaviour commonly observed in practical trading systems.

The ’Alpha Engine’ [62] does not employ a conventional stop-loss mechanism. Instead, each trading agent remains open until the predefined profit target is reached. When the accumulated exposure becomes excessive, the money management module reduces the allocation ratio of subsequent positions, preventing uncontrolled growth in market exposure. This dynamic position adjustment mechanism allows the strategy to manage risk through capital allocation rather than relying solely on fixed exit rules.

In addition to position management, the ’Alpha Engine’ [62] employs multiple DC strategies operating simultaneously with different threshold values of θ_{DC}. By combining four distinct θ_{DC} settings, the framework captures price movements across different market cycles and time scales. Smaller thresholds allow the strategy to respond to short-term fluctuations, whereas larger thresholds capture broader market trends. This multi-scale design improves the adaptability of the strategy under different market conditions. This multi-scale design improves the adaptability of the strategy under different market conditions.The strategy was extensively back-tested using 23 currency rates from 2006 to 2014, demonstrating an impressive return of 21.34% over eight years with a maximum drawdown of 0.71% (calculated daily).

Compared with traditional DC strategies that typically rely on a single threshold and full position execution at each trading signal, the ’Alpha Engine’ [62] provides a more flexible framework by combining event-based trading signals with dynamic capital management. This approach offers an important reference for developing more realistic DC-based trading strategies.

Algorithm 2 DC-TF Trading Model Logic

Initialize: Set up trading environment and set θ_{DC}; set profit target PT
while price data is available **do**
    if there exist open positions **then**
        for all open positions **do**
            if position profit $\geq PT$ **then**
                Close the position
            end if
        end for
    end if
    Read current price point P
    if P is not the directional change triggering price P_{DCC} **then**
        Read next price point
    else
        Capital Check: Is there available capital?
        if available capital exists **then**
            Open a new position using 1/100 of the total available position in the current trend direction; assign profit target PT
        end if
    end if
end while
Terminate Trading Process

A distinguishing feature of the 'Alpha Engine' trading strategy lies in its selective use of funds during position initiation rather than committing the total available capital. This approach closely mirrors real-world trading habits, where traders typically do not stake all their chips at once. Another innovative aspect involves the use of four distinct trading agents that operate at varying DC thresholds. This strategy effectively consolidates performance across different DC thresholds, obviating the need to select an optimal threshold size - a common dilemma as smaller thresholds often promise higher theoretical returns but also carry increased 'noise'. Remarkably, the absence of a stop-loss setting in the 'Alpha Engine' allows for a full manifestation of the inherent characteristics of the DC framework. For these reasons, we utilize the 'Alpha Engine' strategy as a reference to develop our sample DC-TF and DC-CT trading strategies. The specific trading rule entails:

- Set a trading agent with a pre-defined θ_{DC} and profit target PT ($PT = \theta_{DC}$).
- Opening a new position at the beginning of an OS event. The direction of the new position is as same as the current price trend.
- Close the position only when the profit of this position reaches the profit target PT.

To mitigate the risk associated with holding positions without stop-loss protection, we do not allocate the entire capital to a single trading opportunity. Instead, each new agent receives only a small fraction of available capital, and the allocation ratio is reduced when the accumulated exposure becomes excessive.

The profit target PT is defined based on the DC scaling law, which suggests that the expected magnitude of an OS is closely related to the preceding DC threshold. Therefore, we sets the profit target as:

$$PT = \theta_{DC} \tag{3.3}$$

For example, a trading agent with $\theta_{DC} = 1\%$ opens a position at the beginning of an OS event and aims to capture a subsequent price movement of 1% in the same direction. The position is closed only when its accumulated profit reaches this predefined target. Similarly, an agent with $\theta_{DC} = 0.5\%$ closes its position only after achieving a 0.5% return. If the price moves against the position, the agent remains active rather than being closed by a stop-loss rule. Consequently, some positions may experience prolonged holding periods or may fail to reach their profit targets, but the design allows the strategy to fully exploit the statistical characteristics of DC-based price movements.

The pseudocode Algorithm 2 succinctly encapsulates the core logic of DC-TF trading model by detailing the sequential steps of environment initialization, dynamic position evaluation based on profit targets and price deviation, capital availability checks, and strategic trade execution.

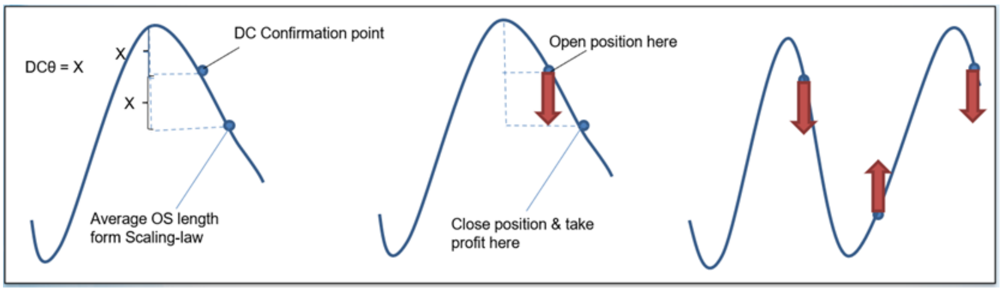

Figure 3.5: Trading rules of Simple trend-following trading strategy. It shows a sample DC-TF trading strategy that opens a position at the DC Confirmation point and closes the position after reaching the predetermined profit target.

Figure 3.6 illustrates the application of the trading rule of the sample DC-TF trading model amidst descending and ascending price trajectories. Within the diagram, the blue curve signifies price fluctuations, while the red arrows demarcate the position initiation and its directionality: downward indicates a short position, whereas upward denotes a long position. Each position is only liquidated upon reaching a pre-specified profit target. The decision to set a take-profit without implementing a stop-loss aims to authentically encapsulate the essence of the DC strategy.

Multiple positions are concurrently executed by the trading agent. These foundational trading directives aid in formulating aCounter-trend trading approach, which exhibits variant dynamism across different price oscillation intervals. The underpinning for these trading directives rests on the scaling laws. Given that the average length of OS events approximates the predefined DC threshold, once a DC event is validated, the market is anticipated to perpetuate along the prevailing trend, presenting an opportune moment to institute a new position.

The theoretical foundation for both the trend-following and Counter-trend strategies is rooted in scaling laws. The primary distinction between these two strategy classifications is the timing of the initiation of new positions. Figure 3.5 exemplifies the execution of trend-following trading rules, with each position closed upon attainment of a pre-determined profit target.

3.2.5 The Simple DC-CT Trading Strategy

In contrast to the sample DC-TF strategy delineated in the preceding section, the sample DC-CT trading approach endorses inverse trading rules, yet they exhibit analogous position management techniques, with their theoretical foundations anchored in scaling laws. The principal divergence between these strategy classifications resides in the timing of new position inception. Figure 3.6 exemplifies the tenets of the DC-CT trading strategy. Consistent with Figure 3.5, the depicted blue trajectory reflects price fluctuations, while the red arrows indicate the positions' initiation and respective orientation. Downward arrows signify short positions, and upward ones, long. Notably, positions contrary to the current trend are only initiated when prices are within the OS event phase and the duration of the current OS event surpasses that of the DC event. When the duration of an OS event surpasses the DC threshold, the market is perceived to be in an overextended phase. At such junctures, the market manifests a pronounced propensity to deviate from the incumbent price trajectory, offering an window of opportunity to instantiate a new position.

The sample DC-CT trading strategies operate on the premise that since the length of an OS event approximates that of a DC event, the culmination of an OS event – hence a likely change in market price direction – can be anticipated when its length reaches or exceeds the fixed value of a DC event (θ_{DC} by design). Thus, establishing a position contrary to the current direction of the OS event when its length surpasses θ_{DC}, and setting a take-profit equal to θ_{DC}, could likely result in a profit. The specific trading rule entails:

- Set a trading agent with a pre-defined θ_{DC} and profit target PT ($PT = \theta_{DC}$).

- Initiation of new positions when the length of an OS event equals the value of θ_{DC}. The position size is 1% of the whole capital and the direction of the new position is inversely proportional to the current price trend.

- The closure of a position occurs solely when the profit from the position meets the profit target PT.

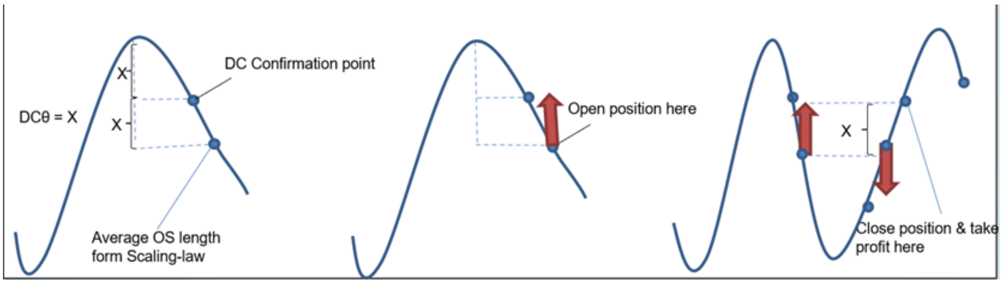

Figure 3.6: Trading rules of Simple Counter-trend trading strategy. It shows a sample DC-CT trading strategy that opens a position when the OS event lasts for a DC threshold length and closes the position after it reaches the predetermined profit target.

The pseudocode Algorithm 3 succinctly encapsulates the core logic of DC-CT trading model by detailing the sequential steps of environment initialization, dynamic position evaluation based on profit targets and price deviation, capital availability checks, and strategic trade execution.

Having delineated both the sample DC-CT and DC-TF trading models, a synthesis emerges: The DC-CT strategy thrives on market overextensions, capitalizing on anticipated reversals, whereas the DC-TF model leverages sustained trends, seeking to ride them for gains. Both methodologies, while operationally distinct, find their underpinnings deeply rooted in scaling laws. Their nuanced approaches to market dynamics exemplify the diverse potentialities inherent within the DC framework. As we venture further, understanding these strategies' congruities and divergences will be pivotal for discerning their respective efficacy in varied market conditions.

This chapter contributes to the DC-based trading literature by setting up two comparable DC-based trading strategies. Unlike previous studies that primarily integrate DC signals with complex trading mechanisms, such as the Alpha Engine, this research separates the two fundamental components of the DC framework and examines their corresponding trading hypotheses: trend continuation following DC confirmation events and mean reversion after extended OS movements. By simplifying the trading mechanism and reducing the influence of additional money management techniques, the proposed strategies enable a clearer evaluation of the intrinsic predictive capability of DC events. Furthermore, this study provides one of the first comprehensive empirical comparisons between DC-based trend-following and counter-trend strategies under a unified framework, offering new insights into how different phases of market movements captured by the DC framework can contribute to algorithmic trading decisions.

Algorithm 3 DC-CT Trading Model Logic

Initialize: Set up trading environment and θ_{DC}; set profit target PT
while price data is available **do**
  if there exist open positions **then**
    for all open positions **do**
      if position profit $\geq PT$ **then**
        Close the position
      end if
    end for
  end if
  if $|P - P_{DCC}| \leq \theta_{DC}$ **then**
    Read next price point ▷ Price has not deviated sufficiently
  else
    Capital Check: Is there available capital?
    if available capital exists **then**
      Open new position with size equal to 1/100 of total capital in the **opposite** trend direction; assign profit target PT
    end if
  end if
end while

3.2.6 DC-based Dynamic Stop-Loss

Stop-loss strategies represent an essential and ubiquitous aspect of risk management in investment practices [144]. In essence, a stop-loss rule prompts the liquidation of an investment position once a predefined loss threshold is breached, thereby serving as a safeguard against escalating losses. The fundamental objective of a stop-loss rule is to confine the potential loss to a manageable range in the event of ill-fated investment decisions. The robustness and practicality of stop-loss rules have contributed to their widespread adoption in actual transactions [158].

However, despite its prevalence, academic investigation into the utilization and implications of stop-loss strategies remains sparse. In particular, a comprehensive examination of the effects of stop-loss rules on the performance of trading strategies is virtually non-existent. One can consult, for instance, the work of Kaminski [144], who studied the impacts of stop-loss orders on the optimal investment strategy in illiquid markets. Chen et al. [38] investigated the effectiveness of dynamic stop-loss limits and stop-gain limits in risk management. More recently, Bernales and Guidolin [159]conducted an empirical analysis of the effects of stop-loss orders on the price dynamics of currency futures contracts. This study endeavors to contribute to the nascent field of research in this area.

The stop-loss strategy is defined as follows. For a given currency pair held as a long position in the original portfolio, the asset is sold once its price falls to a predetermined level. Conversely, in the case of a short position, the position is closed or the currency pair repurchased when the price rises to a specified level. Let this designated level be denoted as the stop threshold S_θ; its mathematical formulation is given by:

$$\begin{cases} \text{for long position, should be closed if} & \frac{P_{\text{now}}-P_i}{P_i} > S_\theta \\ \text{for short position, should be closed if} & \frac{P_i-P_{\text{now}}}{P_i} > S_\theta \end{cases} \tag{3.4}$$

Where P_{now} is the current price, P_i is the price when the position $i-th$ is opened. The existing inventory size for all the agents shows an unavoidable increase which is hard to reduce over time, and which would lead to a huge loss using our trading strategies. In order to meet this challenge, the stop loss system should be added.

The widespread adoption of stop-loss rules in financial markets underscores their practical utility; however, the static nature of conventional stop-loss thresholds has been a subject of scrutiny, particularly in dynamic and volatile trading environments [38,144]. To overcome this limitation, we propose a dynamic stop-loss mechanism based on the Directional Change (DC) concept, whereby the stop-loss threshold is adjusted in accordance with recent market trends rather than being dictated by an arbitrary fixed value S_θ. Given that our trading strategies are inherently based on the DC framework, this approach is particularly well-suited to our methodology.

The DC framework [1,57] provides an alternative paradigm for defining stop-loss strategies through the adaptation to market regime shifts. In contrast to traditional methods that rely on fixed price levels, a DC-based dynamic stop-loss mechanism identifies significant price reversals by capturing intrinsic market fluctuations. Specifically, we design a dynamic DC-based stop-loss value S_{DC} that is contingent on the duration of a position. A position is closed when its holding time exceeds S_{DC}. This design is underpinned by the scaling laws within the DC framework, which statistically suggest that, in the FX market, the duration of OS events is generally twice that of DC events. Consequently, for the DC-TF strategy, an optimal holding period is approximately twice the DC event duration. To encompass the majority of OS events while providing a margin of safety, we add the standard deviation σ_{OS} to obtain the final time-dependent dynamic DC stop-loss threshold. The stop-loss threshold for the DC-CT strategy S_{DC}^{TF} is defined as:

$$S_{DC}^{CT} = 2DC_t + \sigma_{OS}. \tag{3.5}$$

where DC_t denotes the average duration of the past 10 DC events and σ_{OS} represents the standard deviation of the OS event durations.

For the DC-CT strategy, a trend reversal is expected at the time of position opening. Consequently, the maximum holding period is approximately equal to the remaining duration of the upcoming reverse DC-OS event plus the duration of one DC event. According to the scaling laws of the DC framework, the duration of an OS event is roughly twice that of a DC event. Therefore, we use three times the mean duration of a DC event as the baseline and add the standard deviation of OS event duration, σ_{OS} , as a safety margin to account for the majority of situation. Consequently, the stop-loss threshold for the DC-CT strategy S_{DC}^{CT} is defined as

$$S_{DC}^{CT} = 3DC_t + \sigma_{OS}, \tag{3.6}$$

where DC_t denotes the duration of a single DC event.

For a given position, if its holding time T_{holding} exceeds the dynamic threshold, the position is liquidated. The DC-based dynamic Stop-Loss can be expressed mathematically as follows:

$$\begin{cases} \text{Close position if } T_{\text{holding}} > S_{DC}, \\ S_{DC}^{CT} = 2DC_t + \sigma_{OS}. \\ S_{DC}^{CT} = 3DC_t + \sigma_{OS}, \end{cases} \tag{3.7}$$

The dynamic DC-based stop-loss system offers a dual advantage. Firstly, it inherently filters out noise, ensuring that minor fluctuations do not trigger premature liquidations. Secondly, by anchoring stop-loss decisions in market structure rather than arbitrary thresholds, this method mitigates inventory accumulation and reduces prolonged exposure to adverse trends. Despite its potential, academic exploration of dynamic stop-loss mechanisms remains limited. While pioneering studies have examined static stop-loss effects in illiquid markets [144] and the interplay between stop-loss and take-profit strategies [38], integrating directional change principles into stop-loss models represents an emerging research frontier with profound implications for algorithmic trading and portfolio risk management.

3.3 Result and Analysis

In the following discussion, we analyze the potential profitability of DC countertrend and trend-following trading strategies, comparing and evaluating these two conceptual variants alongside the impact of a dynamic DC stop-loss mechanism. This section also outlines the evaluation criteria, metrics, and data sources employed. By adopting an extensive testing framework, we aim to identify the superior model and delineate the specific market conditions under which each strategy excels. First, we provide a concise overview of the experimental workflow described in this chapter, then detail the underlying assumptions of the trading model design, and finally present and analyze the experimental results.

This section delineates and appraises results procured through a methodologically sequenced set of experiments. Initially, trading strategies derived from the scaling laws within the DC framework are first classified into two categories: DC-CT and DC-TF strategies. These strategies were subsequently subjected to empirical testing against the historical FX market dataset spanning 2006 to 2014. The chronological choice was deliberate, aiming to encapsulate the financial turmoil of 2007-2009, thus providing insights into our strategy's resilience under perturbed market conditions. Furthermore, varying values of θ_{DC} (ranging between 0.1% and 2%) were selected to ascertain the respective performances of the two strategies across diverse θ_{DC} thresholds, as presented in Tables 3.2–3.5. We then instantiated trading strategies for four distinct θ_{DC} values, creating multi-agent systems based on both the sample DC-TF and the sample DC-CT trading norms. The rationale for such multi-agent strategies hinged on: 1) simulating traders operating at varied market tiers, and 2) circumventing the quandary of optimal θ_{DC} selection. Conclusively, a stop-loss system was integrated into two multi-agent DC strategies, followed by a subsequent evaluation of their performance.

In the process of our modeling, we posit the following four assumptions:

- Assumption 1: It is presumed that the trading agent commences with the liquidity of \$100,000 units, devoid of any existing positions. This setting amount also satisfies the condition for no market impact.

- Assumption 2: Positional adjustments are considered non-feasible.

- Assumption 3: Positions are initiated exclusively through market orders or limit orders.

- Assumption 4: Implicit transaction costs in the market are disregarded.

3.3.1 Research Data

Pardo et al. [52] emphasized the importance of backtesting in the evaluation of trading strategies (for a detailed definition, see Section 2.2.4). In this chapter, we introduce the dataset used to validate the effectiveness of trading strategies. Given the demonstrated statistical validity of the DC framework in the Forex market [53], we selected the Forex market for backtesting. The dataset used in this study consists of minute-level Forex market data. The analysis covers eight currency pairs—AUD/NZD, EUR/GBP, EUR/JPY, EUR/USD, USD/CAD, USD/CHF, USD/JPY, and NZD/JPY. The midprices of these currency pairs were sampled every minute over 96 months, from January 2006 to December 2014, excluding holidays and weekends. The specific data structure and the method for converting the dataset into an event sequence based on the DC framework are detailed in Section 2.5.

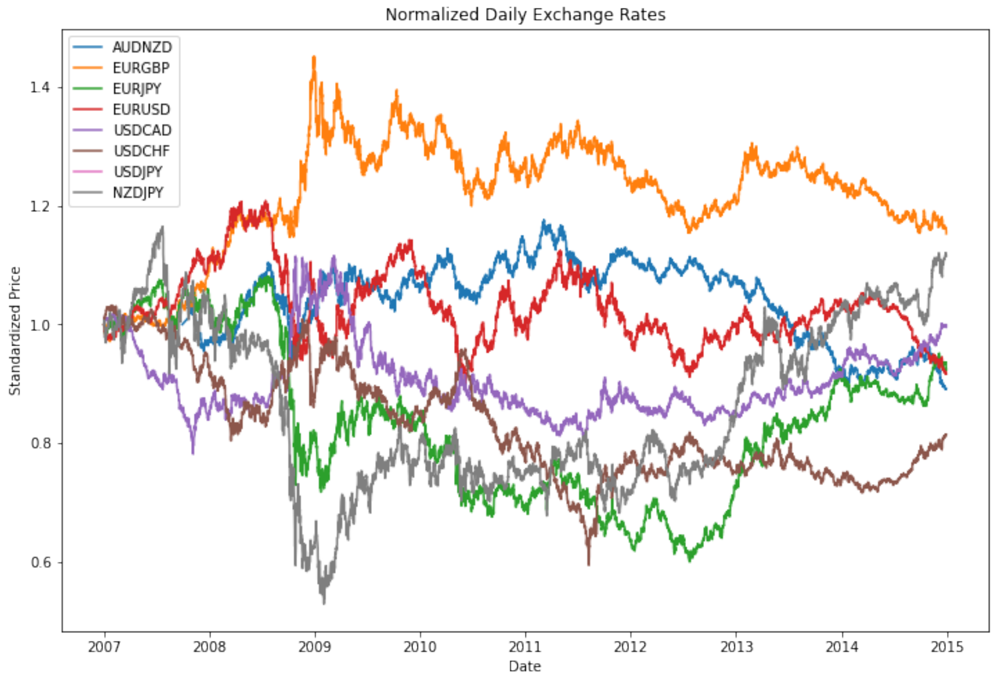

Figure 3.7: Normalized daily exchange rates.

The selection of this dataset is based on three key considerations. First, it enhances the stability of the DC strategy. Previous studies have demonstrated the effectiveness of the DC framework in the FX market, suggesting that testing within this market yields more robust results. Second, it ensures the reliability of the testing data. The high liquidity of the FX market minimizes the impact of certain trading costs, such as slippage and spreads, and its near 24-hour trading—excluding holidays—facilitates standardized data processing [16]. Third, it mitigates the risk of overfitting. From [52], price trend shifts in price movements significantly affect trading strategy performance. Therefore, we selected multiple currency pairs to capture diverse trends and avoid bias associated with a single market condition. Figure 3.7 illustrates the normalized daily exchange rates of the eight selected currency pairs over the test period (January 1, 2006, to December 31, 2014). The trend variations (Figure 3.7) confirm that the experimental setup accounts for different market conditions, minimizing potential trend-related biases.

3.3.2 Evaluation Indicators

The primary objective of this investigation is to assess the efficacy of the DC-CT and DC-TF strategies. To evaluate the performance of these DC-based trading strategies, we need to measure profitability and risk using certain dimensions. To commence, annual returns, the Maximum Drawdown (MDD) and sharp rate during the strategy's operation serve as our evaluation metrics. It thus becomes imperative first to delineate precise definitions for annual returns and MDD. For a stipulated time frame extending from T_0 to T_t, returns can be mathematically articulated as:

$$ROR = \frac{C_t + P\&L}{C_o} \tag{3.8}$$

In this equation, *ROR* represents the rate of return post-operating the trading agent, spanning from the onset at T_0 to culmination at T_t. C_o denotes the predefined initial capital, while C_t symbolizes the capital in possession of the agent at the time instance T_t, with P&L indicating the profit and loss from held positions. It's pertinent to note that the proposed equation does not account for transactional costs; variants inclusive of these costs will be elaborated upon in subsequent sections. Given the trading modality adopted in this study, characterized by frequent position openings and closings, the profit or loss of the trading model at instance t bifurcates into two components: current cash holdings and unrealized gains from positions.

In our investigation, the MDD emerges as a pivotal evaluation metric. While the Rate of Return (ROR) proficiently quantifies the eventual performance of a model, it remains somewhat myopic in capturing intrinsic market volatilities. Addressing this limitation and furnishing a more encompassing portrayal of strategy-associated risk, we harness the MDD as a metric. Serving as a risk indicator, MDD quantitatively delineates the most substantial decline an asset experiences from its apex to its nadir, thereby shedding light on the strategy's susceptibility to pronounced downturns. Mathematically, MDD is articulated as:

$$MDD = \max_{t_1 > t_2} \frac{R_{t_1} - R_{t_2}}{R_{t_1}} \tag{3.9}$$

where t_1 and t_2 are the points of the return of peak and through. R_{t_1} is the return of peak and R_{t_1} is the return of through. It is measured by calculating the maximum observed loss from a peak value of return to a trough before a new peak is reached.

