Novel Machine Learning Pipelines with Applications to Finance

Fang, Fan

Awarding institution:
King’s College London

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without proper acknowledgement.

END USER LICENCE AGREEMENT

Unless another licence is stated on the immediately following page this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International licence. https://creativecommons.org/licenses/by-nc-nd/4.0/

You are free to copy, distribute and transmit the work

Under the following conditions:

• Attribution: You must attribute the work in the manner specified by the author (but not in any way that suggests that they endorse you or your use of the work).
• Non Commercial: You may not use this work for commercial purposes.
• No Derivative Works - You may not alter, transform, or build upon this work.

Any of these conditions can be waived if you receive permission from the author. Your fair dealings and other rights are in no way affected by the above.

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Novel Machine Learning Pipelines with Applications to Finance

Fan Fang

Supervisor: Prof. Carmine Ventre
Dr. Maria Polukarov

The Department of Informatics
King’s College London

This dissertation is submitted for the degree of
Doctor of Philosophy

September 2023
I would like to dedicate this thesis to my loving parents and my wife Miss Lu who encouraged me when times were tough . . .
Declaration

We hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Fan Fang
September 2023
Acknowledgements

Undertaking the PhD has been a truly life-changing and challenging experience for me and it would not have been possible to do without the support and guidance that I received from many people.

An immense thank you to my PhD supervisors: Prof. Carmine Ventre and Dr. Maria Polukarov. Support and guidance through the project from you all has been invaluable. Professor Carmine in particular you’ve been a fantastic primary supervisor. Thank you to all the academics who helped me get to this stage. At the same time, I would like to thank my second supervisor Maria, who gave me a lot of help and academic advice when I just entered King’s College London.

It has been an amazing experience working between two research groups in the Informatics department, thanks to everyone from the MT-Finance and MT-DAI groups I’ve worked with and from whom I’ve learned so much. I have learned a lot from the group seminars and reading group meetings, truly appreciate the organizers. Thank you of course to the researchers with hybrid background I’ve communicated and worked together from UCL, IC and Oxford, it’s been great fun exchanging academic experience with you all. Meanwhile, I cherished the peer researchers in Office N-6.07 in Bush House, Xiaoyue, Buhong and Haochen in particular, thanks for the days and nights of studying and discussing academics and having dinner together.

I greatly appreciate the colleagues in Turintech and DataSpartan: Dr. Lingbo Li, Dr. Fan Wu, Dr. Michail Basios and Dr. Leslie Kanthan. Lingbo in particular thanks for you all your help, support and friendship: your inclusivity is appreciated by many. I have learned a lot during my internship in Turintech and DataSpartan, thanks for providing a good platform to practice the content of theoretical research.

I would also like to say a heartfelt thank you to my Mum, Dad, Jianmei Tian and Guoping Fang for always believing in me and encouraging me to follow my dreams. And I appreciate my fiancee Yutong Lu for giving me meticulous care and I will give you the longest love in the rest of my life.
Abstract

Machine learning is an artificial intelligence technique used to automatically infer rules from data, and use these rules to perform some tasks on unknown data. This technique is widely used in the field of finance and other disciplines and is characterised by the combination of massive amounts of data and the powerful computing abilities of modern computers. The surge of cryptocurrency markets with their high fluctuations has challenged both traditional econometrics tools, based on statistics and time series analysis, and machine learning. Investigations into the analysis of cryptocurrency markets and the use of emerging machine learning techniques within are therefore useful for researchers to compare market performance and technological innovation in traditional equity/bond markets and cryptocurrency markets.

The problem within this area is however the wide array of disciplines contributing to the field. Although there exist a wealth of surveys related to the research on blockchain and cryptocurrencies, none is really comprehensive and able to cut across different fields. The first part of the thesis focuses on a widely cited survey on cryptocurrency trading, which informs much academic and industry work in this area. The survey provides an in-depth analysis of the literature from the perspective of research distribution among properties, categories, technologies, datasets, research trends and opportunities.

One of the research directions identified in the survey is the prediction of signals for cryptocurrency markets on live data. We address this research challenge in the second part of the thesis. An important finding of this work is that by using multi-layer architectures, deep learning model, and dynamic retraining methods, we can overcome the decay in predictive power on live data due to non-stationary features of the order book. A new dynamic retraining structure is proposed and compared to existing training frameworks in this part.

In the last part of the thesis, we look more closely at the model selection motivated by the success of regular retraining for cryptocurrency prediction. We study model selection from the first principles and independently from the application domain, with the objective to