The Sharpe ratio, introduced by William F. Sharpe in 1966 [29], is a measure for calculating risk-adjusted return, and has become one of the industry standards for such calculations. It is particularly useful in comparing the relative performance of different assets or portfolios, taking into account not only the potential returns but also the variability and potential risk involved in those returns. In this chapter, the Sharpe ratio is used as a supplementary evaluation of our strategies.

The Sharpe ratio is defined mathematically as:

$$Sr = \frac{R_p - R_f}{\sigma_p} \tag{3.10}$$

where S is the Sharpe ratio. R_p is the expected return of the portfolio. R_f is the risk-free rate. σ_p is the standard deviation of the portfolio's excess return. By subtracting the risk-free rate from the expected return of the portfolio, we obtain the excess return, which is then divided by the standard deviation of the portfolio's excess return.

3.3.3 Result of Two Opposite Trading Strategy Comparison

This Section offers a comparative performance evaluation of our sample DC-CT and DC-TF trading model. The effectiveness and consistency of the DC-based trading models are underscored through this analysis. In the conducted analysis, four distinct tables present pivotal metrics relevant to the performance of the DC-CT and DC-TF strategies across eight currency pairs. Table 3.2 showcases the annualized returns for the DC-CT strategy, operating with a θ_{DC} range of 0.1% to 1.0%. In a parallel manner, Table 3.3 illuminates the performance outcomes when the θ_{DC} values for the DC-CT strategy extend from 1.1% to 2.0%. Transitioning to the DC-TF strategy, Table 3.4 illustrates its efficiency by representing the annualized returns with a θ_{DC} spectrum of 0.1% to 1.0%. Lastly, Table 3.5 depicts the corresponding results for the DC-TF strategy, but with the θ_{DC} parameters spanning from 1.1% to 2.0%. Collectively, these tables serve as a comprehensive testament to the adaptability and efficacy of both strategies under varied θ_{DC} settings.

$\theta_{DC}(\%)$	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
AUDNZD	3.02%	2.79%	2.38%	1.69%	1.18%	1.02%	0.658%	0.540%	0.369%	0.333%
EURGBP	1.37%	1.23%	1.25%	0.675%	0.712%	0.422%	0.431%	0.504%	0.264%	0.335%
EURJPY	3.42%	3.61%	3.88%	3.18%	2.59%	2.45%	1.90%	1.68%	1.56%	1.30%
EURUSD	2.53%	2.62%	3.18%	2.33%	1.91%	1.60%	1.39%	1.19%	1.10%	1.05%
NZDJPY	2.62%	3.06%	3.32%	2.53%	2.26%	2.17%	1.94%	1.65%	1.46%	1.29%
USDCAD	2.04%	1.57%	1.69%	1.54%	1.37%	0.933%	0.913%	0.754%	0.617%	0.596%
USDCHF	0.26%	-0.45%	-0.30%	-0.48%	-0.25%	-0.45%	-0.286%	-0.204%	-0.203%	-0.290%
USDJPY	2.86%	2.09%	2.28%	2.06%	1.69%	1.43%	1.23%	1.14%	0.990%	0.853%

Table 3.2: Annualized ROR for sample DC-CT on Various Pairs (0.01 to 0.10)

$\theta_{DC}(\%)$	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
AUDNZD	0.33%	0.32%	0.27%	0.23%	0.26%	0.22%	0.19%	0.24%	0.12%	0.17%
EURGBP	0.35%	0.36%	0.28%	0.30%	0.29%	0.27%	0.25%	0.24%	0.22%	0.23%
EURJPY	0.12%	0.14%	0.14%	0.12%	0.12%	0.11%	0.11%	0.10%	0.10%	0.10%
EURUSD	0.87%	0.68%	0.75%	0.67%	0.60%	0.52%	0.44%	0.38%	0.38%	0.37%
NZDJPY	1.19%	1.03%	1.00%	0.89%	0.77%	0.75%	0.69%	0.66%	0.60%	0.72%
USDCAD	0.58%	0.54%	0.48%	0.43%	0.49%	0.36%	0.34%	0.35%	0.27%	0.25%
USDCHF	-0.33%	0.38%	-0.34%	-0.25%	-0.26%	0.09%	-0.28%	-0.20%	-0.13%	-0.07%
USDJPY	0.80%	0.78%	0.66%	0.56%	0.47%	0.36%	0.34%	0.35%	0.26%	0.26%

Table 3.3: Annualized ROR for sample DC-CT on Various Pairs (0.11 to 0.20)

$\theta_{DC}(\%)$	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
AUDNZD	3.48%	3.27%	2.48%	2.17%	1.21%	1.36%	0.683%	0.774%	0.404%	0.370%
EURGBP	1.48%	1.44%	1.38%	0.962%	0.766%	0.643%	0.698%	0.728%	0.531%	0.266%
EURJPY	4.15%	4.14%	4.43%	3.87%	2.77%	2.70%	2.20%	1.89%	1.62%	1.71%
EURUSD	3.02%	2.67%	3.63%	2.63%	2.09%	1.65%	1.35%	1.31%	1.06%	1.03%
NZDJPY	2.92%	2.96%	3.11%	2.71%	2.61%	2.49%	2.46%	1.69%	1.71%	1.45%
USDCAD	2.17%	1.62%	1.88%	1.68%	1.44%	1.11%	1.19%	0.812%	0.804%	0.889%
USDCHF	0.258%	-0.411%	-0.292%	-0.513%	-0.0904%	-0.476%	-0.0820%	-0.121%	-0.172%	-0.339%
USDJPY	3.34%	2.50%	2.23%	2.41%	2.13%	1.42%	1.30%	1.40%	1.23%	1.08%

Table 3.4: Annualized ROR for sample DC-TF on Various Pairs (0.01 to 0.10)

θ_{DC}(%)	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
AUDNZD	0.428%	0.292%	0.416%	0.131%	0.423%	0.326%	0.393%	0.529%	0.0574%	0.432%
EURGBP	0.380%	0.678%	0.233%	0.519%	0.602%	0.416%	0.554%	0.528%	0.516%	0.223%
EURJPY	0.0336%	0.151%	0.263%	0.289%	0.370%	0.341%	0.226%	0.239%	0.309%	0.148%
EURUSD	0.961%	0.892%	1.018%	0.786%	0.713%	0.758%	0.567%	0.390%	0.487%	0.416%
NZDJPY	1.394%	1.313%	1.296%	1.094%	1.007%	0.830%	1.029%	0.952%	0.823%	0.919%
USDCAD	0.862%	0.648%	0.501%	0.355%	0.504%	0.509%	0.624%	0.421%	0.418%	0.168%
USDCHF	-0.281%	0.321%	-0.221%	-0.186%	0.0174%	0.197%	-0.116%	-0.321%	0.00560%	0.127%
USDJPY	0.822%	1.055%	0.714%	0.495%	0.761%	0.617%	0.451%	0.459%	0.413%	0.466%

Table 3.5: Annualized ROR for sample DC-TF on Various Pairs (0.11 to 0.20)

Upon evaluating the annualized Rate of Return (ROR) for both the DC-CT and DC-TF strategies across different currency pairs, several key insights emerge. In most scenarios, the DC-TF strategy tends to outperform the DC-CT strategy. This observation holds particularly true for currency pairs such as EURJPY, EURUSD, and NZDJPY, especially in the θ_{DC} range from 0.1% to 1.0%. As θ_{DC} increases from 0.1 to 2.0, there's a discernible decline in the annualized ROR for both strategies across almost all currency pairs. The sharper decline in the DC-CT strategy as compared to the DC-TF strategy further establishes the superiority of the latter. Notably, the USDCHF currency pair exhibited negative RORs for certain θ_{DC} values, especially in the DC-CT strategy. This suggests that the strategy may not be optimal for this specific currency pair or demands a more meticulous parameter tuning.The DC-TF strategy not only yielded higher RORs but also showcased a more stable performance as θ_{DC} varied. This characteristic is indicative of a potentially more robust and resilient trading strategy. In conclusion, from the point of view of ROR, while both the DC-CT and DC-TF strategies offer varying returns across different currency pairs and θ_{DC} values, the DC-TF strategy consistently stands out as the more advantageous choice in the analyzed scenarios. However, a deeper dive into other risk metrics, strategy drawdown, and transaction costs might be essential to ascertain a holistic view of strategy performance.

To delve deeper into the comparative efficiencies of the DC-CT and DC-TF strategies, we employed sample DC-CL and DC-TF frameworks. These integrated trading agents are characterized by four distinct θ_{DC} values, leading to the formation of two multi-agent models grounded in DC-CT and DC-TF. Figure 3.8 elucidates the yield curves over the span from early 2006 to early 2014 utilizing these multi-agent models. Table 3.6 presents the trading outcomes across diverse currency pairs. Diverging from the adjusted annualized returns documented in Tables 3.2-3.5, the results herein represent an aggregate yield over eight years. To quantify the associated risks, we further furnish data on the MDD and the Sharpe ratio.

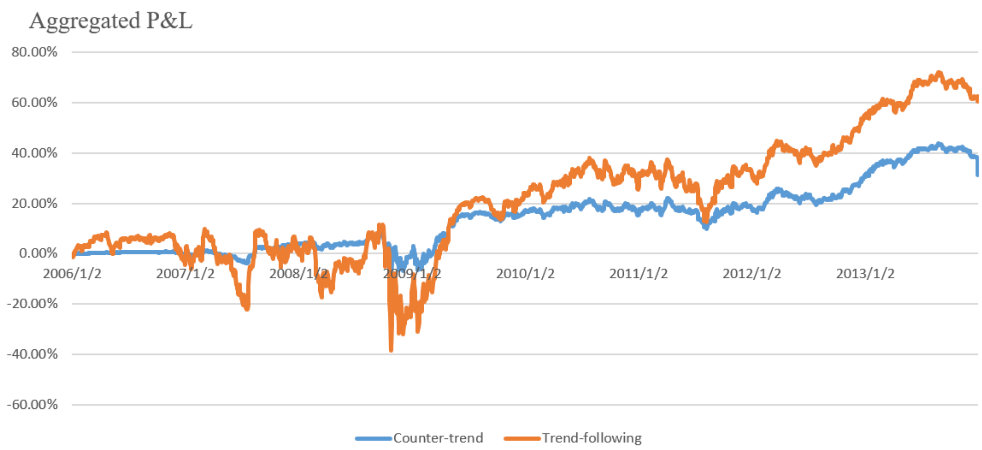

Figure 3.8: Un-levered aggregated Profit & Loss(P&L) of the DC-based simple Counter-trend strategy and simple Trend-following strategy, across 8 currency pairs, for eight years. See details in Appendix.

The analysis, amalgamating data from eight diverse currency pairs, indicates that the simple trend-following strategy exhibits superior profitability during the selected period. However, it also results in an elevated drawdown, calculated on a daily basis. Throughout an eight-year duration, the trend-following strategy outperforms the counter-trend approach, yielding a higher overall rate of return. An examination of return curves discloses a consistent frequency in profit and loss fluctuations, despite the diverging rules governing position openings. This consistency may originate from the scaling laws underpinning both strategies.

The trend-following strategy, on one side, evinces substantial potential profitability, warranting further consideration—it is approximately 2.7 times more profitable than the Counter-trend strategy. Conversely, it also reveals a more severe mean MDD, with the maximum MDD nearing 75.47%. This level of loss can often be considered unacceptable in a trading strategy, as presented in Table 3.6.

	DC-TF Strategy			DC-CT Strategy		
	P&L (%)	MDD (%)	Sharp Rate	P&L (%)	MDD (%)	Sharp Rate
AUD/NZD	45.37	20.37	1.71	15.99	5.26	1.83
EUR/GBP	15.46	30.58	0.36	4.96	15.29	-0.14
EUR/JPY	72.92	32.52	1.06	24.41	11.77	1.16
EUR/USD	65.93	8.45	0.85	25.63	5.76	0.78
NZD/JPY	23.10	38.73	0.98	-0.37	10.98	1.01
USD/CAD	50.01	22.53	1.08	19.45	9.15	0.99
USD/CHF	10.67	75.47	0.59	4.94	26.19	0.81
USD/JPY	34.45	23.26	0.04	10.72	8.96	1.15
Average	39.74	31.49	0.83	13.22	11.67	0.95
Std. Dev.	22.88	19.97	0.51	9.66	6.70	0.55

Table 3.6: Comparison of trend-following and Counter-trend strategies

3.3.4 Statistical Analysis of DC-based Strategy Performance

Although the previous sections demonstrate the performance of DC-based trading strategies through descriptive results, statistical analysis is further conducted to examine whether the observed performance differences are robust across different market conditions and parameter settings. Since each strategy is evaluated on the same set of currency pairs under different θ_{DC} values, the obtained results form paired observations rather than independent samples. Therefore, this section focuses on two aspects: (1) the robustness of strategy performance under different θ_{DC} settings, and (2) the statistical significance of the performance difference between DC-TF and DC-CT strategies.

Robustness Analysis under Different θ_{DC} Values

The DC framework relies on the predefined threshold θ_{DC} to identify market events. Therefore, it is important to investigate whether the profitability of DC-based strategies is sensitive to the selection of θ_{DC}. The annualized ROR results under different θ_{DC} settings are summarized in Tables 3.2, 3.3, 3.4, and 3.5.

For each strategy, the mean return and standard deviation across all currency pairs and θ_{DC} values are calculated. Given a set of observations $R_1, R_2, ..., R_N$, the mean return is defined as:

$$\bar{R} = \frac{1}{N}\sum_{i=1}^{N} R_i \tag{3.11}$$

and the standard deviation is:

$$\sigma_R = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(R_i - \bar{R})^2} \tag{3.12}$$

where N represents the total number of observations. A lower standard deviation indicates that the strategy performance is less sensitive to different θ_{DC} selections.

The results show that both DC-TF and DC-CT strategies generate positive returns across most θ_{DC} settings, suggesting that the profitability of DC-based strategies is not dependent on a single threshold value. For example, at $\theta_{DC} = 0.1\%$, the cross-pair average annualized ROR reaches approximately 2.27% for DC-CT and 2.61% for DC-TF. At $\theta_{DC} = 1.0\%$, these averages decline to approximately 0.68% and 0.81%, respectively, and further decrease to approximately 0.25% and 0.36% when $\theta_{DC} = 2.0\%$. This trend is consistent with the characteristics of the DC framework, where larger thresholds identify fewer directional-change events and consequently generate fewer trading opportunities.

In addition, substantial heterogeneity can be observed across currency pairs. EURJPY and NZDJPY consistently achieve relatively high returns under both strategies, whereas USDCHF exhibits persistently low or negative returns across most θ_{DC} settings. For instance, at $\theta_{DC} = 0.2\%$, USDCHF produces annualized ROR values of -0.45% under DC-CT and -0.411% under DC-TF. These findings suggest that the effectiveness of DC-based trading strategies depends not only on the choice of θ_{DC} but also on the underlying characteristics of individual currency pairs.

Overall, despite the gradual decline in profitability as θ_{DC} increases, both strategies remain profitable across a wide range of parameter settings. This observation indicates that the proposed DC-based trading mechanism exhibits a reasonable degree of robustness and is not overly dependent on a specific threshold configuration.

Statistical Comparison between DC-TF and DC-CT Strategies

To further examine whether the performance difference between DC-TF and DC-CT strategies is statistically significant, a paired sample t-test is conducted. The paired test is appropriate because both strategies are evaluated using identical currency pairs and identical θ_{DC} settings.

Each observation represents the annualized ROR obtained from one specific combination of a currency pair and a θ_{DC} value. Consequently, the complete dataset contains 160 paired observations, corresponding to 8 currency pairs evaluated under 20 different θ_{DC} settings.

The difference between the two strategies for each paired observation is defined as:

$$D_i = R_i^{TF} - R_i^{CT} \tag{3.13}$$

where R_i^{TF} and R_i^{CT} denote the annualized ROR of the DC-TF and DC-CT strategies, respectively.

The null hypothesis assumes that there is no difference between the expected returns of the two strategies:

$$H_0 : \mu_D = 0 \tag{3.14}$$

while the alternative hypothesis is:

$$H_1 : \mu_D \neq 0 \tag{3.15}$$

The t-statistic is calculated as:

$$t = \frac{\bar{D}}{s_D/\sqrt{N}} \tag{3.16}$$

where $\bar{D}$ is the mean of the paired differences, s_D is the standard deviation of the paired differences, and N is the number of paired observations.

Inspection of Tables 3.2–3.5 shows that DC-TF generally achieves higher annualized ROR values than DC-CT across most currency pairs and threshold settings. The performance difference is particularly evident at relatively small θ_{DC} values and gradually decreases as θ_{DC} increases. To determine whether this observed advantage is statistically significant, the paired t-test is performed using all 160 paired observations.

Table 3.7: Paired t-test Results for DC-TF and DC-CT

Comparison	N	Mean Difference	t-statistic	p-value
DC-TF vs DC-CT	160	0.167	13.12	< 0.0001

The resulting t-statistic and corresponding p-value are reported in Table 3.7. These results provide statistical evidence regarding whether the superior performance of DC-TF can be attributed to systematic differences between the two trading strategies rather than random variation arising from specific currency pairs or parameter selections. A statistically significant result would suggest that the paired t-test yields a t-statistic of 13.12 with a p-value below 0.0001. Therefore, the null hypothesis is rejected at the 5% significance level, indicating that DC-TF achieves significantly higher annualized returns than DC-CT across the tested currency pairs and θ_{DC} settings.Although the average performance advantage is relatively modest (0.167% annualized ROR per observation), the consistency of the difference across 160 paired observations suggests that the superior performance of DC-TF is systematic rather than being driven by a small number of exceptional cases.

3.3.5 Result of the DC-CT and DC-TF Strategies with the DC-based Dynamic Stop-Loss

In the previous subsection, we reviewed the profitability and risks associated with both the DC-TF and DC-CT strategies. In this chapter, we also explore the application of a DC-based stop-loss system. This investigation aims to reduce risk while maintaining the profitability of the DC strategy. Under conditions of relatively smooth market volatility, DC and OS events alternate with roughly equal durations, allowing trading agents to consistently secure returns. However, in real market scenarios, the durations of most OS events tend to deviate, which may lead agents to accumulate substantial positions accompanied by significant unrealized paper losses, particularly when encountering abnormal market trends contrary to their positions. To address this challenge, a dynamic stop-loss system based on the DC framework—derived from DC scaling laws—is introduced.

In accordance with the rules outlined in Section 3.2.6, we augmented both the sample DC-TF and DC-CT strategies with a DC-based dynamic stop-loss system. In this framework, each opening position is accompanied not only by a take-profit target but also by a time-dependent stop-loss. We then re-ran these enhanced strategies over a selected period on eight currency pairs, analyzing the performance on each. The parameters required for the DC-based dynamic stop-loss system for each currency pair were computed and are detailed in Table 3.8.

Table 3.8: DC-based dynamic stop-loss system parameters (Unit:hours

Currency Pair	Threshold 0.25		Threshold 0.5		Threshold 1.0		Threshold 1.5	
	DC_t	σ_{OS}	DC_t	σ_{OS}	DC_t	σ_{OS}	DC_t	σ_{OS}
AUDNZD	5.30	16.30	24.10	51.79	116.14	222.67	277.26	430.16
EURGBP	10.17	25.05	37.99	75.38	129.91	254.05	286.37	490.74
EURJPY	4.11	13.78	15.08	36.49	51.66	107.19	116.80	312.78
EURUSD	7.22	20.10	26.00	55.28	93.41	173.86	213.23	410.80
NZDJPY	2.41	9.28	7.90	21.55	28.36	90.90	58.32	129.11
USDCAD	6.30	19.81	23.28	51.43	86.89	164.85	192.27	388.16
USDCHF	6.14	17.25	23.43	68.78	83.21	156.86	193.75	315.08
USDJPY	6.07	18.18	21.57	47.24	80.09	143.54	165.92	299.34

The table below summarizes the parameters of the DC-based dynamic stop-loss system for eight currency pairs under four different thresholds (0.25, 0.5, 1.0, and 1.5), with all values expressed in hours. Here, *Threshold x* indicates the employed θ_{DC}; DC_t represents the mean duration of a DC event at the current DC threshold, and σ_{OS} denotes the standard deviation of OS events under the current θ_{DC}. Based on these data, the DC-based dynamic stop-loss settings for each currency pair can be determined for each θ_{DC}.

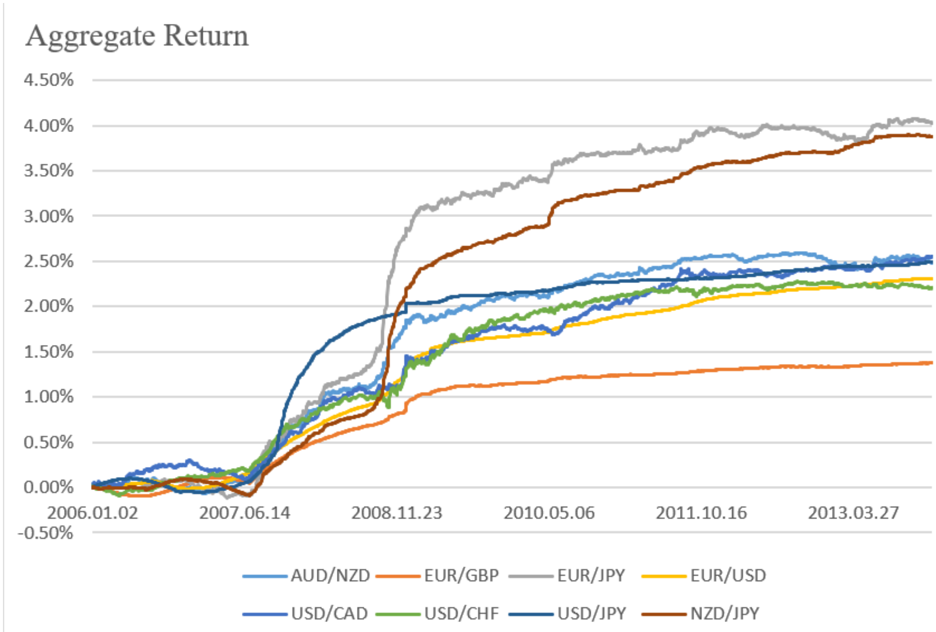

Figure 3.9: Un-levered daily Profit and Loss of the DC-based sample trend Following trading model with the DC-based dynamic Stop-Loss.

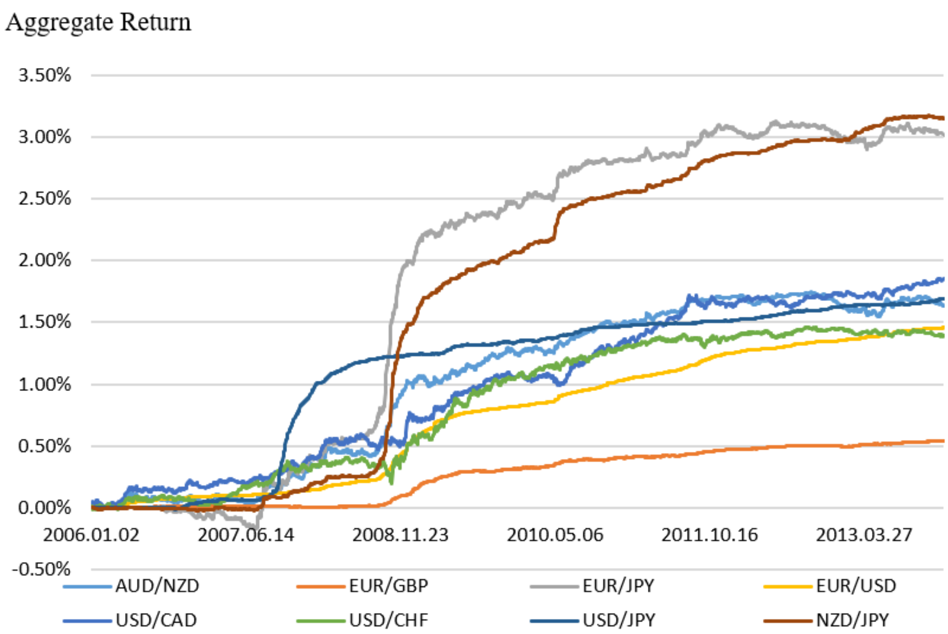

Figure 3.10: Un-levered daily Profit and Loss of the DC-based sample Counter-trend trading model with the DC-based dynamic Stop-Loss.

Figure 3.9 and Figure 3.10 respectively illustrate the cumulative returns on eight currency pairs over the testing period for the trading models that integrate the sample DC-TF strategy and DC-CT strategy with the DC-based dynamic stop-loss system. These figures reveal the impact of the DC-based dynamic stop-loss system on investor returns within the test period. The implementation of the stop-loss mechanism smooths the return curves of the DC contrarian trading model and significantly reduces drawdowns. The reduction in the volatility of the profit-loss curve also implies a lower level of strategic risk. In Table 3.9, we summarize the performance of the DC-TF and DC-CT strategies when combined with the DC-based dynamic stop-loss system across eight currency pairs. For each pair, the table reports the P&L, Sharpe ratio, and MMD. On average, the DC-TF strategy yields a higher P&L of 2.67 compared to 1.84 for the DC-CT strategy. Similarly, the average Sharpe ratio for the DC-TF strategy is 3.61, which is greater than the 3.09 observed under the DC-CT strategy. Conversely, the average MMD for DC-TF is 0.17, whereas the DC-CT strategy demonstrates a lower average MMD of 0.10. These results suggest that, while the dynamic stop-loss system effectively smooths the equity curves and reduces drawdowns, it also results in a trade-off where the overall profitability, particularly under the DC-CT strategy, is compromised. The higher Sharpe ratios associated with the DC-TF strategy indicate better risk-adjusted performance, highlighting the potential benefit of employing the DC-based dynamic stop-loss mechanism to mitigate risk.

Table 3.9: Comparison of DC-TF and DC-CT Strategies with Stop-loss

	DC-TF with Stop-loss			DC-CT with Stop-loss		
Currency Pair	P&L	Sharpe Ratio	MMD	P&L	Sharpe Ratio	MMD
AUD/NZD	2.48	2.16	0.19	1.63	1.47	0.19
EUR/GBP	1.38	4.03	0.12	0.54	3.02	0.02
EUR/JPY	4.03	2.69	0.21	3.02	2.08	0.23
EUR/USD	2.31	6.46	0.09	1.46	6.90	0.00
USD/CAD	2.55	2.16	0.21	1.85	1.61	0.13
USD/CHF	2.21	1.92	0.17	1.39	1.25	0.21
USD/JPY	2.50	4.69	0.16	1.69	4.20	0.01
NZD/JPY	3.88	4.75	0.18	3.15	4.16	0.03
Average	**2.67**	**3.61**	**0.17**	**1.84**	**3.09**	**0.10**

In the pure DC strategy scenario(Table 3.6), the DC-TF strategy achieves an average cumulative profit (P&L) of 39.74%, with an average MDD of 31.49% and an average Sharpe ratio of 0.83. In contrast, the DC-CT strategy generates a considerably lower average profit of 13.22%, accompanied by a significantly lower MDD of 11.67% and a slightly higher average Sharpe ratio of 0.95. These figures indicate that while the DC-TF strategy has a much higher profit potential, it also exhibits substantially greater volatility and risk, as evidenced by its larger drawdowns and lower risk-adjusted returns.

When the DC-based dynamic stop-loss system is integrated into these strategies, the performance profile changes markedly. The dynamic stop-loss mechanism functions by imposing a time-dependent risk control rule at the moment of position opening, effectively setting both a profit target and a stop-loss level. This addition is designed to mitigate adverse price movements and reduce drawdowns, albeit at the cost of overall profitability. For example, under the stop-loss system, the DC-TF strategy's average profit is reduced to 2.67%—a significant decrease from 39.74%—yet the MMD drops drastically to 0.17%, and the Sharpe ratio improves substantially to 3.61. Similarly, the DC-CT strategy sees its average profit decline to 1.84% from 13.22%, while its average MMD decreases to 0.10% and its Sharpe ratio increases to 3.09.

The dramatic reduction in drawdown and the marked improvement in Sharpe ratios suggest that the DC-based dynamic stop-loss system effectively smooths the equity curves and minimizes downside risk. This is particularly important in volatile market conditions, where large drawdowns can erode capital and investor confidence. The enhanced risk-adjusted performance, as reflected by the higher Sharpe ratios, underscores the system's efficacy in controlling risk, despite the trade-off of reduced raw returns. In essence, the stop-loss mechanism serves to shield the portfolio from extreme losses by dynamically adjusting the exit point based on market conditions, thereby preserving capital even if some profit opportunities are curtailed.

Overall, the comparative analysis demonstrates that the pure DC-TF strategy inherently offers higher potential returns than the DC-CT strategy; however, it does so with increased volatility and risk. The incorporation of the DC-based dynamic stop-loss system significantly reduces risk exposure by lowering MMD and enhancing risk-adjusted returns for both strategies. This improvement is a clear testament to the effectiveness of the DC framework when applied to dynamic stop-loss mechanisms. In practical terms, while investors may sacrifice some profit potential, the dynamic stop-loss system provides a more robust and resilient trading model, which is especially valuable during periods of market turbulence.

3.4 Conclusions

In conclusion, this chapter has explored the application of the DC framework in characterizing price fluctuations and its subsequent utilization in trading strategies. Based on the scaling laws derived from the DC framework, we developed two distinct strategies: the DC-TF (trend-following) strategy and the DC-CT (contrarian) strategy. Both strategies were backtested across eight currency pairs, and empirical evidence clearly indicates that the trend-following approach offers substantially higher profit potential relative to the contrarian method, albeit at the cost of increased risk (as reflected by a higher MMD). Notably, both strategies demonstrated profitability over long-term horizons across almost all of the studied currency pairs, underscoring the promise of the DC framework as a predictive model for enhancing financial performance. Furthermore, this study provides a systematic empirical comparison between DC-based trend-following and counter-trend strategies under a unified framework, offering new insights into the distinct roles of DC and OS events in algorithmic trading.

Building on this foundation, our study further extends to the investigation of stop-loss systems within the DC framework. We proposed a dynamic stop-loss model based on the DC framework and integrated it with both the DC-TF and DC-CT strategies. The empirical results reveal that an increase in market volatility enhances the efficacy of the DC-based dynamic stop-loss model in improving overall model performance. Although the implementation of stop-loss rules inevitably curtails the maximum profit potential of these DC-based trading models, it substantially reduces the risk of significant financial losses. Our findings corroborate the notion that, as volatility increases, the positive impact of a stop-loss strategy on risk-adjusted returns becomes more pronounced.

This research culminated in the development of two trading strategies—namely, a DC-based trend-following strategy and a DC-based contrarian strategy—with a rigorous evaluation of their respective strengths and weaknesses. The DC-based dynamic stop-loss system was introduced as a risk mitigation measure and has been shown to effectively manage downside risk, albeit with a trade-off in terms of reduced trading returns. Future investigations will explore the replacement of rigid stop-loss systems with more flexible machine learning algorithms to further enhance adaptability and effectiveness.

Furthermore, existing research on the DC framework has predominantly focused on the FX market. The potential of the DC framework to serve as an analytical tool in other financial markets, such as equities and commodities, warrants further detailed investigation. In addition, our exploration of stop-loss mechanisms within the DC framework has opened a promising new avenue for research. We contend that the full potential of the DC framework in developing robust trading models has yet to be fully exploited. Hence, it is crucial to identify and test alternative methodologies that can be integrated into the broad template of the current DC-based framework, thereby expanding its applicability and utility across diverse market conditions.

Chapter 4

Machine Learning to Manage Risk in the DC-based Trend-following Strategies

The Directional Change (DC) framework provides an event-based representation of financial market dynamics by decomposing price movements into alternating DC and Overshoot (OS) events. Building upon this framework, this chapter investigates whether machine learning (ML) algorithms can improve the predictive power and trading performance of DC-based strategies.

Specifically, the prediction task is formulated as a binary classification problem. Given the characteristics of recent DC and OS events, the model predicts whether the next OS event will exceed a predefined DC threshold and therefore generate a tradable overshoot opportunity. Rather than forecasting general market trends, asset prices, or market volatility, the proposed models focus exclusively on predicting the occurrence of DC-defined overshoot events in the foreign exchange (FX) market.

To evaluate the effectiveness of different ML approaches, five predictive models—Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN)—are developed and integrated into DC-based trading strategies. The resulting trading performance is assessed using eight currency pairs in the FX market. The empirical results demonstrate that ML models can effectively exploit information embedded in DC events and improve the profitability of DC-based trading strategies by enhancing the prediction of overshoot-event outcomes.

4.1 Introduction

Accurately predicting market price trends and developing appropriate trading strategies based on these predictions have long been key concerns for market participants in the financial industry. Both academics and practitioners continuously seek innovative methods to more precisely forecast financial market price movements and propose better trading strategies. The Directional Change (DC) method offers a novel data sampling approach by dividing traditional time-series market price data into alternating intrinsic time events. The primary goal of research deploying the DC framework is to create innovative algorithmic trading strategies by rigorously evaluating existing price prediction methods.

Dupuis [64] proposed a counter-trend trading strategy based on the DC paradigm, leveraging the scaling laws of the FX market. Gypteau [70] introduced a strategy aimed at predicting the daily closing prices of financial markets. Additionally, Bakhach et al. [74] developed a DC-based model to predict the future direction of price trends, which was later extended by Bakhach et al. [71] into a counter-trend trading strategy called "TSFDC." In another study, Kampouridis and Otero [144] proposed a strategy known as "DC + GA," which uses a series of DC (θ_{DC}) indicators with variable weights to predict future price trends. Meanwhile, Tsang et al. [65] introduced a set of market indicators within the DC framework, arguing for their effectiveness in mapping price movements in the FX market. These studies underscore the potential of the DC framework in developing profitable trading strategies and uncovering new patterns in financial markets. Golub et al. [62] devised a profitable automated trading model titled the 'Alpha Engine'. It not only formulated trading rules according to the scaling laws of the DC Framework but also incorporated a trading position management system for rule-setting. Aligning with this focal area, our research aims to explore innovative trading models by integrating the DC framework with machine learning algorithms.

This study seeks to enhance the profitability of DC-based trading strategies by leveraging the decision-support capabilities of machine learning. Given the vast amounts of financial data, its precise historical records, and its quantifiable nature, the financial sector is well-suited for comprehensive data analysis. Machine learning, with numerous successful applications, remains one of the most promising investigative tools in finance. By utilizing the scaling laws of the DC framework, we designed a binary classification model based on the duration of intrinsic time events to predict the strength of future market fluctuations. We employed five widely used machine learning algorithms to manage this model: Random Forest, Support Vector Machine, Artificial Neural Network, Long Short-Term Memory, and Convolutional Neural Network. The primary models were implemented using renowned open-source platforms such as TensorFlow and Scikit-learn. Consistent with most DC-based studies, our research data is derived from the FX market.

The remainder of this chapter is organized as follows: Section 4.2 briefly describes the DC event method and introduces trend-following strategies based on the DC framework. Section 4.3 outlines the mainstream machine learning algorithms (MLAs) used in this study and provides an overview of the methods for enhancing DC-based trading strategy profitability with these ML techniques. Section 4.4 discusses the performance of simple DC-based trend-following strategies and their combinations with machine learning algorithms. Finally, Section 4.5 concludes the study and outlines directions for future research.

4.2 Directional Change Framework and the DC-based Trading strategies

This subsection provides a concise introduction to the core concept of the Directional Change framework, as well as the extended concepts derived from it that are employed in this study. At its core, the DC framework defies the constraints of traditional time series analysis. Through its dynamic manipulation of θ_{DC} values, it unveils layers of granularity in DC curves. This innovation transcends the temporal boundaries that have historically confined financial analysis, enabling a fresh perspective on asset price dynamics. The TF-DC trading strategy, a DC-based approach developed upon the principles of the Directional Change framework, is also introduced here. This strategy serves as the prototype for the DC-based methodologies constructed in this section.

4.2.1 Directional Change Framework

The Directional Change framework provides an event-based representation of financial price movements by replacing conventional time intervals with intrinsic market events. As introduced in Section 2.2, price movements are decomposed into two sequential components: DC events and OS events. A DC event is identified when the price movement exceeds a predefined threshold θ_{DC}, with the transition between upward and downward trends determined by DC confirmation points. Following the formal definition proposed by Tsang [58], this study applies the DC framework to identify market events and construct DC-based trading strategies.

$$\begin{cases} P_{DCC\downarrow}: \; P_t \leq P_{EXTh} \cdot (1-\theta_{DC}) \\ P_{DCC\uparrow}: \; P_t \geq P_{EXTl} \cdot (1+\theta_{DC}) \end{cases}$$

Where, $P_{DCC\downarrow}$ and $P_{DCC\uparrow}$ are downturn DC confirmation points and up DC confirmation points respectively. P_t is the current price, P_{EXTh} and P_{EXTl} denote the extreme highest point and lowest extreme point respectively.

4.2.2 Scaling Laws

The DC scaling laws provide the theoretical foundation for the trading strategies developed in this study. Previous empirical studies, particularly in the FX market, have demonstrated two important regularities: the magnitude of an OS event is approximately equal to the preceding DC event, and the duration of an OS event is approximately twice that of its corresponding DC event [1]. These relationships can be illustrated in Figure 4.1 and expressed as:

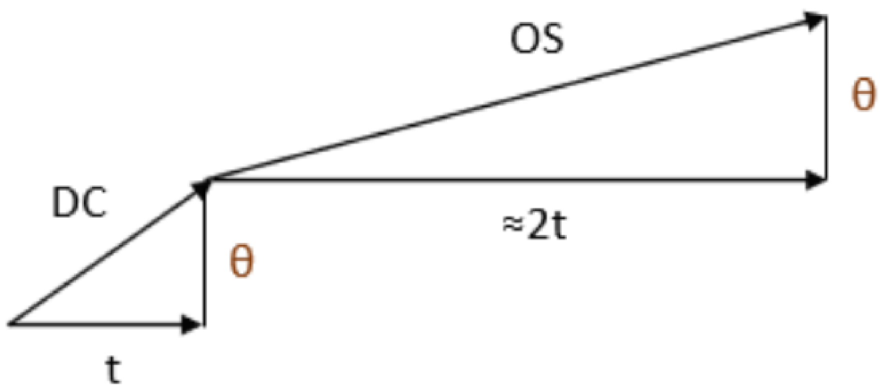

Figure 4.1: Two scaling laws from [1], that shows (a) price changes in DC and followed OS is approximately same and equals to θ_{DC}, (b) if DC event takes t physical time, its subsequent OS takes approximately $2t$.

$$DC_l \approx C \cdot OS_l \tag{4.1}$$

where the constant number C is equal to 1. Another important regularity in Figure 4.1 is that the duration of an OS event is approximately twice the duration of its DC event. We still can formulate it as:

$$\bigtriangledown OS_T \approx 2 \cdot \bigtriangledown DC_T \tag{4.2}$$

4.2.3 Simple Trend-following DC-based Trading Strategy

The objective of applying the DC framework in this study is to develop event-driven trading strategies based on its statistical properties, particularly the scaling laws. As reviewed in Section 2.2.4, several DC-based trading strategies have been proposed in the literature. Among them, the Alpha Engine developed by Golub et al. [62] provides an important reference for this study. The Alpha Engine employs multiple trading agents with different θ_{DC} values, which avoids the need to determine a single optimal threshold and enables the strategy to capture market movements across different time scales.

The DC-based trend-following strategy introduced in Chapter 3 is adopted as the baseline trading framework in this chapter. The detailed trading rules of the DC-based trend-following strategy remain consistent with those described in Chapter 3. This strategy is constructed based on the event-driven characteristics of the DC framework, where trading positions are initiated according to DC events and closed when the predefined profit target is achieved. In essence, any initiation of new positions or management of existing positions in the market aligns with the alternating movements of DC or OS events. The specific trading rule entails:

• Set 4 trading agents with different pre-defined DC threshold θ_{DC} (in this research they are 0.25%, 0.5%, 1.0% & 1.5% respectively) and their own profit target Pt. (The same value as their DC threshold).

• Opening a new position at the beginning of an OS event. The direction of the new position is as same as the current price trend.

• Close the position only when the profit of this position reaches the profit target P_t.

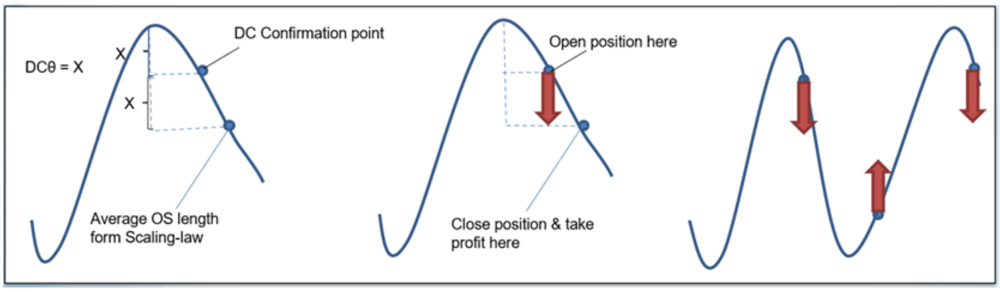

Figure 4.2: Trading rules of Simple trend-following trading strategy.

Figure 4.2 exemplifies the execution of trend-following trading rules, with each position closed upon attainment of a pre-determined profit target. To improve robustness across different market conditions, the proposed framework employs four independent trading agents with different DC thresholds (θ_{DC}). These thresholds represent different market observation scales, allowing the strategy to capture price movements across multiple time horizons while avoiding dependence on a single optimal threshold. In this study, the selected θ_{DC} values are 0.25%, 0.5%, 1.0%, and 1.5%, respectively.

In the preceding study, we investigated the role of a fixed-value stop-loss system in enhancing the performance of the Simple Trend-Following Trading Strategy. Currently, we are poised to introduce a novel system, the forecasting OS event two-classification model based on Machine Learning Algorithms, designed to supersede the effects of the stop-loss system.

4.3 FOS: A Trading Strategy based on Forecasting Overshoot of the DC Framework

4.3.1 Machine Learning Algorithms

This subsection introduces the machine learning models adopted in this chapter to investigate the predictability of OS events within the DC framework. Five machine learning algorithms were employed, namely Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN). Since the theoretical foundations of these models have been discussed in Chapter 2, only a brief overview of their roles in this study is provided here.

Random Forest: RF is an ensemble learning model that combines multiple decision trees to capture nonlinear relationships between input features and target variables. In this study, RF is employed to identify the relationship between DC event characteristics and subsequent OS event outcomes while providing robust prediction performance. Its ensemble structure also reduces the risk of overfitting and provides a useful benchmark for evaluating more complex models.

Support Vector Machine: SVM is a supervised learning model that constructs optimal decision boundaries in a high-dimensional feature space. Within this research, SVM is used to model the nonlinear relationship between DC-derived features and future OS event outcomes. The kernel-based formulation of SVM enables the model to capture complex feature interactions even when the available training data are limited.

Artificial Neural Network: ANN is a multilayer learning model capable of approximating complex nonlinear functions through interconnected neurons. In this study, ANN is applied to learn hidden patterns embedded in DC event features and generate predictions of future OS event outcomes. Its flexible architecture allows the model to represent highly nonlinear relationships that may not be captured by traditional machine learning approaches.

Long Short-Term Memory: LSTM is a recurrent neural network architecture designed to capture temporal dependencies in sequential data. Since DC events occur in a chronological sequence, LSTM is adopted to model the temporal evolution of market dynamics and improve OS event prediction. By explicitly maintaining information from previous events through its memory cells, LSTM can exploit long-range dependencies within the DC event sequence.

Convolutional Neural Network: CNN is a deep learning model that extracts informative local patterns from structured data through convolution operations. In this study, CNN is applied to identify local feature representations within DC event sequences that may contain predictive information about future OS events. The convolutional structure enables CNN to automatically learn discriminative event patterns without requiring manually designed feature interactions.

Within the DC framework, these machine learning models utilize DC-related features, such as event duration, price movement, and volatility information, to classify whether the subsequent OS event will exceed the predefined threshold associated with the trading strategy. Rather than forecasting general market trends or price levels, the models focus specifically on the prediction of DC-defined overshoot-event outcomes. The resulting predictions provide additional information for position management and trade selection within the DC-based trading strategies developed in this thesis. Detailed hyperparameter settings for all models are reported in Table 4.1.

Table 4.1: Hyperparameters and their best settings for RF, SVM, ANN, LSTM and CNN models used in this study.

Model	**Hyperparameters**	**Best Option**
RF	n_estimators	61
	max_depth	14
	max_features	8
	min_samples_leaf	3
	min_samples_split	12
SVM	Sigmoid kernel gamma	0.47
	Sigmoid coef0 (γ)	1
ANN	Number of hidden_layer	3
	hidden_layer	(140, 400, 370)
	learning rate optimizer	0.1 Adam
	activation function	sigmoid
	loss function	binary_crossentropy
LSTM	Number of LSTM_layer	3
	LSTM_layer	64,128,64
	dropout	0.1
	sequence length	20
	optimizer	ReLu
	activation function	sigmoid
	loss function	binary_crossentropy
CNN	Number of CNN_layer	3
	Types of CNN_layer	Conv1D
	CNN layer filters	(32,64,128)
	kernel size	20
	fully connected layers	2
	optimizer	sigmoid
	activation function	ReLU (hidden layers), softmax (output)
	loss function	binary_crossentropy

The performance of machine learning models is highly dependent on the choice of hyperparameters. Prior to the final experiments, a hyperparameter tuning procedure was performed for each model. For RF, SVM, ANN, LSTM, and CNN, several combinations of key hyperparameters were evaluated using the training dataset and a separate validation dataset. Model selection was based on classification accuracy for the OS-event prediction task. The hyperparameter configuration achieving the best validation performance was retained for the final experiments. The optimal hyperparameter settings obtained through this procedure are reported in Table 4.1.

4.3.2 The Forecasting OS Event Two-classification Model Based on Machine Learning Algorithms

In the results of the previous chapter, we have already glimpsed the profitability and risk of the DC strategy. Naturally, we will now address the issue of reducing risk while preserving profitability. First of all, we need to clarify where the retracement of the DC strategy comes from. During relatively stable market conditions, where DC and OS events exhibit durations and magnitudes close to their empirical averages, the price movement after a DC confirmation point is more likely to extend into the subsequent OS event. Therefore, the DC-based trading strategy can generate returns by opening positions at DC events and benefiting from the expected continuation of the price trend. However, most OS events will appear at different lengths in the real markets. It's easy to get results that those agents tend to build up large positions with huge paper losses that they cannot close. This always occurs when the market experiences a strange trend movement with the opposite direction for the hold positions. In order to manage this challenge of risk from market trends, we introduce ML algorithms to reduce the impact of strong market trends.

Financial forecasting has attracted considerable research attention due to its importance in identifying market trends and supporting investment decisions [160]. Among various approaches, machine learning techniques such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), Random Forests (RF), and Recurrent Neural Networks (RNN) have demonstrated strong capabilities in modelling the noisy and nonlinear characteristics of financial time series. Previous studies have successfully applied these methods to market trend prediction using technical indicators, price signals, and trading volume information [65, 161, 162]. More recently, machine learning has also been integrated with the Directional Change (DC) framework. For example, Kampouridis and Otero [144] employed Genetic Programming to optimise DC indicators, while Adegboye et al. [163] investigated the relationship between DC and OS lengths using machine learning techniques. Although these studies demonstrate the potential of combining machine learning with the DC framework, the development of DC-based trading strategies remains relatively limited and warrants further investigation.

In short, we aim to predict the probability of future strong market trends using the ML tool, which improves profitability and reduces the risks associated with original DC-based trading models. Compared to the previous simple model classification model, we developed the OS event two-classification model to better match the DC-based trading strategy. It divided the OS event into two types: OS_a and OS_b:

$$\begin{cases} OS_a : & OS_l > DC_l \\ OS_b : & OS_l \leq DC_l \end{cases}$$

A simple example of this two-classification model is shown in Figure 4.3. The OS event threshold T dividing OS_a and OS_b is derived from the previously introduced scaling law within the DC framework and our trading strategy. The scaling law reveals that in the FX market, the average duration of OS events largely mirrors the length of DC events, which is determined by the preset θ_{DC}. Building upon this, we developed the Simple Trend-following DC-based Trading Strategy. In essence, this strategy initiates a position in the direction of the trend at the conclusion of a DC event. OS events exceeding the length of θ_{DC} result in immediate profits and position closure for the opened trades, whereas those shorter than θ_{DC} lead to prolonged open positions and unrealized losses. In summary, the efficacy of the strategy is contingent upon the length of the OS events

Figure 4.3: A simple example of type OS_a and OS_b. The red part is the DC events, while the black part is the OS events.

Then, by this logic, according to the θ_{DC}, it is clear to definite the OS event classification threshold. An OS event is divided into OS_a because the length of it is longer than the pre-set DC θ, meanwhile, it is divided into OS_b when the spread of it is shorter than the pre-set DC θ. It will classify whereas an OS event is a long OS event, an OS_a length is longer than the pre-set DC threshold, or a short event OS_b whose length is shorter than the θ_{DC}. Back to our established trading strategy, if the model predicts that the next OS event is type a, the new position will open. And if the forecast result shows the next OS event is type b, the model can avoid a loss from its' holding position. Using this strategy enhances the possibility of greater profitability. Based on this two-classification model, the OS events in Figure 4.4 can be divided into OS_a and OS_b.

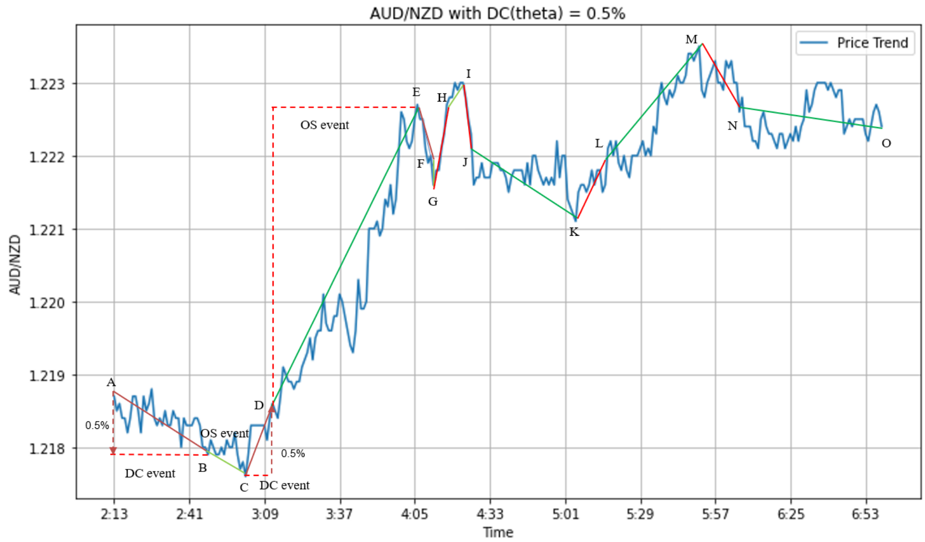

Figure 4.4: An example of a DC-based summary of the price series from Figure 2.1. $\theta_{DC} = 0.5\%$. OS_{BC}, OS_{FG}, OS_{HI}, OS_{NO} belong to OS_a and OS_{DE}, OS_{JK}, OS_{LM} belong to OS_b

In the previous chapter, we developed a simple trend-following trading strategy based on the DC framework. Now, we enhance this strategy by incorporating ML algorithms to predict the length of OS events and navigate the trading decisions accordingly. By doing so, the overall trading model will experience smaller drawdowns and better risk-adjusted performance due to a more controlled increase in the agent's inventory. Specifically, in the original trading model, the trading agent opens a new position at the beginning of each OS event with a default position size of S. However, with the ML binary classification model predicting the type of the upcoming OS event, the position size will be adjusted. If the OS event is predicted to be of type B (shorter OS event), the new position size will be significantly reduced (set to 0.1 times S). Conversely, if the OS event is predicted to be of type A (longer OS event), the position size will be substantially increased (set to 2.0 times S). The classification threshold T for OS events is set to the same θ_{DC} used in the test dataset. The size of new position S_{new} under the ML binary classification model will be:

$$S_{new} = \begin{cases} 2.0 \cdot S & If \;\; OS_A, \;\; T = \theta_{DC} \\ 0.1 \cdot S & If \;\; OS_B, \;\; T = \theta_{DC} \end{cases}$$

This architecture necessitates our trading strategy to focus predominantly on the projected lengths of OS events. In harmony with the core conjecture of the Efficient Market

Hypothesis, it is understood that all influencing elements, both internal and external, pertaining to price shifts are already embodied within the price. Nevertheless, when market circumstances resemble or replicate past situations, investors, guided by previous successful experiences or instructive failures, shape their present investment decisions, thus initiating a recurrence of market behavior and securities price trends [164] [165]. This logic grants us reasonable grounds to infer that indications towards future price trends can be drawn from historical price movement data. Accordingly, we implement machine learning models to identify recurring historical trend patterns and leverage them to deduce the lengths of forthcoming OS events.

4.4 Experiments

4.4.1 Research Data

This study relies on data collected from eight distinct currency pairs, namely AUD/NZD, EUR/GBP, EUR/JPY, EUR/USD, USD/CAD, USD/CHF, USD/JPY, and NZD/JPY, spanning from January 2006 to December 2020. All data employed in this research are retrieved from ”https://www.histdata.com/”. The dataset used for the training process comprises currency pair data from 2006 to 2016.

To conduct the tests, the period from 2017 to 2020 is utilized, wherein the dataset is tailored into eight distinct subsets to align with the eight different currency pairs. This differs from the training set, where all the currency pairs serve as collective training data. The study then uses the forecasting outcomes from five distinct machine learning algorithms to steer the Directional Changes (DC)-based trading strategy across these eight currency pairs within the same period, from 2017 to 2020. Figure 4.5 delineates the average unleveraged returns curve of the ’buy and hold’ strategy, the original DC-based simple trend-following strategy, and five iterations of modifying strategies incorporating five common machine learning algorithms applied to these eight currency pairs during the specified time frame.

4.4.2 Hyperparameters and Features

In the DC framework, historical prices are no longer regarded as a sequence of time-series data but as a collection of consecutive events. Accordingly, we leverage the characteristics of OS and DC events—such as propagation, length, and duration—as input features for training machine learning models. The features of OS and DC events have been introduced in Section 2, where the formula from 2.7 is directly applied as:

$$e = \left[e_{type}, \quad d, \quad p_s, \quad t_s, \quad p_e, \quad t_e, \quad l, \quad T \quad ROE \right] \tag{4.3}$$

where e_{type} denotes the category of the event and distinguishes between Directional Change (DC) events and Overshoot (OS) events. It is encoded as a binary variable, where $e_{type} = 1$ represents a DC event and $e_{type} = 0$ represents an OS event. The variable d describes the price movement direction associated with the event, where $d = 1$ indicates an upward movement and $d = -1$ indicates a downward movement.

The variables p_s and p_e represent the starting and ending prices of the event, respectively, while t_s and t_e denote the corresponding start and end timestamps. The event length l measures the magnitude of the price movement during the event, and the duration T represents the elapsed time between the event start and end points. The rate of event movement, denoted as ROE, is calculated as the ratio between the event length and its duration:

$$ROE = \frac{l}{T} \tag{4.4}$$

which captures the intensity of price movement during the event. In addition, the predefined DC threshold θ_{DC} is included as an input variable to indicate the sensitivity level used for constructing the DC events.

Excluding p_s, t_s, p_e, and t_e, the final set of features characterizing each event is as follows:

$$e = \left[e_{type}, \quad d, \quad l, \quad T \quad ROE \quad \theta_{DC} \right] \tag{4.5}$$

The selection of these features is motivated by the event-driven nature of the DC framework. Unlike conventional time-series models that primarily rely on fixed-time price observations, the DC framework represents market dynamics through discrete price movement events. Therefore, the selected features aim to capture the essential characteristics of a DC event that may influence the formation and magnitude of the subsequent OS event. Specifically, e_{type} and d describe the directional information of the current event, which is important for identifying whether the following OS event occurs in an upward or downward market condition. The event length l and duration T characterize the magnitude and temporal scale of price movements, respectively, while ROE reflects the intensity of the price movement by measuring the relationship between price change and elapsed time. In addition, θ_{DC} is included to represent the resolution of the DC framework, as different thresholds generate different event representations and may affect the statistical properties of DC and OS events. By combining directional, magnitude, temporal, and volatility-related information, these features provide a comprehensive representation of the current market state for predicting the length of subsequent OS events.The detailed definitions and descriptions of these DC-related variables are provided in Table 2.1 in Chapter 2.

To predict the length of the subsequent OS event, we assume that the evolution of recent DC and OS events contains valuable information about future market movements. Therefore, the input data are constructed from the characteristics of the previous 10 events. However, different machine learning models require different representations of historical event information due to their architectural differences.

For LSTM and CNN models, the sequential characteristics of the DC framework are directly preserved. Specifically, the models take the original attributes of the previous 10 DC/OS events as sequential inputs, where each event is represented by its corresponding features, such as event length, duration, price movement, and movement speed. The LSTM model is designed to capture temporal dependencies among these consecutive events through its memory mechanism, while the CNN model extracts local patterns and interactions within the event sequence through convolution operations. Therefore, these two models can directly learn the evolution patterns embedded in the sequence of recent market events without additional feature aggregation.

In contrast, RF, SVM, and ANN models do not have inherent mechanisms for processing sequential information. To provide these models with sufficient historical context, the characteristics of the previous 10 events are transformed into statistical features before training. Specifically, the simple moving average (SMA) and weighted moving average (WMA) of the attributes from the previous 10 OS and DC events are calculated and used as input variables. The SMA provides a general representation of recent market behaviour, while the WMA assigns larger weights to more recent events, allowing the models to capture relatively recent market conditions. This transformation enables RF, SVM, and ANN models to incorporate historical event information while maintaining their original non-sequential learning structures.The final feature set includes information related to the DC threshold, event magnitude, event duration, movement speed, and price momentum. The detailed definitions of these input features are presented in Table 4.2.

Features	Name	Description
X0	θ_{DC}	pre-set θ_{DC}
X1	$OS-SLA$	Simple 10-OS event length average
X2	$OS-WLA$	Weighted 10-OS event length average
X3	$OS-SPA$	Simple 10-OS event start price average
X4	$OS-WPA$	Weighted 10-OS event start price average
X5	$OS-SDA$	Simple 10-OS event duration average
X6	$OS-WDA$	Weighted 10-OS event duration average
X7	$OS-SSA$	Simple 10-OS event speed average
X8	$OS-WSA$	Weighted 10-OS event speed average
X9	$DC-SDA$	Simple 10-DC event duration average
X10	$DC-WDA$	Weighted 10-DC event duration average
X11	$Momentum_{short}$	$P_{OS_i} - P_{OS_{(i-2)}}$
X12	$Momentum_{long}$	$P_{OS_i} - P_{OS_{(i-10)}}$

Table 4.2: OS length prediction features - A brief description of independent variables used for classifying OS_i events. There are a total of 13 features. Where P_{OS_i} is the start price of OS_i.

This subsection outlines the key hyperparameter settings for each ML prediction model utilized in this study, as detailed in the subsequent table. These parameters were determined via a grid search experiment conducted on a dataset configured with a DC threshold of 0.5%. Specifically, since these models are primarily used for two-class classification to predict OS event types, the hyperparameters were selected based on the best accuracy performance achieved on the test dataset during training. The grid search experiment utilized data from 8 currency pairs over a 4-year period (2006–2010) as the training set, with the following year (2011) serving as the test set. Accuracy results were calculated as the average across test datasets for all currency pairs. It is worth noting that the purpose of these ML models is to enhance the profitability of DC-based trading strategies. Upon predicting the type of an OS event, if the OS event is classified as type B, the trading model reduces the size of new opening positions. Conversely, if the event is classified as type A, the position size is increased.

4.4.3 Result and Analysis

The "Buy and Hold" (B&H) trading strategy, also referred to as a "long-term investing" approach, involves purchasing securities, such as stocks or mutual funds, and holding them for an extended period. A detailed description of this strategy is provided in Section 2.1. Generally, the B&H strategy is widely recommended as a benchmark for comparing the performance of other investment strategies or individual stocks. For example, Hsieh (2021) examined the asymptotic log-optimality of the B&H strategy in achieving long-term wealth growth [34]. Additionally, Hsieh et al. (2019) compared the performance of high-frequency trading strategies and the B&H strategy under execution delay conditions [35]. While the B&H strategy is distinct from market return, the two concepts are closely related and often analyzed together in the context of investment performance.

In our research, the 'buy and hold' strategy is still introduced as a way of measuring the overall performance of the basket of currency pairs during the test period. If the returns of DC-based strategies exceed the 'buy and hold' strategy over the same period, this will powerfully indicate the Profitability of DC strategies. We presume the trading agent of the 'buy and hold' strategy will buy all their funds into this currency pair at the beginning of the test period and calculate the return at the end of the test period. Correspondingly, we also run our DC-based strategies during the test period.

The trading model is back-tested on historical data comprised of 8 currency pairs. All the DC-based strategies got positive results of the return on testing time period. The B&H strategy as the benchmark shows negative returns in the whole test period, which means the market return is negative. All the DC-based strategies can also obtain excess returns in case of a negative market return. After combining the five different versions of strategy returns, the comparison shows that the ANN model demonstrated higher profitability during the chosen test period. The strategy based on the RF predictor also demonstrated great profitability, however, with maximum drawdown. Although the overall performance of the

LSTM and CNN model, serving as an improved version of the ANN model, is not as good as that of the ANN model, they also have a positive impact. The SVM model barely improves the P&L for the DC-based strategy, and it deprives the performance. From the rate of return curves, it is obvious that the frequency of P&L increase or decrease is consistent, even though they have different guide ML algorithms for opening positions. The reason for this may derive from the scaling laws which they followed.

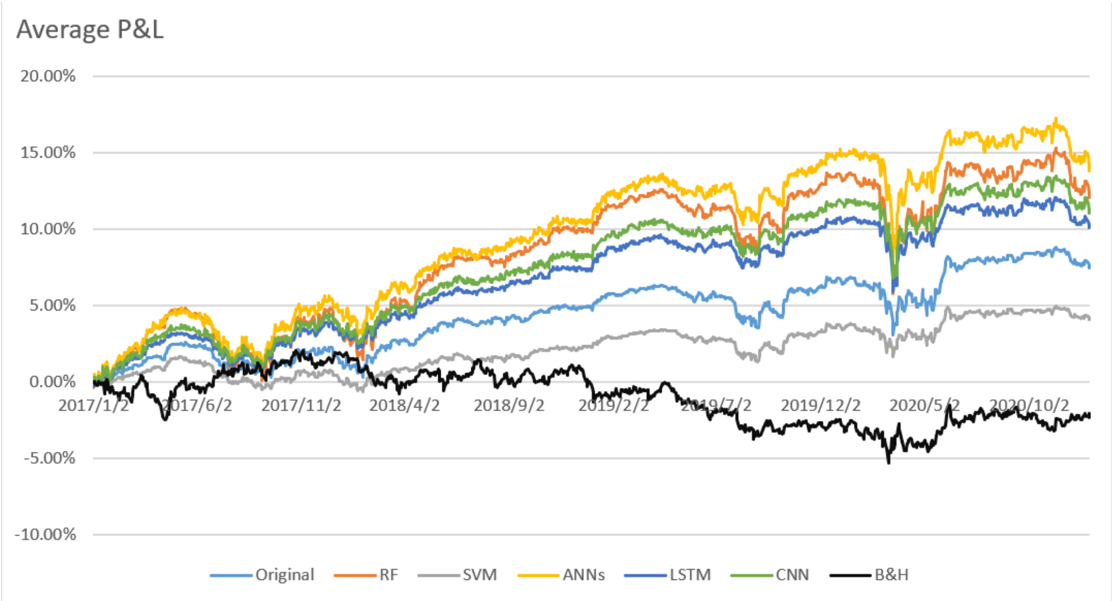

Figure 4.5: Average Profit and Loss for the 'buy and hold' the original DC-based Trend-following and five ML-based strategies on the FX market from 2016 to 2020.

The evaluation of trading strategies presented in Table 4.3 highlights significant performance variations across currency pairs and ML models, providing valuable insights into the applicability of the DC framework in the FX market from 2017 to 2020. On average, RF, LSTM, and ANNs outperform other ML models and the B&H strategy in profitability (P&L) and risk management (MDD).

Currency Pair Analysis reveals distinct patterns. RF achieves the highest profitability with AUD/NZD (P&L: 23.75%, MDD: 5.12%), demonstrating its ability to capture this pair's trend characteristics, while ANNs excel in EUR/GBP (P&L: 27.47%, MDD: 7.24%) and EUR/USD (P&L: 16.58%, MDD: 7.25%). Notably, all the DC-based trading strategies, including the original sample and modified ML-based strategies, yielded positive P&L results on most currency pairs except USD/JPY. On USD/CHF, RF achieved the best result (P&L: 14.88%). For USD/JPY, where the sample DC-based trading strategies generally performed poorly, LSTM and CNN slightly improved the results, while the SVM model worsened them.

Performance Trends indicate that ANNs, RF, LSTM, and CNN models improved the profitability of the original strategy, with ANNs showing the most consistent enhancements across multiple currency pairs. On AUD/NZD, EUR/GBP, EUR/USD, and USD/CAD, the unleveraged final profits using ANNs reached 18.95%, 27.47%, 16.58%, and 18.98%, respectively. The results highlight that ANNs, LSTM, and CNN models could improve profitability by approximately 1.2 to 1.5 times over the original trend-following strategy, emphasizing their potential for future exploration.

Risk and Drawdown Analysis shows mixed results. While ML-based trading strategies improved profitability, the mean MDD did not significantly decrease, except for LSTM and CNN. For NZD/JPY, the largest MDD among ML-based strategies exceeded 35%, indicating challenges in managing high-volatility currency pairs. By comparing price movements, it was observed that severe drawdowns occurred during strong trend states (either uptrend or downtrend), with most position losses eventually recovered through subsequent price reversals. This suggests that the DC-based trading strategy could achieve positive returns over a long time horizon when applied to assets with fluctuating price ranges.

	Original		RF		SVM		ANNs		LSTM		CNN		B&H	
	P&L(%)	MDD(%)	P&L(%)	MDD(%)	P&L(%)	MDD(%)	P&L(%)	MDD(%)	P&L(%)	MDD(%)	P&L(%)	MDD(%)	P&L(%)	MDD(%)
AUD/NZD	12.97	2.84	23.75	5.12	7.92	2.72	18.95	3.15	12.02	0.47	12.21	0.61	5.18	8.91
EUR/GBP	12.51	5.24	10.62	11.59	8.32	3.89	27.47	7.24	22.55	21.04	22.92	1.18	2.17	7.52
EUR/JPY	2.94	6.23	7.28	13.86	0.21	4.32	8.49	11.24	-0.46	0.89	0.32	1.58	2.18	6.18
EUR/USD	6.84	6.35	13.48	9.15	2.92	6.79	16.58	7.25	10.17	1.48	10.27	1.38	13.15	8.54
NZD/JPY	3.39	5.86	-2.55	35.18	3.51	13.48	1.52	28.49	2.78	14.14	2.55	12.21	-4.61	11.72
USD/CAD	12.64	6.25	15.05	25.12	5.59	15.32	18.98	9.57	7.97	9.73	10.12	12.04	-9.43	16.84
USD/CHF	8.33	8.62	15.88	8.62	2.87	2.61	9.44	7.39	5.34	0.21	6.15	0.29	-15.81	16.41
USD/JPY	-3.93	4.37	-0.63	17.65	-5.25	5.62	-1.31	15.26	0.16	17.35	0.21	19.56	-13.16	11.75
Average	8.49	2.14	12.9	6.33	4.67	2.19	13.65	5.81	11.46	3.46	12.49	3.86	-3.48	5.63

Table 4.3: Trading strategies evaluation table for FX market (2017-2020)

Model Comparisons and Market Conditions reveal that RF and ANNs perform well in relatively stable markets, while LSTM and CNN demonstrate adaptability to moderate volatility. However, SVM underperformed across most currency pairs, reducing the original strategy's profitability. For highly volatile pairs like NZD/JPY and AUD/NZD, risk management remains a challenge, requiring further refinement of the ML models.

In conclusion, the findings suggest that DC-based strategies integrated with ML algorithms, particularly ANNs, LSTM, and CNN, hold promise for improving profitability in FX trading. However, managing drawdowns in high-volatility conditions and further optimizing the models are essential steps for future research. These insights underscore the potential of the DC framework in generating positive returns under appropriate market conditions, provided the strategies are deployed over a sufficiently long period.

4.5 Conclusion

This research explores the application of the DC framework to predict price movements and evaluate its suitability for trading strategies. Five machine learning algorithms—Random Forest, Support Vector Machine, Artificial Neural Networks, Long Short-Term Memory, and Convolutional Neural Networks—have been employed on historical FX market data. The study successfully integrates ML models with the DC framework by establishing and refining a binary classification model to classify OS events in the FX market into long and short types and predict the type of future OS events. The input features include simple and weighted averages of OS event duration, spread, length, and start price. Results indicate that the DC-based trend-following strategy demonstrates profitability in underlying assets with fluctuating price ranges. While this provides preliminary evidence that ML algorithms are compatible with the DC framework, further in-depth analysis and validation are necessary.

This chapter highlights the importance of accurately predicting the length of OS events as a critical factor in increasing profitability and mitigating risks in DC-based strategies. Although the current prediction results are suboptimal, future work will focus on refining the ML models to enhance prediction accuracy and evaluate their broader applicability to financial market data. Reinforcement learning will also be considered for further exploration, as it offers potential for dynamic decision-making within the DC framework. Improving the performance of ML algorithms in the context of the DC framework is crucial for enhancing trading strategy profitability.

Most existing research on the DC framework has been concentrated on the FX market, and this study similarly focuses on FX data. However, applying DC-based trend-following models to commodity futures markets and Chinese stock markets poses challenges due to the non-applicability of the DC scaling law in these markets. Future research will examine how the DC scaling law adapts to other financial markets and explore modifications necessary for its broader applicability.

Chapter 5

Reinforcement Learning for a Novel Trading Algorithm

In the domain of algorithmic trading, the development of profitable trading strategies plays a critical role in enabling automated trading systems to make effective and successful decisions. This research delves into the training of a value judgment agent to enhance the profitability of trend-following strategies, specifically those based onDCprinciples, within a dynamic algorithmic trading setting. To achieve this objective, we propose a comprehensive representation of environmental states by leveraging the DC framework. The Directional Change (DC) framework offers a paradigm shift in financial time series analysis, emphasizing event-driven sampling over fixed time intervals. Our novel approach combines reinforcement learning (RL) and the DC-based trend-following strategy to identify optimal trading opportunities. The proposed algorithmic trading strategy aims to generate profitable trading rules based on the RL agent's learning from the environment's states and the DC-based trend-following strategy.We assess the effectiveness of our DC-based RL trading strategy on the FX market using the RL framework, and our findings demonstrate substantial trading returns resulting from the proposed policies.

5.1 Introduction

Algorithmic trading has become a subject of great interest to investors and financial analysts as it has the potential to make timely stock trading decisions. However, the decision-making process for financial trading is challenging due to the diverse factors that can impact stock prices. Algorithmic trading is based on computer algorithms that make automated trading decisions and place orders in the market, and recent advancements in information technologies and machine learning (ML) techniques have led to the creation of quantitative trading. Classic algorithmic trading strategies include trend-following and mean reversion strategies, which have been studied extensively. ML techniques for algorithmic trading can be divided into supervised learning and RL algorithm-based methods. RL, in particular, has been effective in different domains, including algorithmic trading, by achieving policy improvement through continuous interaction and evaluation of its environment. Despite the effectiveness of the RL algorithm, employing an algorithmic trading strategy in real-world trading remains challenging due to the physical fixed time interval, difficulty in selecting appropriate features and data, and complexity of the ML algorithms. Therefore, in this study, the authors propose a framework that combines RL with the Directional Change-based trend-following strategy to develop an algorithmic trading strategy. This study aims to address the challenges in algorithmic trading by developing an RL-based algorithmic trading strategy using the framework of RL and Directional Change-based trend-following strategy. The DC event approach emerges as an alternative approach for price time-series analysis that can capture periodic patterns in price time-series.

The RL algorithm has been applied effectively in different domains, such as job scheduling, pattern recognition, and algorithmic trading. Despite the effectiveness and robustness of the RL algorithm, employing an algorithmic trading strategy remains a challenge in real-world trading for three reasons. First, using a physically fixed time interval to represent the environment states can make the flow of price time-series irregularly spaced. Second, selecting appropriate features and data to represent the environment states can be difficult. Finally, ML algorithms have a complex structure and a large number of different parameters. In reality there may not be enough data to train these large models.

Moody et al. [166] pioneered the use of RL in financial trading, optimizing utility functions and evaluating investment performance, departing from traditional supervised machine-learning methods. With the rapid advancement of deep learning, its application in the financial domain has become increasingly widespread. Deep architectures possess remarkable approximation capabilities, ensuring better convergence. Advanced Deep Reinforcement Learning (DRL) algorithms can be classified into two main categories: value-based algorithms [167–169] and policy-based algorithms [49, 121, 131].

Subsequent research explored various RL-based approaches. Almahdi and Yang [88] proposed an RL method for optimal asset allocation, optimizing the expected maximum drawdown. Gold [170] and Gabrielsson and Johansson [171] applied RL in high-frequency trading (HFT) for Forex and futures markets. Deng et al. [122] integrated deep learning with RL for direct trading and financial signal processing, utilizing deep neural networks and fuzzy networks for price time-series data.Carapuço et al. [172] investigated the performance of the Deep Q-Network (DQN) algorithm in the Forex market, adapting small granularities of price time-series to create the state space. Sornmayura [173] used DQN for robust Forex trading, surpassing buy-and-hold strategies. Li et al. [86] studied DQN, double DQN, and dueling DQN algorithms. Zarkias [174] employed DQN with a price trailing mechanism to track price trends. Ensemble value-based DRL methods were also investigated, such as Lee and Kim [175] proposing an action-specialized ensemble learning approach based on DQN for automated trading. Clempner [176] introduced a continuous-time RL model for portfolio management, utilizing an actor-critic structure. Jiang and Liang [87] implemented DRL for investment portfolio management in the cryptocurrency market. Li et al. [177] employed DQN and asynchronous advantage actor-critic (A3C) for financial trading, while Zhang, Zohren, and Roberts [178] applied DRL with DQN, Policy Gradient, and Advantage Actor-Critic (A2C) for designing trading strategies for future contracts. These findings suggest that the proposed RL-based algorithmic trading strategy has great potential for real-world applications and can contribute to the development of the field of algorithmic trading.

The utilization of the DC framework is another focal point of this study. The concept of DC can be traced back to the 'Zigzag' indicator in technical analysis [145]. Guillaume et al. [57] first introduced the concept of DC. Tsang [147] provided a formal definition of the DC concept and showed an example of the DC events price curve for FX markets. Unlike traditional time series frameworks, DC samples data at peaks and troughs [157], reducing noise impact and offering a targeted entry point for trading. Fundamentally, DC's event-based approach is a powerful tool in financial trading and market analysis, fostering innovative trading strategies. Several notable DC-based trading strategies have already been proposed, each contributing valuable insights and practical applications [64] [67, 70] [58, 150] [62, 73, 150]. These studies collectively demonstrate the potential and applicability of the DC Framework in creating effective trading strategies for financial markets.

Combining the DC framework with RL techniques represents a frontier in algorithmic trading research, leveraging the strengths of both approaches to design adaptive and potentially profitable trading systems. [68] introduced a novel RL-based algorithmic trading strategy operating within the DC framework, marking the first such integration in the field. By dynamically identifying DC events and optimizing trading decisions through RL agents, the model provides a unique method for capturing market patterns. This innovation was evaluated on empirical datasets from the Saudi stock market, demonstrating its applicability and performance in identifying significant price changes. Despite its contributions, the DCRL model proposed by [68] has limitations worthy of discussion. While it demonstrates strong performance in pattern detection and dynamic threshold optimization, its simplified assumptions about market conditions and transaction costs make it less aligned with real-world trading practices. Moreover, the model primarily focuses on using RL model to address the issue of DC threshold selection rather than the comprehensive design of trading strategies. This approach may constrain the actions of the RL agent and fail to fully capture its ability to learn and interact with the trading environment. This study aims to shift the focus toward designing a trading strategy driven by an autonomous trading agent capable of distinguishing DC market conditions and independently determining trading strategies.

The subsequent sections of this Chapter are organized as follows. Section 5.2 outlines the automated trading agent problem that the present study aims to address through the use of the DC-based strategy. Section 5.3 provides a succinct description of the DC event approach and introduces the DC-based trend-following strategy. Section 5.4 expounds upon the RL algorithmic trading strategy that is based on the DC-based trend-following approach. Section 5.5 presents the datasets, experiment settings, profitability results, and a discussion of the empirical findings. Lastly, Section 5.6 concludes the paper and provides recommendations for potential future research directions.

5.2 Problem Definition

Automated trading agent management refers to the process of designing, developing, and deploying trading agents that can autonomously make trading decisions on behalf of human traders or investors. These agents are typically software programs that utilize algorithms and artificial intelligence techniques, such as ML and natural language processing, to analyze vast amounts of market data and execute trades in real-time. The management aspect of automated trading agent management involves ensuring the optimal functioning of these agents by monitoring their performance, adjusting their strategies, and mitigating any risks associated with their use. This field has become increasingly popular in recent years, as more and more traders and investors seek to automate their trading activities to improve efficiency, reduce costs, and increase profits.

This study proposes an event-driven trading algorithm, utilizing the DC framework. The DC framework provides a unique approach to market analysis by presenting alternating DC and OS events data instead of traditional time series price data, which is better suited to reflect market fluctuations. The algorithm is further enhanced by utilizing the RL algorithm, a form of ML that enables the development of automated trading agents. The RL algorithm allows the agent to make optimal decisions by interacting with the environment through a trial-and-error process, learning from its actions and outcomes to maximize rewards or profits. In the context of automated trading, the agent can observe market conditions, adjust its behavior, and improve its decision-making capability over time. The DC framework, coupled with RL, offers a powerful approach to developing an automated trading agent that can effectively adapt to dynamic market conditions.

5.2.1 Mathematical Formalism of Trading Algorithm

The ultimate objective of this study is to explore how to enhance the returns of a trading algorithm. Therefore, it is imperative to first establish a precise definition of the returns. For a given period of time, T (from T_0 to T_t), the returns can be defined as follows:

$$ROR = \frac{C_t + P\&L}{C_o} \tag{5.1}$$

where ROR is the rate of return after running the trading agent from the start time T_0 to the end time T_t. C_o is the pre-set initial capital, meanwhile, C_t is the capital that the agent has at time T_t and P&L is the profit and loss of the holding positions. This equation is presented without considering transaction costs, and the version incorporating transaction costs will be addressed in subsequent sections.

As the trading strategy adopted in this study frequently opens and closes positions, the total P&L consists of two components: realized profit from closed positions and unrealized profit from positions that remain open at the evaluation time. For an open long position i, with position size S_i and entry price P_i, the unrealized profit at time T_t is defined as:

$$R_i^{long} = (P_t - P_i)S_i \tag{5.2}$$

where P_t denotes the current market price at time T_t. A positive value of R_i^{long} indicates that the position is currently profitable, whereas a negative value represents an unrealized loss.

Similarly, for an open short position i, the unrealized profit is calculated as:

$$R_i^{short} = (P_i - P_t)S_i \tag{5.3}$$

For a position that has been closed before time T_t, the unrealized profit is replaced by the realized profit using the closing price P_c. The realized profits for long and short positions are defined as:

$$R_i^{close,long} = (P_c - P_i)S_i \tag{5.4}$$

$$R_i^{close,short} = (P_i - P_c)S_i \tag{5.5}$$

where P_c represents the closing price of the position.

Accordingly, assuming that the trading agent holds n long positions and m short positions during the evaluation period, the total rate of return is calculated as:

The total return of the trading agent is evaluated based on the final value of the trading account, which consists of available capital, realized profits from closed positions, and unrealized profits from open positions. Specifically, the rate of return (ROR) over a given period T (from T_0 to T_t) is defined as:

$$ROR = \frac{C_t + \sum_{i=1}^{N_l^c} R_i^{long,c} + \sum_{i=1}^{N_l^o} R_i^{long,o} + \sum_{j=1}^{N_s^c} R_j^{short,c} + \sum_{j=1}^{N_s^o} R_j^{short,o} - C_0}{C_0} \tag{5.6}$$

where C_0 represents the initial capital allocated to the trading agent, and C_t denotes the available cash balance at the end of the evaluation period. N_l^c and N_s^c represent the number of closed long and short positions, respectively, while N_l^o and N_s^o denote the number of open long and short positions at time T_t. The superscripts c and o indicate closed and open positions, respectively.

This formulation provides a mark-to-market evaluation of the trading strategy by considering both realized and unrealized profits. Therefore, the final performance reflects not only completed transactions but also the value of positions that remain open at the end of the trading period.

5.2.2 Transaction Cost

Equation 5.6 provides a definition for the returns that disregards transaction costs. However, in actual trading, transaction costs cannot be simply disregarded. Transaction costs refer to the expenses incurred when buying or selling an asset, such as fees, commissions, bid-ask lengths, and market impact costs. In financial markets, transaction costs can significantly affect the profitability of a trading strategy. As this study is based on the FX market, the substantial trading volume enables us to temporarily neglect the transaction costs caused by bid-ask lengths and slippage. Additionally, even though we use minute-level trading data, the frequency of trades in our strategy is not high, averaging about one trade per day, with the busiest trading days not exceeding ten trades. Therefore, we consider the transaction costs as a fixed percentage trading commission. In this study, the transaction costs μ is defined as follow:

$$\mu = \frac{(P_i - P_t)}{P_i} * S_i * w \tag{5.7}$$

Where the S_i is the size of the opening position i. the w is the commission rate. In this study, w is equal to 0.3%. The trading commission rate used in this study is based on the broker with the largest trading volume in the FX market, IG Group. IG Group offers a variety of trading instruments, such as FX and contracts for difference, on a global scale, and provides multiple trading platforms and tools. Although the commission rates for each currency pair may vary, the rates for the main currency pairs are generally between 0.05% to 0.3%. We chose the highest value as our trading commission rate. The transaction costs for the holding positions are not calculated.

After accounting for transaction fees, the final rate of return of our trading algorithm is expressed as:

$$ROR = \frac{C_t + \sum_{i=1}^{N_l^c}(R_i - \mu_i)}{C_0} \tag{5.8}$$

where C_0 denotes the initial capital and C_t represents the available cash balance at the end of the trading period. R_i and μ_i denote the realized profit from positions (including closed and open, short and long) for position i and transaction cost for position i, respectively,

In this study, we exclusively focus on back-testing trading strategies and we ensure that no "future" market information is provided to the trading agent. In order to mitigate the impact of excessively large trading volumes on the financial market, we have set the initial trading amount to 100,000 US dollars.

5.3 The Simple DC-based Trend-following Strategy

In Chapter 3, we introduced a Simple DC-based Trend-following strategy derived from the scaling properties of the DC framework. This strategy serves as the baseline trading model for the RL framework proposed in this chapter. The trading decisions are determined by DC events rather than fixed time intervals, allowing the strategy to naturally adapt to market dynamics.

The strategy follows an event-driven trading mechanism. A trading agent is established with a predefined DC threshold θ_{DC} and a corresponding profit target P_t ($P_t = \theta_{DC}$). A new position is opened at the beginning of an OS event, following the current market trend. The position remains open until its profit reaches the predefined target. The trading rules are summarized as follows:

- Set a trading agent with a predefined θ_{DC} and profit target P_t ($P_t = \theta_{DC}$).
- Open a position at the beginning of an OS event in the direction of the current trend.
- Close the position only when the accumulated profit reaches P_t.

The setting of $P_t = \theta_{DC}$ follows the DC scaling law, which suggests that the magnitude of OS events is statistically related to the preceding DC events [146]. This simple strategy provides a benchmark for evaluating whether reinforcement learning can further improve DC-based trading decisions.

Although the Simple DC-based Trend-following strategy benefits from the event-driven nature of the DC framework, it has several limitations. First, the strategy does not include a dynamic position management mechanism, which may lead to excessive exposure during unfavorable market movements. Second, the fixed position size assumes identical confidence levels for all trading opportunities. To address these limitations, we introduce an RL framework that allows the trading agent to adaptively determine trading actions based on market conditions.

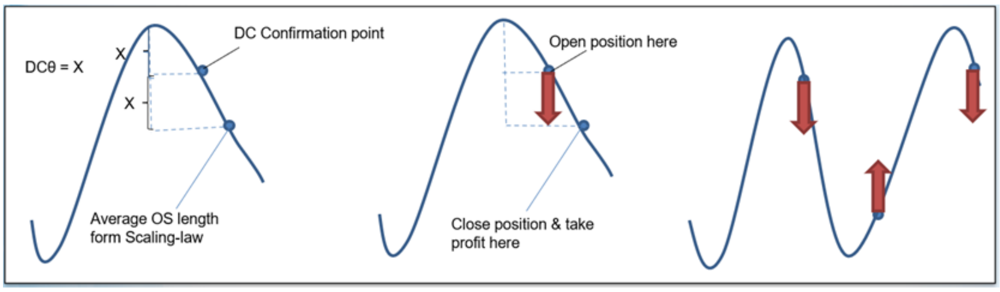

Figure 5.1: Trading rules of the Simple trend-following trading strategy. Within the figure, red arrows are indicative of the process encompassing both the opening and closing of a position. The commencement of the red arrow denotes the opening point of the position, which concurrently serves as the DC Confirmation point. Conversely, the arrow's endpoint signifies the closing point of the opened position, marking the juncture where the price has reached the predefined profit target, P_t. The orientation of the arrow illustrates the direction of the position: a downward trajectory indicates a short position, while an upward direction signals a long position.

A significant body of research has explored the application of RL in algorithmic trading. However, a common challenge in these studies is the definition of appropriate decision points. Many RL-based trading frameworks discretize the market into fixed time intervals and allow agents to adjust their positions at every time step. Although this approach provides a simple implementation, it may introduce redundant decisions during inactive market periods and fail to capture the occurrence of meaningful price movements.

The DC framework provides an alternative event-driven mechanism by defining trading opportunities according to significant price changes rather than predefined time intervals. In the Simple DC-based Trend-following Strategy, the beginning of an OS event following a DC confirmation point naturally serves as a potential position-opening signal. This allows the trading agent to interact with the market based on intrinsic market events and avoids unnecessary decisions caused by uniform time sampling.

In this study, we retain the event-driven characteristics of the DCTF strategy within the RL framework. Specifically, the DC framework continues to determine potential trading opportunities and the direction of the position, while the RL agent is introduced to improve the decision-making process by determining whether to execute a trade under the current market condition. Therefore, the proposed RL model does not replace the DC-based trading mechanism but enhances its adaptability by introducing a learning-based decision layer.

5.4 Reinforcement Learning with Directional Change

RL is a subfield of ML concerned with enabling an agent to make optimal decisions in a sequential decision-making process by interacting with an environment and receiving feedback in the form of rewards or punishments. This study focuses on the seamless integration of RL into our trading model based on the DC framework, intending to maximize cumulative rewards over time, enhancing the trading strategy's effectiveness and robustness.

Based on the trading strategy discussed in the preceding section, the trading agent initiates a position after each DC event. In cases where the market exhibits gentle fluctuations, with DC events and OS events alternating at identical intervals, the agents achieve consistent returns. However, in actual markets, OS events tend to occur at varying intervals. Consequently, the agents tend to accumulate large positions that they cannot close when they initiate positions at the onset of short OS events. This is particularly problematic when the market experiences unusual trend movements in the opposite direction to the held positions. To mitigate the risks posed by market trends, we propose an RL framework that enables the agent to independently decide whether to initiate a new position at current OS events. Our RL framework relies on the Keras and TensorFlow libraries.

The RL framework consists of four key components: the environment and state representation, the action space, the reward function, and the policy network. The environment is modeled as a Markov Decision Process (MDP), where the agent observes the current state and selects an action based on the so-called policy network. The state representation captures relevant market information and the agent's position. The action space defines the available actions, such as buying or selling assets. The reward function evaluates the agent's performance and updates its policy network accordingly, aiming to learn a policy that maximizes the expected return on investment in the trading context.

- Environment: Original time series price historic data
- State s: The stacking of the 4 normalized DC event matrices (the time series price data is transformed into the alternate DC & OS event data) and the position information.
- Action a: a vector of actions over the currency pair based on the DC framework. The allowed actions on each financial instrument are 0, 1, 2, 3, 4. Here, 0 means do nothing. 1, 2, 3, and 4 respectively mean open new positions with 0.0001, 0.001, 0.01, and 0.1 times of the initial capital. All the new positions have the target value which is without stop loss.
- Reward R (s, a, s'): the direct reward of taking action at state s and arriving at the new state s'.
- Policy network $\pi(s)$: it decides the trading strategy for an agent at state s. The output is a one-dimensional array including all the probability distributions of actions at state s.

5.4.1 Environment and State

The financial market constitutes a complex and dynamic environment in which technical traders assume that all relevant information is reflected in publicly available asset prices [179, 180]. In this study, the market environment is represented using the DC framework. Rather than describing market dynamics through uniformly sampled time-series observations, the proposed RL framework operates in the intrinsic time domain defined by DC events.

To capture market behaviour at multiple scales, the same price series is simultaneously transformed into four DC event streams using different threshold values. This design is motivated by the observation that market participants may react to trends of different magnitudes. Consequently, the environment provides the agent with multi-scale information describing both short-term and long-term market movements.

Unlike conventional RL environments that evolve at fixed chronological intervals, the proposed environment evolves according to the occurrence of DC events. A state transition is triggered whenever a new DC confirmation point is detected in any of the four DC event streams. Therefore, one RL step corresponds to one DC-event update rather than a fixed amount of calendar time. At each step, the agent observes the current market state, selects an action, receives a reward, and then transitions to the next state when a subsequent DC event is confirmed.

The state s represents the agent's observation of the current environment and serves as the input to the policy network. It is constructed as a rank-3 tensor with shape (m, E, e), where m denotes the number of DC thresholds, E denotes the number of historical events included in the observation, and e represents the feature vector describing an individual event.

As discussed in the previous chapter, each DC event can be represented by a set of descriptive attributes. Since multiple DC thresholds are employed in this chapter, an additional feature is included to identify the threshold associated with the event. The resulting event representation is defined as

$$e = \left[e_{type}, \quad d, \quad p_s, \quad t_s, \quad p_e, \quad t_e, \quad l, \quad T \quad \theta \right] \tag{5.9}$$

where e_{type} identifies the event type and takes values from $\{1, -1\}$, representing upward and downward DC events respectively. The remaining variables describe the direction, start and end prices, start and end times, event length, duration, and the associated DC threshold.

Since recent market events are expected to contain more relevant information than distant historical observations, only the most recent ten events are retained for each DC threshold. Therefore, ollowing function 2.8, one of the event sequence tensors of the state is:

$$E = \left[e_1 \mid \quad e_2 \mid \quad e_3 \mid \quad \ldots \mid \quad e_{10} \right] \tag{5.10}$$

For this study, four DC thresholds (0.25%, 0.5%, 1.0%, and 1.5%) were employed to generate four event streams from the same price series. The state representation is obtained by stacking the corresponding event matrices, thereby allowing the agent to simultaneously observe market structures at different scales.

In addition to market information, portfolio information is incorporated into the state representation. A capital matrix C is introduced to encode the agent's current trading status, including the available capital, the total long-position exposure, and the total short-position exposure. To ensure compatibility with the event tensors, zero-padding is applied to expand the portfolio representation to the same dimensionality as the DC event matrices. Consequently, the final state tensor contains five channels and has shape $(5,10,7)$.

This transformation can be represented mathematically as follows. Let the multi-scale DC tensor E_4 have shape $(4,10,7)$, and let the portfolio vector C_0 have shape $(3,)$. The portfolio tensor C is first constructed according to

$$\begin{cases} For\ i = 1,2,3: \;\; C(j,i) = C_0(i),\ \forall j \in [1,10] \\ \quad For\ i = [4,9]: \;\; C(j,i) = 0,\ \forall j \in [1,10] \end{cases}$$

The tensor C is then concatenated with E_4 along the first dimension, forming the final state tensor S. Figure 5.2 illustrates the resulting state structure

$$[C,\ E_{\theta=1.5\%},\ E_{\theta=1.0\%},\ E_{\theta=0.5\%},\ E_{\theta=0.25\%}].$$

At each state s, the trading agent selects one of five available actions. Trading decisions are made only when a new DC event triggers a state update. When a position is opened, its direction follows the current market trend, as defined by the corresponding DC event. Each position is assigned a profit target P_t, which is set equal to the associated DC threshold. Positions remain open until the predefined profit target is reached. The position size is determined by the policy network and therefore becomes part of the agent's decision-making process.

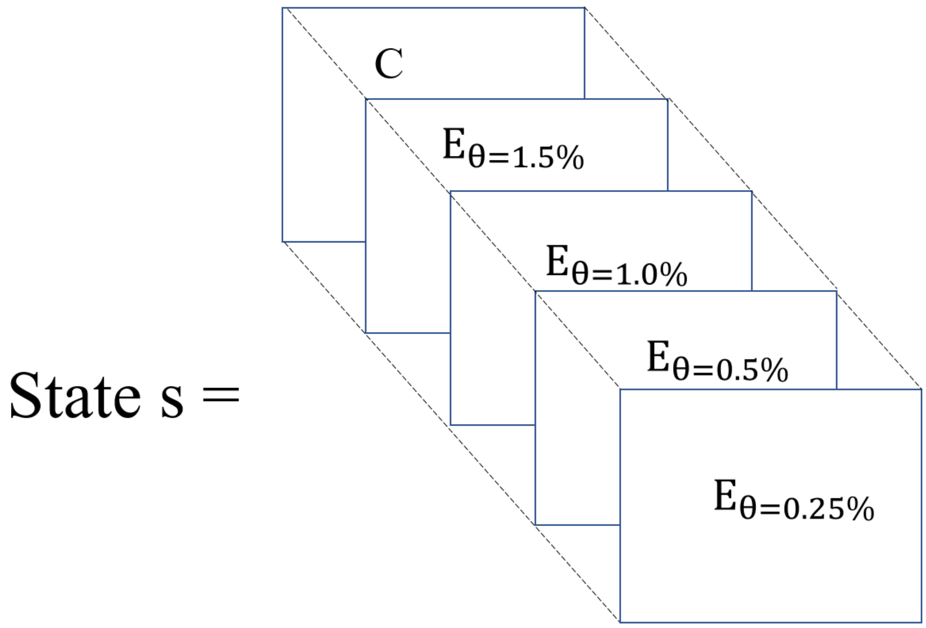

Figure 5.2: Tensor state s

5.4.2 Reward Function

The reward function used in this research aims to guide the agent toward selecting actions that lead to the best possible outcome. It is computed at the end of a game, rather than for each individual move. This approach is known as the "rollout" process. In accordance with the characteristics of the DC trend-following strategy, we also established an instantaneous reward that can be obtained after each step. The reward for a single action is the sum of the instantaneous reward r_{in} and the final result. According to equation (6), we get the expression of reward r as:

$$r = r_{in} + ROR - 1 = r_{in} + \frac{C_t + \sum_i^n R_{ci} + \sum_j^m R_{hj} - C_o}{C_o} \tag{5.11}$$

The instantaneous reward r_{in} is designed to reflect whether the current OS event provides sufficient price movement to achieve the predefined profit target of the trading position. In the proposed DC-based trading strategy, the profit target is set equal to the corresponding DC threshold θ_{DC}. Therefore, when the length of the OS event exceeds this predefined threshold, the price movement during the OS event is sufficient to close the position with the target profit. In this case, the agent receives a positive instantaneous reward of 0.1, indicating that the selected action has contributed to a profitable trading opportunity. Conversely, if the OS event length does not reach the predefined threshold, the position cannot achieve its target profit based solely on the current OS movement, and the instantaneous reward is set to 0. The updated reward is used to update the weights of the neural network using backpropagation.

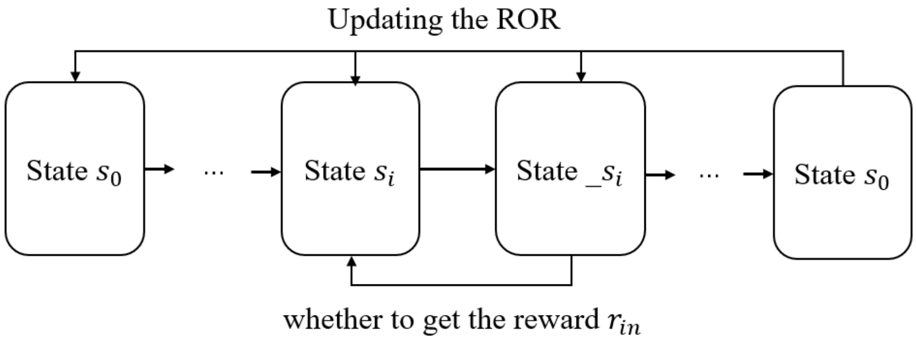

Figure 5.3: Reward update

5.4.3 Training Procedure

The proposed reinforcement learning framework was trained through repeated interactions with the DC-event environment. At the beginning of training, the policy network was randomly initialized, resulting in largely exploratory trading behaviour. As training progressed, the agent gradually shifted from exploration to exploitation by reducing the probability of selecting random actions.

To balance exploration and exploitation, an ε-greedy strategy was adopted. At each decision step, the agent selected a random action with probability ε and selected the action with the highest estimated reward with probability $1-\varepsilon$. Initially, a high exploration rate was used to encourage broad sampling of the state-action space. The exploration probability was then gradually reduced throughout training until only approximately 5% of actions were selected randomly. This schedule allowed the agent to first explore different trading behaviours and subsequently exploit the most profitable policies discovered during training.

Inspired by successful reinforcement learning systems such as the Asynchronous Advantage Actor-Critic (A3C) framework proposed by Mnih et al. [181], policy evaluation and policy improvement were implemented as separate computational tasks. During policy evaluation, the training samples were divided into multiple subsets and processed concurrently by different CPU threads. Each thread independently interacted with the DC-event environment and generated experience samples. After all evaluation tasks were completed, the collected experiences were aggregated and used to update the policy network. This parallelized design significantly improved computational efficiency and enabled effective utilization of the available multi-core processing resources.

To improve generalization and reduce overfitting, regularization techniques were incorporated during training. Dropout layers were employed within the neural network architecture to prevent excessive co-adaptation of hidden neurons, while weight decay was applied during parameter optimization to penalize overly complex models. These regularization mechanisms improved the stability of the learned policy and enhanced its robustness across different market conditions.

The training process consisted of alternating policy evaluation and policy improvement stages. During policy evaluation, the agent interacted with the environment and generated state-action-reward trajectories. During policy improvement, the collected experiences were used to update the policy network parameters, gradually increasing the probability of selecting actions that produced higher cumulative rewards. Through repeated iterations of these two stages, the policy converged towards a trading strategy that balanced short-term trading opportunities with long-term portfolio profitability.

5.4.4 Policy Iteration

Specifically, our RL algorithm consists of four main steps:

1. Initialization of the DC event environment and policy networks π. (A policy is a function that maps the state space to the action space, π: S $\rightarrow$ A. In the current framework, the policy produces a deterministic action for a given state s using full exploitation.)

2. trading strategy running: This strategy randomly selects an action with probability ε and selects the action that yields the highest expected rate of return for the current state with probability $1-\varepsilon$. Here, ε is a hyperparameter that controls the degree of exploration. In this research the ε is set as 0.9 and would be decreased by 0.005 each episode until it is 0.

3. Neural network training: Using the training samples generated during trading strat-

egy running, our RL framework trains the policy networks to improve their accuracy and generalization ability. The mean squared error loss function is used to optimize network parameters.

4. Policy iteration: Policy iteration is an RL algorithmic approach for solving RL problems. It involves two main steps: policy evaluation and policy improvement. In the policy evaluation step, the value function of the current policy is computed by iteratively applying the Bellman equation. In the policy improvement step, the policy is updated based on the current value function by selecting the action that maximizes the expected cumulative reward from the current state. These two steps are then repeated iteratively until the policy converges to the optimal policy, which maximizes the expected cumulative reward from the initial state.

The proposed policy iteration framework follows the general principles of policy-gradient methods, where the policy network is updated to maximize the expected cumulative reward obtained under the current policy.

During policy improvement, the agent updates the policy according to the estimated returns obtained during policy evaluation. To encourage both short-term profitable decisions and long-term trading performance, a composite reward function is employed.

$$R_t = r_{t+1} + \gamma r_E\left(\frac{n}{m}\right) \tag{5.12}$$

where R_t denotes the reward estimate associated with the current state-action pair. The term r_{t+1} represents the instantaneous reward obtained immediately after executing action a_t. In this study, the instantaneous reward is determined by the behaviour of the first subsequent OS event. If the OS length exceeds the predefined profit target of the position, a positive reward is assigned; otherwise the reward is zero.

The second component r_E represents the final rate of return (ROR) generated after the trading strategy has completed a full simulation on the selected chip. The variable m denotes the total number of events contained within the chip, while n denotes the current event index. Consequently, $r_E(n/m)$ provides a scaled estimate of the final trading performance and allows long-term profitability information to influence earlier trading decisions. The discount factor γ controls the relative importance of this long-term reward component.

The composite reward therefore combines local trading feedback with the final profitability of the entire trading episode. During training, the policy network parameters θ are updated using experiences sampled from the replay buffer, increasing the probability of actions that lead to higher composite rewards. Through repeated policy evaluation and policy improvement, the agent gradually learns a policy that balances short-term opportunities and long-term profitability.

5.4.5 Policy Network

The policy network $\pi(s)$ utilized in this study is based on a modified version of the ResNet architecture, originally introduced by He et al. [117], which has been widely adopted due to its effectiveness in various domains, including RL [125]. The ResNet architecture employed here consists of 14 convolutional layers, organized into seven residual blocks. Each residual block comprises two convolutional layers followed by a skip (residual) connection, which facilitates gradient flow and addresses the vanishing gradient problem during training [118]. These skip connections enable the effective training of deeper networks, leading to improved convergence rates and reduced error rates [117, 118].

Each convolutional layer in the network applies a set of learnable filters (kernels) to extract hierarchical features from the input data, with ReLU (Rectified Linear Unit) activations applied to introduce non-linearity and enhance the network's expressive power [182]. Pooling operations, such as global average pooling, are employed in the architecture to reduce the spatial dimensions and aggregate features efficiently, thereby mitigating overfitting [183]. The residual blocks in this network leverage identity mapping [118], which simplifies optimization and accelerates convergence, particularly in RL tasks where complex state representations are required.

Additionally, this architecture incorporates parameter sharing by reusing weights across multiple layers, significantly reducing the number of trainable parameters. This approach not only minimizes memory usage but also accelerates training, making it feasible to train larger networks with limited computational resources [120]. The ResNet's robustness and efficiency make it particularly suitable for RL applications, where both policy and value functions must be learned simultaneously.

In summary, the integration of ResNet into the policy network offers critical advantages, such as the ability to train deeper networks effectively, reduced computational overhead through parameter sharing, and enhanced learning capabilities in complex environments. This architecture's application in high-impact research, such as AlphaGo [125], highlights its reliability and suitability for advanced trading strategy design in this study.

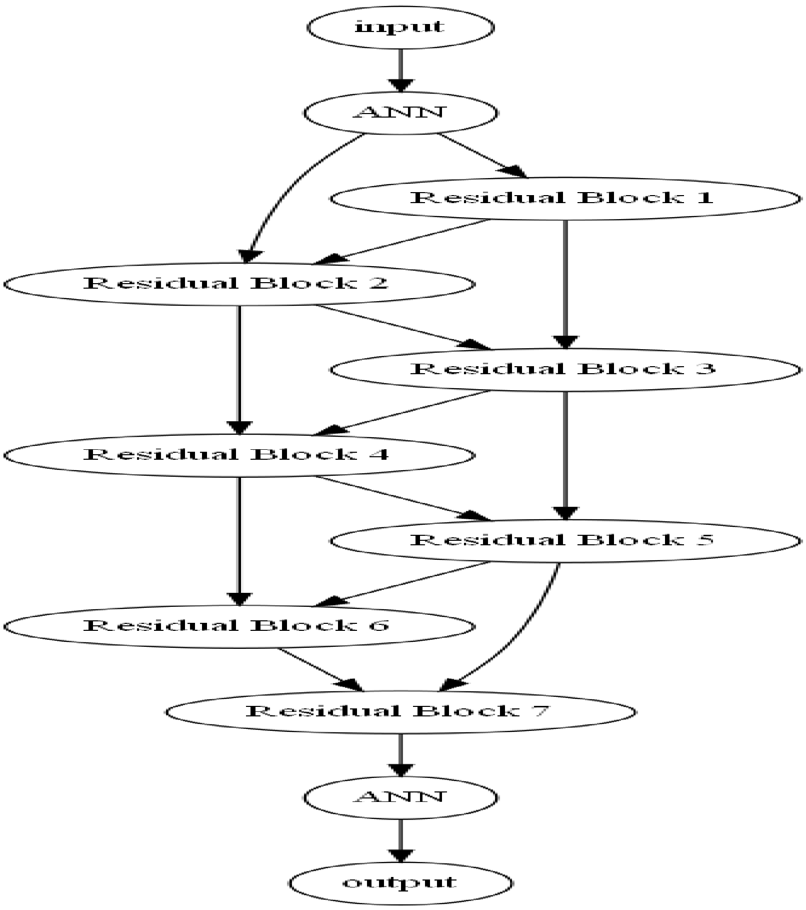

Figure 5.4: ResCNN

In conclusion, the ResNet architecture offers several advantages, including the ability to train very deep neural networks without encountering vanishing gradients, improved convergence rates, and lower error rates. Additionally, the use of parameter sharing reduces the number of parameters required and memory usage, allowing for the training of larger networks with limited computational resources.

5.4.6 Experience Management

In RL, experience refers to the data obtained by the agent interacting with the environment, including information such as the current state, the action taken, the reward received, and the next state. Typically, an experience can be represented as a tuple (s, a, r, s'), where s represents the current state, a represents the action taken, r represents the immediate reward obtained, and s' represents the next state entered after taking action. Multiple experiences are collected to form an experience pool, also called a replay buffer, which is used for the agent to learn and make decisions. The method of computing by aggregating the experiences obtained from multiple policy evaluations is called aggregation computing.

During network training, our RL algorithm randomly samples a certain number of experiences from the experience pool for updates to avoid overly focusing on recent experiences and to make the training more random and diverse. At the same time, the capacity of the experience pool is also limited. When the experience pool is full, new experiences will overwrite old experiences to ensure that the data in the experience pool is always the most up-to-date and representative. Through this approach, we can efficiently utilize previous experiences and gradually optimize the policy and value networks during training.

In this study, the training dataset consists of eight currency pairs spanning from 2006 to 2017. Using a DC threshold of 0.5%, the trainting set contains nearly 100,000 DC events.

Directly training the trading agent on the entire dataset from the initial point would generate a large amount of experience, resulting in substantial computational costs and a potential risk of overfitting to a specific market trajectory.To address this issue, we divide the continuous market environment into multiple independent sub-environments, which we refer to as ”chip”.

A chip is not an individual experience sample, but a complete short-term simulation environment extracted from the original market data. Specifically, a random starting point is selected from the testing dataset, and the subsequent 500 events from the minimum DC framework ($E_{\theta=0.25}$) are used to construct one chip. Within each chip, the trading agent interacts with the market environment from an initial state and generates a complete sequence of experiences, including states, actions, rewards, and trading outcomes. Therefore, each chip represents an independent market episode with different market conditions.

During training, multiple chips are executed in parallel to efficiently generate diverse trading experiences. In our experiment, 36 chips are processed simultaneously due to the limitation of available CPU cores. After completing one simulation round, the experiences generated from all chips are collected and stored in the experience pool, from which the policy network is updated. This approach differs from the conventional experience replay mechanism, where experiences are typically generated from a single continuous environment and randomly sampled for training. Instead, our method first introduces diversity at the environment level by generating multiple independent market trajectories and then applies experience replay to improve training stability.

Furthermore, this approach is different from multi-agent reinforcement learning, where multiple agents independently learn and interact with the environment. In our framework, all chips share the same trading agent and policy network; the parallel execution only serves to provide diverse training experiences and accelerate data collection.

When the experience pool exceeds 500,000 transitions, experiences are randomly removed to maintain a fixed memory size. The performance obtained from each chip is also used as an evaluation measure of the current policy, reflecting its robustness under different market conditions. This procedure is repeated until the trading performance converges to a stable positive level. The overall experience generation and management process is illustrated in Figure 5.5.

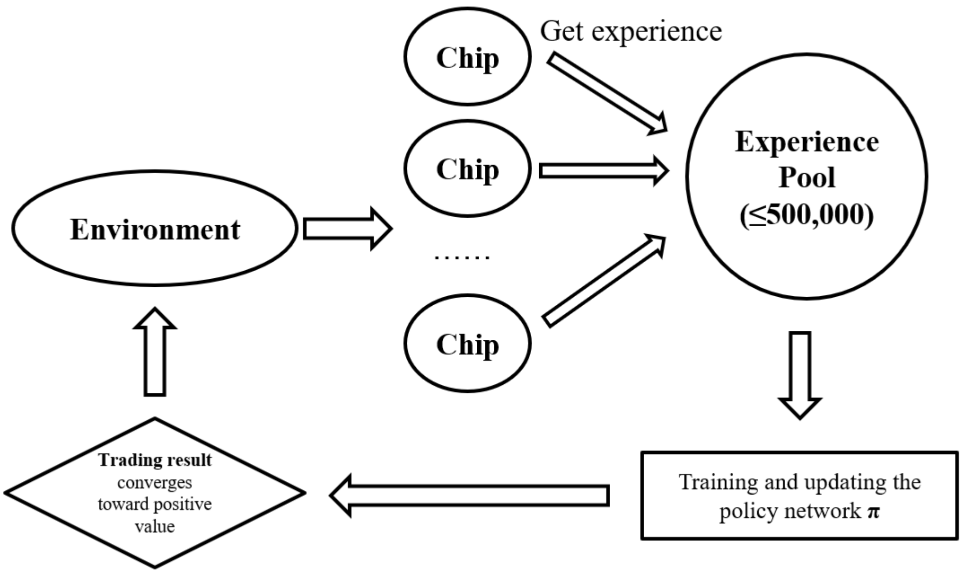

Figure 5.5: the process of Gain experience and policy improvement.

5.4.7 CPU Parallel Technology

The hardware configuration employed in this study comprises a 32-core i9 CPU and a 4090 GPU. To accelerate the model training process and enhance computational efficiency, we adopted CPU parallel computing and multithreading techniques. By decomposing large-scale computational tasks into smaller subtasks that are executed concurrently across different CPU cores, the full processing potential of the multi-core CPU is leveraged, thereby significantly reducing the computational time required for complex calculations—particularly in scenarios characterized by large datasets or computationally intensive tasks.

Drawing on successful strategies in the reinforcement learning domain, such as the Asynchronous Advantage Actor-Critic (A3C) algorithm proposed by Mnih et al. (2016) [181], our study modularizes policy evaluation and policy improvement into independent tasks that are executed in parallel. Specifically, during the policy evaluation phase, the state space or sampling trajectories are partitioned into multiple subsets, with individual threads concurrently computing the value function for their respective segments. This approach maximizes the utilization of available CPU cores. Subsequently, once all parallel evaluation tasks are completed, a unified policy improvement step is executed—employing, for instance, greedy updates—which substantially reduces the overall iteration time.

In the multithreaded execution framework, the program spawns multiplethreads at startup, with each thread independently processing a portion of the data or computational tasks (policy evaluation). Use asynchronous mechanisms to aggregate calculation results and perform policy improvement. During the synchronization phase, the use of locking mechanisms or lock-free data structures ensures data consistency, guaranteeing that the system maintains model accuracy and reliability while significantly reducing training time.

As illustrated in Figure 5.6, the implementation of CPU parallel computing markedly enhances training speed. This observation validates the advantages of modularizing the policy evaluation and policy improvement tasks for independent parallel execution, and it demonstrates the method's effectiveness in improving system time efficiency in scenarios involving large-scale datasets and high iteration counts.

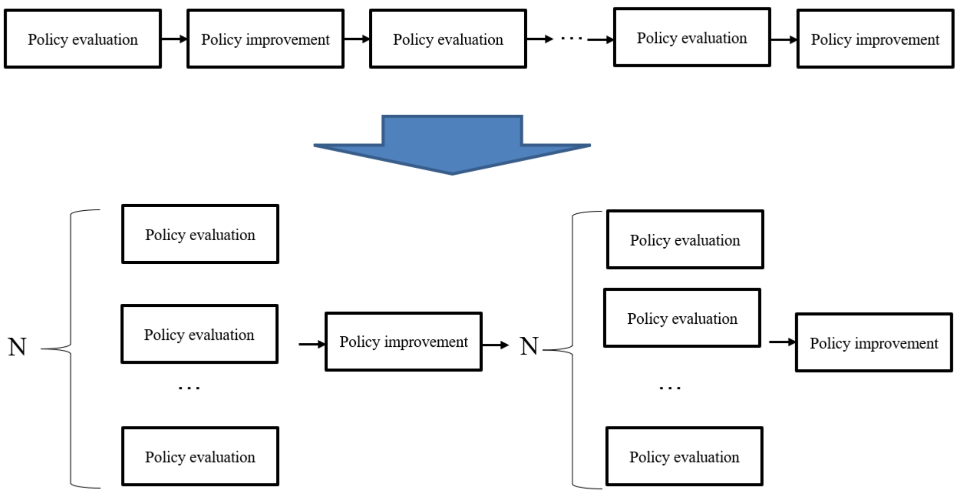

Figure 5.6: the introduction of CPU parallel technology on policy iteration. Where N is the number of the core of the CPU. In this study, N is equal to 36

Due to the use of aggregation operation, although the training data volume of each iteration is increased by N times, the required time is only increased by double size. And the use of CPU multi-threading allows for 36 policy evaluations to be performed in the time it took to perform a single policy evaluation before. The acceleration effect of CPU parallel technology can be expressed as:

$$speedup = \frac{a/n + 2*b}{a*n+b} \tag{5.13}$$

Where speedup means the speed increase magnification, a represents the time required to run a task on a single CPU core, n is the number of the CPU core, and b represents the original time required to run the training task on GPU. The formula above the equation denotes the computational time after employing CPU parallel technology, whereas the formula below the equation represents the original computational time. If the speedup ratio is s, it means that the task runs s times faster than the original process of policy iteration. In our case, the time required for a and b is very similar. So the speedup is equal to approximately 21 times the original version.

5.5 Experiments

We analyzed eight currency pairs from the market to form our dataset. In the realm of global FX, approximately 180 currencies coexist. Theoretically, any two currencies can be coupled to form a currency pair; however, only a subset of these pairs is typically available for trading on most platforms. For the purposes of our analysis, we selected eight currency pairs from the market to serve as our dataset. This selection encompasses four primary currency pairs and an additional four minor or 'cross-currency' pairs that do not involve the US dollar. The chosen pairs are as follows: 'AUDNZD', 'EURJPY', 'EURUSD', 'NZDJPY', 'USDCHF', 'USDCAD', 'USDJPY', and 'EURGBP'. This selection was made considering the trade volume and diversity of the currencies. All the FX market data is obtained from the "https://www.histdata.com" website. The data spans a 15-year period from 2006 to 2020. We allocate the first ten years (2006-2016) for training our RL algorithm to create a trading agent. The remaining four years (2017-2020) serve as the testing phase, where we apply the RL-derived agent to guide DC-based trading strategies for the eight currency pairs. The outcomes, compared against three benchmark trading strategies (the MA strategy, the B&H strategy, and FOS), are presented in this section.

In this section, we introduce and evaluate the results obtained from the experiments conducted through the following steps. Firstly, we collect and preprocess historical minute-level trading data and transform it into DC event sequence data. Secondly, we train the RL model with specified hyperparameters on the training set. Thirdly, we perform multiple tests on the selected training set, which consists of eight currency pairs, to evaluate the performance of the RL agent. The results are presented in Figure 5.8. Fourthly, we compare our novel trading strategy with three established benchmarks: The Moving Average (MA) strategy, the Buy & Hold (B&H) strategy, and a previously developed strategy named FOS.

Compared to other RL-based trading strategies operating on time series data, the trade execution points within our strategy are actually dictated by the structure of the DC framework. The principal objective of RL here is to regulate the size of each position opened, with the aim of optimizing the returns of the Trend-following DC strategy while mitigating the risk of maximum drawdown. The primary baselines for comparison in this paper are derived from the DC strategy and the most well-known trading strategies.

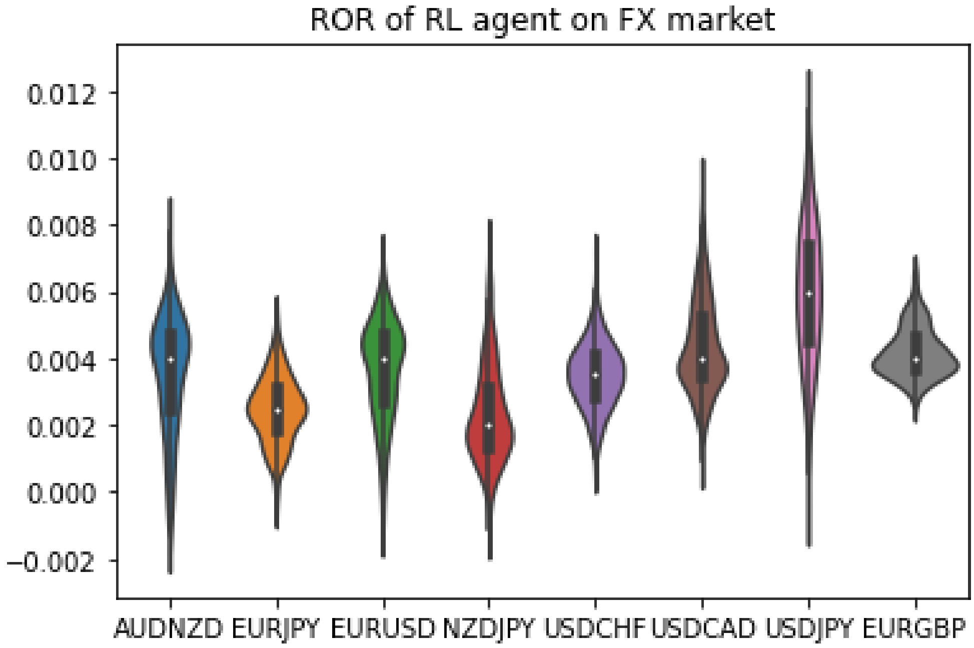

Figure 5.7: The performance of RL agent on FX market

Specifically, after completing the RL model training, we first conducted evaluations on the test dataset. We randomly selected a specific time point of one currency pair from the test set and executed our RL agent. We recorded the final Return on Investment over the period of running the trading agent for 200 trading events. The duration of 200 OS events is approximately one month, so the results shown in Figure 5.8 represent monthly returns rather than annualized trading ROR. We discarded the recording if the selected time point did not reach 200 trading events. Since the test set consists of 8 currency pairs, we categorized the ROR results into 8 groups and visualized them using violin plots (Figure 5.8). Notably, 80% of the runs resulted in positive ROR, with an average return of approximately 0.3% to 0.5% for each currency pair. This result highlights the stability and profitability of our model.

Subsequently, we proceeded to compare our novel trading strategy with three established benchmarks. The approach involved running all four trading models on the historical price datasets of eight currency pairs from 2017 to 2020. The average rate of return over the four-year period was then plotted as a yield curve (Figure 5.8). Here, we provide a concise overview of the three benchmark strategies.

The MA trading strategy is a popular method based on time series analysis, employed widely in financial markets due to its simplicity and efficacy. The strategy operates by computing the average asset price over a specific period, creating a series of averages. When the asset price crosses the moving average line, it signals potential buying or selling opportunities. MA trading strategy is widely used in financial markets for its simplicity and efficacy [37, 184–188].

The Buy & Hold strategy (B& H), by contrast, is a long-term investment strategy predicated on the belief that financial markets will provide a good rate of return over the long run. It involves buying and holding assets for an extended period, regardless of fluctuations in the market. This approach is considered as a passive investment strategy, which has been the subject of numerous studies due to its simplicity and historical success [137, 185, 186]. This is also a very common benchmark strategy [37, 137, 187].In our research, the 'buy and hold' strategy is still introduced as the market return of measuring the overall performance of the basket of currency pairs during the test period.

In the prior chapter, I developed a DC-based simple trend-following strategy, which I have denoted as the Forecasting Overshoot (FOS) strategy. This strategy leverages a ML model to predict the duration of OS events, informing the decision to open a position. The ANN model demonstrated superior performance among five tested ML algorithms. Therefore, in the present study, we employ the ANN-based FOS strategy as an additional benchmark for comparative purposes.

These methods serve as robust benchmarks for our new trading strategy. By comparing the performance of our proposed strategy with these established approaches, we aim to elucidate its potential advantages and disadvantages, thereby providing valuable insights into its applicability in the real world.

5.5.1 Trading Strategies Performance Evaluation

To evaluate the performance of these DC-based trading strategies, we need to measure profitability and risk using certain dimensions. The rate of return (ROR) and maximum drawdown (MDD) are the main evolution indicators in my report. The definition of return, shown in Eq5.8 in section2, is the accumulated profit or loss during our trading period. C_0 is the pre-set initial capital and is set as 100000\$. This setting amount also satisfies the condition for an unleveraged trading model.

In this report, the transaction cost is considered as 0.3% of the quantity of Forex available for trading. The assumption of a 0.3% transaction cost in this study is based on practical FX market observations and supported by existing literature. Transaction costs in retail FX trading mainly consist of bid-ask spreads, commissions, and slippage. Studies such as Bessembinder [189] and Chaboud et al. [190] emphasize that retail traders often face relatively high costs due to larger spreads and occasional slippage. Menkhoff et al. [191] further noted that implicit costs typically range between 0.2% and 0.5% for most currency pairs, depending on market conditions. Setting the transaction cost at 0.3% reflects the midpoint of this range, making it a reasonable and practical estimate for moderately liquid currency pairs, such as EUR/USD and GBP/USD. This value captures realistic trading conditions while ensuring generalizability for evaluating trading strategies.

While the ROR effectively gauges a model's ultimate performance, it falls short of capturing the inherent market risks. To address this deficit and offer a more comprehensive representation of the strategy's risk profile, we incorporate the MDD measure. MDD, a risk metric, quantifies the maximum loss from a peak to a trough of an asset, thus illuminating the strategy's susceptibility to substantial downturns. Mathematically, MDD shown in Eq (5.14) is :

$$MDD = \max_{t_1 > t_2} \frac{R_{t_1} - R_{t_2}}{R_{t_1}} \tag{5.14}$$

where t_1 and t_2 are the point of the return of peak and through. R_{t_1} is the return of peak and R_{t_1} is the return of through. It is measured by calculating the maximum observed loss from a peak value of return to a trough before a new peak is reached.

5.5.2 Experimental Results

This section offers a comparative performance evaluation of our RL strategy grounded in the DC framework against three benchmark strategies: the MA strategy, the B&H strategy, and the FOS strategy, which is likewise based on the DC framework. The effectiveness and consistency of the current RL ML framework are underscored through this analysis. The selected test set encompasses eight specified currency pairs over the period from 2017 to 2020. Figure 5.8 illustrates the average rate of return, calculated by averaging the results across the eight currency pairs.

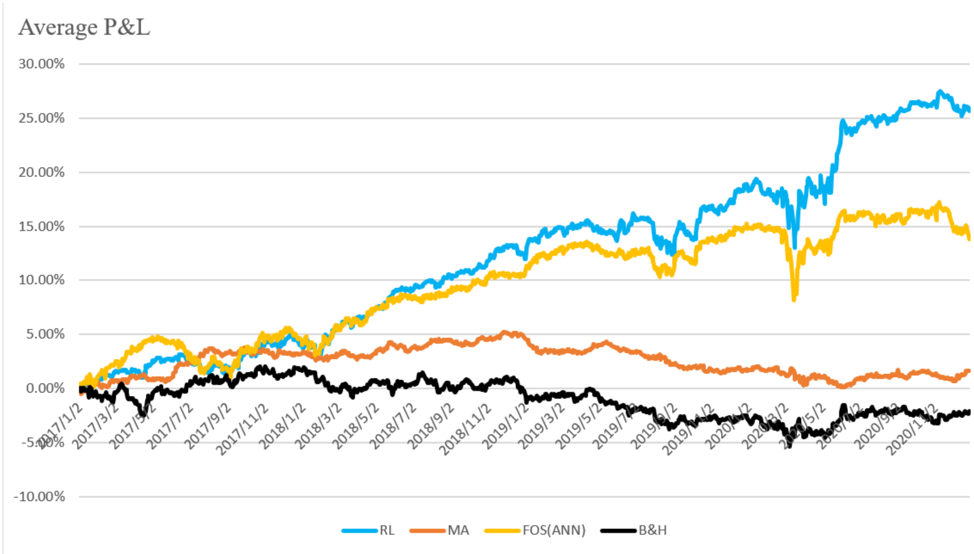

Figure 5.8: Average Profit & Loss for our RL strategy based on the DC framework, MA strategy, FOS strategy, and the 'B&H' strategy on the FX market from 2017 to 2020

The proposed RL trading model was subjected to trials in the FX market. We selected eight test sets, each representing a distinct currency pair, over a period spanning from 2017 to 2020. The RL trading strategy, based on the DC framework, generated positive returns during all test intervals, with an average return reaching 25% over the four-year period. When benchmarked against the B&H strategy, negative returns were noted in four of the test currency pairs, indicating a market downturn with an average negative return of -3.48%. However, our DC-framework-based RL strategy, together with the Forecast Overshoot (FOS) (ANN) strategy and MA strategy, all yielded positive average returns, thereby achieving excess returns under market downturns. The proposed DC-framework-based RL strategy exhibited the highest profitability during the selected test periods and achieved positive returns across all currency pairs. Furthermore, when compared to the previously proposed FOS (ANN) strategy, the RL-based strategy demonstrated a reduced Maximum Drawdown, indicative of lower risk exposure. The superior performance of the RL strategy and FOS (ANN) strategy over the traditional MA strategy and B&H strategy is clearly reflected in the yield curve.

	RL		FOA		MA		B&H	
	P&L(%)	MDD(%)	P&L(%)	MDD(%)	P&L(%)	MDD(%)	P&L(%)	MDD(%)
AUD/NZD	28.23	2.85	18.95	3.15	17.92	4.58	5.18	8.91
EUR/GBP	34.85	6.51	27.47	7.24	13.28	6.59	2.17	7.52
EUR/JPY	35.16	10.15	8.49	11.24	-9.66	14.13	2.18	6.18
EUR/USD	14.37	14.16	16.58	7.25	3.72	11.47	13.15	8.54
NZD/JPY	23.70	18.59	5.52	28.49	3.08	23.85	-4.61	11.72
USD/CAD	27.63	18.42	18.98	9.57	1.66	6.49	-9.43	16.84
USD/CHF	7.20	11.57	9.44	7.39	-1.9	5.21	-15.81	16.41
USD/JPY	30.81	5.18	-1.31	15.26	-13.64	11.39	-13.16	11.75
Average	25.24	6.17	13.65	5.81	1.81	5.11	-3.48	5.63

Table 5.1: Trading strategies evaluation table for FX market (2017-2020)

Additionally, performance varied among the different currency pairs. Compared to the previously proposed FOS (ANN) strategy, the DC-based RL strategy yielded more positive P&L results in all currency pairs except USD/CHY and EUR/USD, with particularly notable results in NZD/JPY (from 5.52% to 23.70%). The DC-based RL strategy achieved the highest test returns on EUR/GBP, with a non-leveraged final profit of 34.85%. For USD/JPY, the FOS (ANN) strategy returned a negative result, whereas the DC-based RL strategy achieved nearly 30% positive returns. In summary, the DC-based RL trading strategy not only enhances the profitability of DC-based trading strategies—nearly doubling the final wealth of the second-best performer—but also mitigates drawdown risk in certain aspects. Table 5.1 provides the performance of the DC-based RL strategy and three benchmark strategies across the eight test currency pairs discussed in this paper. These empirical results reinforce the notion that our proposed RL trading strategy presents a compelling case for enhancing profitability and controlling risk in FX market trading. The demonstrated resilience in the face of market downturns signals its potential as a robust trading tool, poised to carve a niche in the world of algorithmic trading.

Robustness Analysis Across Currency Pairs

To further evaluate the robustness of the proposed DC-based RL trading strategy, its performance consistency across different currency pairs was examined. While average profitability provides an overall indication of performance, robustness analysis focuses on whether the strategy can maintain stable results under different market environments and currency characteristics.

As shown in Table 5.1, the proposed RL strategy generated positive returns for all eight currency pairs during the out-of-sample test period from 2017 to 2020. This corresponds to a success rate of 100%, whereas the FOS strategy achieved positive returns in seven out of eight currency pairs (87.5%), the MA strategy in five out of eight currency pairs (62.5%), and the Buy-and-Hold strategy in only four out of eight currency pairs (50%). The ability to consistently generate positive returns across all tested markets suggests that the proposed RL framework is not dependent on the behaviour of a particular currency pair.

In terms of profitability, the RL strategy achieved the highest average return among all benchmark strategies, producing an average P&L of 25.24%, compared with 13.65% for the FOS strategy, 1.81% for the MA strategy, and -3.48% for the Buy-and-Hold strategy. Furthermore, the RL strategy delivered the highest return in six of the eight currency pairs, demonstrating that its superior performance was not driven by a small number of exceptionally profitable markets. Particularly notable improvements were observed for EUR/JPY, NZD/JPY, USD/CAD, and USD/JPY, where the RL framework substantially outperformed all benchmark strategies.

The distribution of returns across currency pairs also indicates stable performance. Although profitability varied among different markets, the strategy remained profitable under all test conditions. The standard deviation of returns across the eight currency pairs was approximately 10.1%, corresponding to a coefficient of variation of 0.40. This moderate level of dispersion suggests that the observed profitability is relatively stable and not concentrated in a small subset of markets.

From a risk perspective, the RL strategy maintained a comparable level of drawdown to the benchmark strategies while achieving substantially higher returns. Although several currency pairs, such as NZD/JPY and USD/CAD, exhibited relatively larger drawdowns, these were accompanied by significantly higher profitability. Consequently, the overall risk-return profile remained favourable compared with the benchmark approaches.

Overall, the out-of-sample results provide evidence that the proposed RL-based trading framework exhibits strong robustness across different FX markets. The consistent profitability observed across all eight currency pairs suggests that the learned policy captures general characteristics of DC-event dynamics rather than pair-specific market patterns. This finding supports the practical applicability of the proposed approach in diverse market conditions.

5.6 Conclusion

In this chapter, a DC-based algorithmic trading strategy enhanced by reinforcement learning was proposed. The primary objective was to investigate whether an event-driven representation of market dynamics could improve the decision-making process of an RL trading agent. The DC framework provided a structured representation of market states based on intrinsic price events, allowing the agent to operate on directional change and overshoot events rather than fixed time intervals. The effectiveness of the proposed approach was evaluated using out-of-sample FX market data from 2017 to 2020. Experimental results showed that the RL-enhanced strategy consistently generated positive returns across all tested currency pairs. Compared with the benchmark strategies, the proposed method achieved substantially higher profitability while maintaining a comparable level of risk. Furthermore, the results suggest that the proposed framework exhibits a reasonable degree of robustness across the tested currency pairs, indicating that its performance is not driven by a single market or specific currency pair.

The primary contributions of this research are summarized as follows. First, this study proposes a DC-based event-driven environment for reinforcement learning, where market states are represented through intrinsic price events rather than fixed time intervals. While previous studies have explored the integration of the DC framework and reinforcement learning, this research specifically investigates how RL can enhance DC-based trend-following strategies by enabling adaptive trading decisions under different market conditions. Second, a two-component reward function is designed to incorporate both short-term trading outcomes and long-term portfolio performance, allowing the agent to evaluate individual actions while considering cumulative trading returns. Third, a ResNet-based policy network is employed to improve the representation capability of the RL agent when processing complex DC-event observations. Finally, a multi-core CPU parallel experience generation mechanism is adopted to improve training efficiency and accelerate policy learning. Collectively, these contributions provide further insights into the integration of reinforcement learning with event-driven DC-based trading strategies and highlight its potential for developing adaptive algorithmic trading models.

Several limitations should also be acknowledged. The proposed framework was evaluated only on eight FX currency pairs and within a specific historical period. Therefore, the generalizability of the results to other asset classes, market regimes, or future market conditions requires further investigation. In addition, the action space was restricted to trading decisions at DC confirmation points, while position sizing remained fixed throughout the experiments.

Future research may extend the framework by allowing the RL agent to optimize position sizing directly as part of the action space. Additional studies may also evaluate the proposed approach across a broader range of currency pairs, longer time horizons, and alternative financial markets. Furthermore, extending the framework to support simultaneous trading across multiple assets may provide additional opportunities for portfolio-level optimization and risk management.

Chapter 6

Conclusions

This thesis investigates the application of the DC framework in algorithmic trading by integrating machine learning and reinforcement learning techniques. The research systematically develops and evaluates DC-based trading strategies, including trend-following and counter-trend approaches, investigates DC-based risk management through stop-loss mechanisms, examines the predictability of DC-defined overshoot events using machine learning models, and explores the use of reinforcement learning for adaptive trading decisions. This chapter summarizes the main research findings, contributions, limitations, and potential directions for future work.

6.1 Research Summary

The research began by investigating the capability of the DC framework to represent financial price movements and its application in trading strategy design. In Chapter 3, two DC-based strategies were developed based on the characteristics of directional change and overshoot events: the trend-following strategy and the counter-trend strategy. Empirical evaluation using multiple FX currency pairs showed that the trend-following strategy achieved higher profitability, while the counter-trend strategy exhibited better risk control characteristics. Furthermore, the introduction of a DC-based dynamic stop-loss mechanism reduced drawdown risk, although it also limited the maximum achievable returns. These results indicate a trade-off between profitability and risk management within DC-based trading strategies.

Chapter 4 extended the DC framework by incorporating machine learning models to investigate whether DC-defined overshoot events contain predictive information. Five machine learning models, including Random Forest, Support Vector Machine, Artificial Neural Network, Long Short-Term Memory, and Convolutional Neural Network, were employed to predict overshoot event characteristics based on DC-related features. The empirical results suggest that machine learning models can extract certain patterns from DC event information; however, the predictive performance remains limited, reflecting the difficulty of forecasting financial market dynamics. These findings highlight both the potential and challenges of applying machine learning techniques to event-driven financial data.

Building upon the DC-based machine learning framework, Chapter 5 investigated the use of reinforcement learning to improve trading decisions within a DC event-driven environment. Instead of directly optimizing DC parameters, this study designed a multi-scale DC state representation that allows the RL agent to observe market information from different event thresholds. A ResNet-based policy network was adopted to process the constructed state representation, and a two-component reward mechanism was introduced to consider both short-term trading outcomes and long-term portfolio performance. The proposed RL strategy was evaluated using out-of-sample FX market data. The empirical results show that the RL-based strategy achieved higher profitability than the benchmark strategies under the tested market conditions, while maintaining a comparable level of risk.

Overall, this thesis provides an empirical investigation into the integration of the DC framework with machine learning and reinforcement learning for algorithmic trading. The findings suggest that DC event representations can provide useful information for modelling financial market dynamics, while reinforcement learning offers a flexible framework for adapting trading decisions based on event-driven market states. However, the observed performance is based on specific FX datasets and historical periods, and further investigation is required to evaluate the general applicability of the proposed approaches across different markets and trading environments.

6.2 Research Contributions

Our research made significant contributions to the field of financial market analysis, particularly in the utilization of the DC Framework and ML techniques for devising trading strategies.

(1) **Analysis of DC-based Trading Strategies:** This research provides a systematic com-

parison between DC-based trend-following DC-TF and DC-CT strategies. Through empirical analysis across multiple FX currency pairs, this study demonstrates their different risk-return characteristics: DC-TF provides higher return potential, while DC-CT generally exhibits better risk control. This analysis improves the understanding of how DC events can be used for constructing different trading strategies.

(2) **Development of a DC-based Risk Management Mechanism:** A dynamic stop-loss mechanism based on DC characteristics is introduced to improve risk management. The results show that the proposed mechanism can reduce drawdowns and improve risk-adjusted performance, although with a reduction in absolute profitability. This provides further evidence regarding the role of DC information in controlling trading risk.

(3) **Investigation of ML-based OS Event Prediction:** This research investigates whether DC-derived features contain predictive information about future OS events. Five machine learning models, including RF, SVM, ANN, LSTM, and CNN, are applied to classify OS event characteristics. The results show that ML models can extract useful patterns from DC events, while their predictive capability remains constrained by the noisy nature of financial markets. This analysis provides insights into the limitations and potential applications of ML methods within the DC framework.

(4) **Development of an RL-based DC Trading Framework:** This thesis proposes an RL-based trading framework that models the financial market as a sequence of DC events and enables adaptive trading decisions. Compared with previous DC-based approaches that mainly focus on threshold optimization or prediction tasks, the proposed method directly optimizes trading actions within the DC environment. The empirical results demonstrate improved profitability and risk performance on the selected FX test dataset.

(5) **Improvement of RL Training Efficiency:** To address the computational requirements of RL training, a multi-core parallel experience generation mechanism is implemented. This approach reduces training time and improves computational efficiency, facilitating the application of RL methods to event-driven financial environments.

Collectively, this thesis extends the application of the DC framework by integrating machine learning and reinforcement learning techniques. The empirical findings indicate that

DC-based representations can provide useful information for financial trading models, while also revealing the limitations associated with market dependency, prediction uncertainty, and dataset selection. These findings provide directions for further investigation of adaptive trading strategies based on event-driven financial representations.

6.3 Research Limitations

Despite the empirical findings obtained in this thesis, several limitations should be acknowledged. First, the validation of the proposed DC-based strategies is mainly conducted on the foreign exchange (FX) market. Although the FX market is suitable for evaluating event-driven trading strategies due to its high liquidity and near-continuous trading characteristics, the conclusions obtained from this market may not directly generalize to other asset classes. The effectiveness of DC-based representations and trading strategies in equity, commodity, or fixed-income markets requires further empirical investigation.

Second, the empirical evaluation is based on specific historical periods and a limited number of currency pairs. Although the selected datasets include different market conditions, including periods of high volatility, the results may still be affected by particular market regimes. Therefore, the robustness of the proposed strategies under different economic cycles, structural market changes, and future unseen market conditions remains to be further examined.

Third, the performance of DC-based strategies depends on the selection of DC thresholds. Although multiple θ_{DC} values are considered in this thesis, the threshold settings are predefined rather than dynamically adjusted according to changing market conditions. Future research could investigate adaptive threshold selection mechanisms or alternative DC event construction methods to improve the flexibility of the framework.

Fourth, the machine learning models developed in this thesis focus on predicting characteristics of DC-defined Overshoot (OS) events using price-based DC features. The predictive performance of these models remains limited by the available information and feature representation. Incorporating additional information sources, such as macroeconomic variables, market microstructure data, order-flow information, or alternative financial data, may provide further improvements in OS event prediction.

Fifth, the reinforcement learning framework proposed in this thesis adopts a simplified trading environment to focus on the learning capability of the agent under DC event-driven states. The action space, position management, and execution mechanisms are simplified compared with real-world trading systems. In particular, transaction costs, market impact, liquidity constraints, and dynamic portfolio allocation are not fully incorporated, which may influence the practical performance of the strategy.

Finally, although the results demonstrate that the proposed RL-based DC strategy achieves competitive performance against benchmark strategies within the evaluated FX datasets, the findings should not be interpreted as evidence of universal superiority over existing trading approaches. Further validation across broader markets, longer periods, and more realistic trading environments is necessary to assess the practical applicability and robustness of the proposed framework.

These limitations provide several directions for future research, including the development of adaptive DC event generation methods, richer reinforcement learning action spaces, more realistic transaction models, and applications to other financial markets.

6.4 Future Work

The findings of this thesis suggest several directions for future research. These directions aim to further investigate the applicability of the DC framework, improve the learning capability of AI-based trading models, and develop more realistic algorithmic trading systems.

- **Adaptive DC Event Representation and Threshold Selection**: In this thesis, DC events are generated using predefined threshold values. Although multiple θ_{DC} settings are investigated, the selection of thresholds is still determined prior to the trading process. Future research could explore adaptive threshold selection mechanisms, where DC parameters dynamically adjust according to changing market conditions, volatility regimes, or market structures. Such approaches may further improve the flexibility of DC-based market representation.

- **Improvement of Machine Learning-based OS Event Prediction**: The ML models investigated in this thesis demonstrate the potential of using DC features to predict characteristics of Overshoot (OS) events, while their predictive performance remains limited. Future research could explore more advanced feature construction methods, alternative machine learning architectures, and additional market information, including macroeconomic indicators, order-flow information, and cross-market signals, to improve the prediction of DC-defined events.

- **Expansion of Reinforcement Learning Trading Framework**: The proposed RL framework demonstrates competitive performance within the evaluated FX datasets. However, the current environment adopts a simplified action space and trading mechanism. Future work could extend the framework by allowing the RL agent to determine position sizing, portfolio allocation, and more flexible trading actions. Furthermore, incorporating realistic transaction costs, liquidity constraints, and market impact models would provide a closer representation of real-world trading environments.

- **Advanced Neural Network Architectures for DC-based Financial Modelling**: This thesis employs a ResNet-based policy network to process DC event representations. Future studies could investigate alternative neural network architectures specifically designed for event-driven financial data, including models that integrate temporal information, multi-scale market patterns, and heterogeneous financial information. Such developments may provide further insights into how different representations of financial markets affect learning performance.

- **Validation across Different Markets and Asset Classes**: The empirical evaluation in this thesis focuses mainly on the FX market. Future research could extend the investigation to other asset classes, including equities, commodities, and digital assets, to examine whether DC-based representations and trading strategies maintain their effectiveness under different market structures. In particular, further investigation is required to understand how DC scaling properties vary across different financial markets and how such variations affect strategy performance.

- **Multi-Asset and Portfolio-level Trading Strategies**: The current research evaluates trading strategies mainly on individual currency pairs. Future work could investigate portfolio-level decision making, where an RL agent simultaneously manages multiple assets or trading targets. Such extensions may provide further understanding of diversification effects and risk-adjusted performance in DC-based trading systems.

Overall, future research should focus on improving the adaptability, realism, and generalizability of DC-based artificial intelligence trading frameworks. The directions above provide opportunities to further examine the interaction between event-driven market representation, machine learning, and reinforcement learning within financial decision-making systems.

Bibliography

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

[2] D. Lavbič and M. Bajec, "Rapid development of executable ontology for financial instruments and trading strategies," in *Software Engineering and Computer Systems: Second International Conference, ICSECS 2011, Kuantan, Pahang, Malaysia, June 27-29, 2011, Proceedings, Part I 2*. Springer, 2011, pp. 232–243.

[3] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," *Quantitative Finance*, vol. 1, no. 2, pp. 223–236, 2001.

[4] S. Gu, B. Kelly, and D. Xiu, "Empirical asset pricing via machine learning," *The Review of Financial Studies*, vol. 33, no. 5, pp. 2223–2273, 2020.

[5] A. W. Lo and A. C. MacKinlay, "Econometric challenges in empirical finance: Implications for risk management and financial engineering," *Journal of Econometrics*, vol. 45, no. 1-2, pp. 181–198, 1990.

[6] A. N. Kercheval and Y. Zhang, "Modelling high-frequency limit order book dynamics with support vector machines," *Quantitative Finance*, vol. 15, no. 8, pp. 1315–1329, 2015.

[7] J. L. Ticknor, "A bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, vol. 40, no. 14, pp. 5501–5506, 2013.

[8] B. Kelly, D. Xiu *et al.*, "Financial machine learning," *Foundations and Trends® in Finance*, vol. 13, no. 3-4, pp. 205–363, 2023.

[9] B. M. Henrique, V. A. Sobreiro, and H. Kimura, “Literature review: Machine learning techniques applied to financial market prediction,” *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.

[10] S. Almahdi and S. Yang, “An adaptive financial trading system using deep learning and reinforcement learning,” *Expert Systems with Applications*, vol. 87, pp. 267–273, 2017.

[11] B. Hambly, R. Xu, and H. Yang, “Recent advances in reinforcement learning in finance,” *Mathematical Finance*, vol. 33, no. 3, pp. 437–503, 2023.

[12] 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.

[13] R. B. Olsen, “An intrinsic time scale for asset prices,” *Applied Mathematical Finance*, vol. 7, no. 4, pp. 345–361, 2000.

[14] M. R. King and D. Rime, “The global fx market: a review of market structure, participants and trading activity,” *BIS Quarterly Review*, vol. 2011, no. December, pp. 27–43, 2011.

[15] Q. Zhao, “DC_trading,” 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8342818

[16] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, “Visual simultaneous localization and mapping: a survey,” *Artificial intelligence review*, vol. 43, no. 1, pp. 55–81, 2015.

[17] O. Kodongo and K. Ojah, “The dynamic relation between foreign exchange rates and international portfolio flows: Evidence from africa’s capital markets,” *International Review of Economics , Finance*, vol. 24, pp. 71–87, 2012.

[18] ——, “Real exchange rates, trade balance and capital flows in africa,” *Journal of Economics and Business*, vol. 66, pp. 22–46, 2013.

[19] C. Li, Z.-W. Su, T. Yaqoob, and Y. Sajid, "Covid-19 and currency market: a comparative analysis of exchange rate movement in china and usa during pandemic," *Economic Research-Ekonomska Istraživanja*, vol. 35, no. 1, pp. 2477–2492, 2022.

[20] B. D. Krušković, "Central bank intervention in the inflation targeting," *Journal of Central Banking Theory and Practice*, vol. 11, no. 1, pp. 67–85, 2022.

[21] F. Aslam, S. Aziz, D. K. Nguyen, K. S. Mughal, and M. Khan, "On the efficiency of foreign exchange markets in times of the covid-19 pandemic," *Technological forecasting and social change*, vol. 161, p. 120261, 2020.

[22] J. Coakley, M. Marzano, and J. Nankervis, "How profitable are fx technical trading rules?" *International Review of Financial Analysis*, vol. 45, pp. 273–282, 2016.

[23] N. Zarrabi, S. Snaith, and J. Coakley, "Fx technical trading rules can be profitable sometimes!" *International Review of Financial Analysis*, vol. 49, pp. 113–127, 2017.

[24] P.-H. Hsu, M. P. Taylor, and Z. Wang, "Technical trading: Is it still beating the foreign exchange market?" *Journal of International Economics*, vol. 102, pp. 188–208, 2016.

[25] T. A. Maurer, L. Pezzo, and M. P. Taylor, "Importance of transaction costs for asset allocations in fx markets," *Available at SSRN 3143970*, 2020.

[26] J. James, I. Marsh, and L. Sarno, *Handbook of exchange rates*. John Wiley , Sons, 2012.

[27] B. G. Malkiel, *A Random Walk Down Wall Street: The Time-Tested Strategy for Successful Investing*. W. W. Norton , Company, 1973.

[28] S. B. Achelis, "Technical analysis from a to z," 2001.

[29] W. F. Sharpe, "The sharpe ratio," *Streetwise–the Best of the Journal of Portfolio Management*, vol. 3, pp. 169–185, 1998.

[30] M. Rubinstein, "Markowitz's" portfolio selection": A fifty-year retrospective," *The Journal of finance*, vol. 57, no. 3, pp. 1041–1045, 2002.

[31] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[32] C. Krauss, X. A. Do, and N. Huck, "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s ,p 500," *European Journal of Operational Research*, vol. 259, no. 2, pp. 689–702, 2017.

[33] J. Y. Campbell, A. W. Lo, A. C. MacKinlay, and R. F. Whitelaw, "The econometrics of financial markets," *Macroeconomic Dynamics*, vol. 2, no. 4, pp. 559–562, 1998.

[34] M. Hsieh, "On asymptotic log-optimal buy-and-hold strategy," *arXiv preprint arXiv:2103.04898*, 2021.

[35] M. Hsieh *et al.*, "The impact of execution delay on kelly-based stock trading: High-frequency versus buy and hold," *arXiv preprint arXiv:1907.08771*, 2019.

[36] C. H. Park and S. H. Irwin, "What do we know about the profitability of technical analysis?" *Journal of Economic Surveys*, vol. 21, no. 4, pp. 786–826, 2007.

[37] W. Brock, J. Lakonishok, and B. LeBaron, "Simple technical trading rules and the stochastic properties of stock returns," *The Journal of finance*, vol. 47, no. 5, pp. 1731–1764, 1992.

[38] J.-S. Chen, J.-L. Hou, S. Wu, and Y. Chang-Chien, "Constructing investment strategy portfolios by combination genetic algorithms," *Expert System With Application*, vol. 36, pp. 3824–3828, 2009.

[39] H. Zhu and G. Zhou, "Technical analysis: An asset allocation perspective on the use of moving averages," *Journal of Financial Economics*, vol. 92, no. 3, pp. 519–544, 2009.

[40] R. Sullivan, A. Timmermann, and H. White, "Data-snooping, technical trading rule performance, and the bootstrap," *The Journal of Finance*, vol. 54, no. 5, pp. 1647–1691, 1999.

[41] W. Brock, J. Lakonishok, and B. LeBaron, "Simple technical trading rules and the stochastic properties of stock returns," *The Journal of Finance*, vol. 47, no. 5, pp. 1731–1764, 1992.

[42] H. Bessembinder and K. Chan, "The profitability of technical trading rules in the asian stock markets," *Pacific-Basin Finance Journal*, vol. 3, no. 2-3, pp. 257–284, 1995.

[43] J. J. Siegel, *Stocks for the long run: The definitive guide to financial market returns , long-term investment strategies*. McGraw-Hill Education, 2021.

[44] J. C. Bogle, *Common sense on mutual funds: New imperatives for the intelligent investor*. John Wiley , Sons, 1999.

[45] M. Potter, "The mean reversion of the real interest rates: Evidence from oecd countries," *The Journal of International Money and Finance*, 2001.

[46] E. Chan, *Algorithmic Trading: Winning Strategies and Their Rationale*. Wiley, 2012.

[47] N. Jegadeesh and S. Titman, "Returns to buying winners and selling losers: Implications for stock market efficiency," *The Journal of Finance*, 1993.

[48] E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst, "Pairs trading: Performance of a relative value arbitrage rule," *The Review of Financial Studies*, 2006.

[49] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 653–664, 2016.

[50] J. A. Sirignano, "Deep learning for limit order books," *Quantitative Finance*, vol. 19, no. 4, pp. 549–570, 2019.

[51] C. R. Harvey and Y. Liu, "Evaluating trading strategies," *Available at SSRN 2474755*, 2014.

[52] R. Pardo, *The evaluation and optimization of trading strategies*. John Wiley & Sons, 2011.

[53] J. B. Glattfelder and R. B. Olsen, "The theory of intrinsic time: A primer," *arXiv preprint arXiv:2406.07354*, 2024.

[54] U. A. Müller, M. M. Dacorogna, R. D. Dave, and et al., "Fractals and intrinsic time: A challenge to econometricians," 1993, unpublished manuscript, Olsen , Associates, Zürich.

[55] B. Mandelbrot and H. M. Taylor, "On the distribution of stock price differences," *Operations research*, vol. 15, no. 6, pp. 1057–1062, 1967.

[56] P. K. Clark, "A subordinated stochastic process model with finite variance for speculative prices," *Econometrica: journal of the Econometric Society*, pp. 135–155, 1973.

[57] D. M. Guillaume, M. M. Dacorogna, R. R. Dave, and et al., "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.

[58] M. Aloud, E. Tsang, R. B. Olsen, and et al., "A directional-change events approach for studying financial time series," *Economics Discussion Paper*, 2011.

[59] G. Galilei, *Dialogue Concerning the Two Chief World Systems*. University of California Press, 1954.

[60] G. K. Zipf, "Human behavior and the principle of least effort," 1949.

[61] V. Pareto, *Cours d'économie politique*. Droz, 1897.

[62] A. Golub, J. B. Glattfelder, and R. B. Olsen, "The alpha engine: Designing an automated trading algorithm," in *High-Performance Computing in Finance: Problems, Methods, and Solutions*, E. Vynckier, J. Kanniainen, J. Keane, and M. A. H. Dempster, Eds. CRC Press, 2017.

[63] M. E. Aloud, "Directional-change event trading strategy: Profit-maximizing learning strategy," in *Seventh International Conference on Advanced Cognitive Technologies and Applications*, 2015.

[64] A. Dupuis and R. Olsen, *High frequency finance: using scaling laws to build trading models*. Wiley, 2012, pp. 563–582.

[65] 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.

[66] E. Masry, Tsang, "Simulating market clearance dynamics under a simple event calculus market model," in *3rd Computer Science and Electronic Engineering Conference*, 2011.

[67] 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.

[68] 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.

[69] A. Kablan and W. L. Ng, "Intraday high-frequency fx trading with adaptive neuro-fuzzy inference systems," *International Journal of Financial Markets and Derivatives*, vol. 2, no. 1-2, pp. 68–87, 2011.

[70] J. Gypteau, F. Otero, and M. Kampouridis, "Directional changes based trading strategy with genetic programming," in *EvoApplications, EvoStar*, A. M. Mora and G. Squillero, Eds. Springer, 2015, pp. 1–12.

[71] A. Bakhach, "Developing trading strategies under the directional change framework," Ph.D. dissertation, University of Essex, 2018.

[72] M. Kampouridis and F. E. Otero, "Evolving trading strategies using directional changes," *Expert Systems with Applications*, vol. 73, pp. 145–160, 2017.

[73] N. Alkhamees and M. Fasli, "Event detection from time-series streams using directional change and dynamic thresholds," in *IEEE International Conference on Big Data (Big Data)*, Boston, MA, 2017, pp. 1882–1891.

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

[75] N. Alkhamees and M. Fasli, “An exploration of the directional change based trading strategy with dynamic thresholds on variable frequency data streams,” in *2017 International Conference on the Frontiers and Advances in Data Science (FADS)*, Xi’an, 2017, pp. 108–113.

[76] ——, “A directional change based trading strategy with dynamic thresholds,” in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Tokyo, 2017, pp. 283–292.

[77] M. Aloud, “Profitability of directional change based trading strategies: The case of saudi stock market,” *International Journal of Economics and Financial Issues*, vol. 6, no. 1, pp. 87–95, 2016.

[78] ——, “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.

[79] M. Aloud and M. Fasli, “Exploring trading strategies and their effects in the fx market,” *Computational Intelligence*, vol. 33, no. 2, pp. 280–307, 2017.

[80] M. Aloud, “Investment opportunities forecasting: Gp-based dynamic portfolio trading system under directional-change framework,” *Journal of Computational Finance*, vol. 22, no. 1, pp. 1–35, 2017.

[81] H. Ao and E. Tsang, “Trading algorithms built with direction changes,” in *IEEE Conference on Computational Intelligence for Financial Engineering and Economics (CIFEr) Conference*, Shenzhen, China, 2019.

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

[83] 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.

[84] T. M. Mitchell, “Artificial neural networks,” *Machine learning*, vol. 45, no. 81, p. 127, 1997.

[85] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[86] B. Li, K. Li, and H. Kanagawa, “A study on reinforcement learning applied to forex trading,” *arXiv preprint arXiv:1908.05632*, 2019.

[87] Z. Jiang and X. Liang, “Deep reinforcement learning for trading,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 653–664, 2017.

[88] A. Almahdi and J. Yang, “A reinforcement learning-based asset allocation model,” *Expert Systems with Applications*, vol. 69, pp. 131–142, 2017.

[89] L. Breiman, “Random forests,” *Machine learning*, vol. 45, pp. 5–32, 2001.

[90] A. Booth, E. Gerding, and F. McGroarty, “Automated trading with performance weighted random forests and seasonality,” *Expert Systems with Applications*, vol. 41, no. 8, pp. 3651–3661, 2014.

[91] A. Si, J. Zhang, Y. Zhang, E. Kazuva, Z. Dong, Y. Bao, and G. Rong, “Debris flow susceptibility assessment using the integrated random forest based steady-state infinite slope method: A case study in changbai mountain, china,” *Water*, vol. 12, no. 7, p. 2057, 2020.

[92] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, pp. 273–297, 1995.

[93] W. Huang, Y. Nakamori, and S.-Y. Wang, “Forecasting stock market movement direction with support vector machine,” *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.

[94] C.-F. Huang, “A hybrid stock selection model using genetic algorithms and support vector regression,” *Applied soft computing*, vol. 12, no. 2, pp. 807–818, 2012.

[95] C. L. Dunis, S. D. Likothanassis, A. S. Karathanasopoulos, G. S. Sermpinis, and K. A. Theofilatos, “A hybrid genetic algorithm–support vector machine approach in the task of forecasting and trading,” *Journal of Asset Management*, vol. 14, pp. 52–71, 2013.

[96] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, “Scikit-learn: Machine learning in python,” *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[97] E. Azoff, *Neural Network Time Series Forecasting of Financial Markets*. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[98] A.-P. Refenes, *Neural Networks in the Capital Markets*. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[99] R. Trippi and E. Turban, *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-world Performance*. Burr Ridge: Irwin Professional Publishing Co., 1996.

[100] G. Zhang, B. Patuwo, and M. Hu, “Forecasting with artificial neural networks: The state of the art,” *International Journal of Forecasting*, vol. 14, pp. 35–62, 1998.

[101] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[102] C. Olah, “Understanding lstm networks,” http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

[103] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *European journal of operational research*, vol. 270, no. 2, pp. 654–669, 2018.

[104] W. Bao, J. Yue, and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory,” *PloS one*, vol. 12, no. 7, p. e0180944, 2017.

[105] M. Kraus, S. Feuerriegel, and A. Oztekin, “Deep learning in business analytics and operations research: Models, applications and managerial implications,” *European Journal of Operational Research*, vol. 281, no. 3, pp. 628–641, 2020.

[106] C. Almeida *et al.*, “High-performance stock index trading: making effective use of a deep lstm neural network,” *arXiv preprint arXiv:1902.03125*, 2019.

[107] C. Bao, R. Chan, and Y. He, “Forecasting directional movements of stock prices for intraday trading using lstm and random forests,” *arXiv preprint arXiv:2004.10178*, 2020.

[108] C. Munoz, M. Gehrig, and other authors, “An optimal control strategy for execution of large stock orders using lstms,” *arXiv preprint arXiv:2301.09705*, 2023.

[109] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[110] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[111] O. B. Sezer and M. M. Ozbayoglu, “Financial trading model with stock bar chart image time series with deep convolutional neural networks,” *Expert Systems with Applications*, vol. 136, pp. 165–177, 2020.

[112] Z. Zhang, S. Zohren, and S. Roberts, “Deeplob: Deep convolutional neural networks for limit order books,” *IEEE Transactions on Signal Processing*, vol. 67, no. 11, pp. 3001–3012, 2019.

[113] H. Lu, Q. Yang *et al.*, “Cnn-lstm structure for modeling financial time series prediction,” *arXiv preprint arXiv:2005.12318*, 2020.

[114] J. Heaton, “Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618,” *Genetic Programming and Evolvable Machines*, vol. 19, no. 1-2, pp. 305–307, 2018.

[115] J.-J. Wang, J.-Z. Wang, Z.-G. Zhang, and S.-P. Guo, "Stock index forecasting based on a hybrid model," *Omega*, vol. 40, no. 6, pp. 758–766, 2012.

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

[117] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[118] ——, "Identity mappings in deep residual networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV*, vol. 14. Springer International Publishing, 2016, pp. 630–645.

[119] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[120] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, and Y. Chen, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[121] J. Moody and M. Saffell, "Learning to trade via direct reinforcement," *IEEE transactions on neural Networks*, vol. 12, no. 4, pp. 875–889, 2001.

[122] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 653–664, 2017.

[123] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley , Sons, 2014.

[124] A. Singh, "Reinforcement learning: Markov decision process (part 1)," *Towards data science*, 2018.

[125] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[126] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[127] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[128] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[129] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[130] B. Li and S. C. Hoi, "On-line portfolio selection with moving average reversion," *arXiv preprint arXiv:1206.4626*, 2012.

[131] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," *arXiv preprint arXiv:1706.10059*, 2017.

[132] F. Bertoluzzo and M. Corazza, "Testing different reinforcement learning configurations for financial trading: Introduction and applications," *Procedia Economics and Finance*, vol. 3, pp. 68–77, 2012.

[133] J. Sun, H. Fujita, P. Chen, and H. Li, "Dynamic financial distress prediction with concept drift based on time weighting combined with adaboost support vector machine ensemble," *Knowledge-Based Systems*, vol. 120, pp. 4–14, 2017.

[134] A. Mosavi, Y. Faghan, P. Ghamisi, P. Duan, S. F. Ardabili, E. Salwana, and S. S. Band, "Comprehensive review of deep reinforcement learning methods and applications in economics," *Mathematics*, vol. 8, no. 10, p. 1640, 2020.

[135] F. Soleymani and E. Paquet, "Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—deepbreath," *Expert Systems with Applications*, vol. 156, p. 113456, 2020.

[136] Y. Ye, H. Pei, B. Wang, P.-Y. Chen, Y. Zhu, J. Xiao, and B. Li, "Reinforcement-learning based portfolio management with augmented asset movement prediction states," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 1112–1119.

[137] B. Li and S. C. Hoi, "Online portfolio selection: A survey," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–36, 2014.

[138] Y. Nevmyvaka, Y. Feng, and M. Kearns, "Reinforcement learning for optimized trade execution," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 673–680.

[139] S. Idowu, D. Strüber, and T. Berger, "Asset management in machine learning: A survey," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 2021, pp. 51–60.

[140] F. Fang, C. Ventre, M. Basios, L. Kanthan, D. Martinez-Rego, F. Wu, and L. Li, "Cryptocurrency trading: a comprehensive survey," *Financial Innovation*, vol. 8, no. 1, pp. 1–59, 2022.

[141] I. Chakraborty and P. Maity, "Covid-19 outbreak: Migration, effects on society, global environment and prevention," *Science of the total environment*, vol. 728, p. 138882, 2020.

[142] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, J. Pineau *et al.*, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[143] S. J. Grossman and Z. Zhou, "Optimal investment strategies for controlling drawdowns," *Mathematical Finance*, vol. 3, pp. 241–276, 1993.

[144] K. M. Kaminski and A. W. Lo, "When do stop-loss rules stop losses?" *Journal of Financial Markets*, vol. 18, pp. 234–254, 2014.

[145] R. W. Colby and T. A. Meyers, “The encyclopedia of technical market indicators,” *McGraw-Hill*, 2002.

[146] F. Guillaume, M. M. Dacorogna, R. Dave, U. A. Müller, R. B. Olsen, O. V. Pictet, and M. Vettori, “Gab: A profitable range-based financial trading system,” *European Physical Journal B-Condensed Matter and Complex Systems*, vol. 1, no. 4, pp. 585–586, 1997.

[147] E. P. Tsang, “Directional change and trading volume,” *Journal of Banking & Finance*, vol. 35, no. 12, pp. 3306–3315, 2011.

[148] M. Covel, “Trend following: how great traders make millions in up or down markets,” *(No Title)*, 2005.

[149] M. Aloud, E. Tsang, and R. Olsen, “Modeling the fx market traders’ behavior: An agent-based approach,” in *Banking, Finance, and Accounting: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2015, pp. 350–384.

[150] 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.

[151] M. E. Aloud, “Profitability of directional change based trading strategies: The case of saudi stock market,” *International Journal of Economics and Financial Issues*, vol. 6, no. 1, pp. 87–95, 2016.

[152] 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.

[153] 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.

[154] M. Aloud, “Investment opportunities forecasting: a genetic programming-based dynamic portfolio trading system under a directional-change framework,” *Journal of Computational Finance, Forthcoming*, 2017.

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

[156] 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.

[157] E. P. Tsang, "Applying the directional-change technique to financial trading," *Decision Support Systems*, vol. 95, pp. 1–8, 2017.

[158] Y. Han, G. Zhou, and Y. Zhu, "Taming momentum crashes: A simple stop-loss strategy," *Available at SSRN 2407199*, 2016.

[159] A. Bernales and M. Guidolin, "Can we forecast the implied volatility surface dynamics of equity options? predictability and economic value tests," *Journal of Banking , Finance*, vol. 46, pp. 326–342, 2014.

[160] A. Brabazon, M. Kampouridis, and M. O'Neill, "Applications of genetic programming to finance and economics: past, present, future," *Genetic Programming and Evolvable Machines*, vol. 21, pp. 33–53, 2020.

[161] T. Chavarnakul and D. Enke, "Intelligent technical analysis based equivolume charting for stock trading using neural networks," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1004–1017, 2008.

[162] A. J. Hussain, D. Al-Jumeily, H. Al-Askar, and N. Radi, "Regularized dynamic self-organized neural network inspired by the immune algorithm for financial time series prediction," *Neurocomputing*, vol. 188, pp. 23–30, 2016.

[163] 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.

[164] C.-H. Park and S. H. Irwin, "The profitability of technical analysis: A review," 2004.

[165] D. Isakov and D. Marti, "Technical analysis with a long term perspective: Trading strategies and market timing ability," in *International Conference of the French Finance Association (AFFI)*, 2011.

[166] J. Moody, M. Saffell, and K. Veeramachaneni, "Reinforcement learning for trading," *Neural Networks*, vol. 11, no. 9, pp. 1513–1524, 1998.

[167] L. Chen and Q. Gao, "Application of deep reinforcement learning on automated stock trading," in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2019, pp. 29–33.

[168] Q.-V. Dang, "Reinforcement learning in stock trading," in *International Conference on Computer Science, Applied Mathematics and Applications*. Springer, 2019, pp. 311–322.

[169] G. Jeong and H. Y. Kim, "Improving financial trading decisions using deep q-learning: Predicting the number of shares, action strategies, and transfer learning," *Expert Systems with Applications*, vol. 117, pp. 125–138, 2019.

[170] S. Gold, "High-frequency trading in the foreign exchange market," *Journal of International Money and Finance*, vol. 22, no. 3, pp. 341–376, 2003.

[171] A. Gabrielsson and P. A. Johansson, "Enhancing technical trading strategies: An application to the omxs30 index," *International Review of Financial Analysis*, vol. 42, pp. 343–354, 2015.

[172] J. Carapuço, M. Oliveira, L. Pádua, and A. Neves, "Investigating deep q-networks for the forex market," *Expert Systems with Applications*, vol. 101, pp. 184–198, 2018.

[173] P. Sornmayura and C. Kaewrak, "Robust forex trading using dqn," *arXiv preprint arXiv:1903.03213*, 2019.

[174] M. Zarkias, G. Siolas, and A. Tsakonas, "Tracking financial market trends using deep reinforcement learning," *arXiv preprint arXiv:1906.08688*, 2019.

[175] H. Leem and K. Kim, “Action-specialized ensemble learning based on deep reinforcement learning for automated stock trading,” *Neurocomputing*, vol. 378, pp. 217–230, 2020.

[176] A. J. García-Galicia, A. Carsteanu, and J. B. Clempner, “Continuous-time reinforcement learning for trading,” *arXiv preprint arXiv:1910.10007*, 2019.

[177] Q. Li, B. Cui, Y. Ding, K. K. Lai, L. Li, and L. Wang, “Dqn and a3c reinforcement learning for financial trading: Combination and performance,” *arXiv preprint arXiv:1905.06334*, 2019.

[178] X. Zhang, S. Zohren, and S. Roberts, “Deep reinforcement learning for trading and beyond,” *Applied Mathematical Finance*, vol. 27, no. 8, pp. 728–754, 2020.

[179] A. W. Lo, H. Mamaysky, and J. Wang, “Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation,” *The journal of finance*, vol. 55, no. 4, pp. 1705–1765, 2000.

[180] C. D. Kirkpatrick II and J. A. Dahlquist, *Technical analysis: the complete resource for financial market technicians*. FT press, 2010.

[181] V. Mnih, “Asynchronous methods for deep reinforcement learning,” *arXiv preprint arXiv:1602.01783*, 2016.

[182] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.

[183] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, 2013.

[184] J.-S. Chen, J.-L. Hou, S.-M. Wu, and Y.-W. Chang-Chien, “Constructing investment strategy portfolios by combination genetic algorithms,” *Expert Systems with Applications*, vol. 36, pp. 3824–3828, 2009.

[185] C.-H. Park and S. H. Irwin, “What do we know about the profitability of technical analysis?” *Journal of Economic Surveys*, vol. 21, no. 4, pp. 786–826, 2007.

[186] H. Zhu and G. Zhou, "Technical analysis: An asset allocation perspective on the use of moving averages," *Journal of Financial Economics*, vol. 92, no. 3, pp. 519–544, 2009.

[187] R. Sullivan, A. Timmermann, and H. White, "Data-snooping, technical trading rule performance, and the bootstrap," *The Journal of Finance*, vol. 54, no. 5, pp. 1647–1691, 1999.

[188] H. Bessembinder and K. Chan, "The profitability of technical trading rules in the asian stock markets," *Pacific-Basin Finance Journal*, vol. 3, no. 2-3, pp. 257–284, 1995.

[189] H. Bessembinder, "Trading costs and return patterns in the us equity markets," *Financial Analysts Journal*, vol. 55, no. 4, pp. 29–42, 1999.

[190] A. P. Chaboud, S. V. Chernenko, E. A. Howorka, and J. H. Wright, "The high-frequency effects of us macroeconomic data releases on prices and trading activity in the foreign exchange market," *International Finance Discussion Papers*, no. 823, 2004.

[191] L. Menkhoff, L. Sarno, M. Schmeling, and A. Schrimpf, "The global role of the us dollar and trade costs: evidence from the fx market," *Journal of International Economics*, vol. 100, pp. 65–75, 2016.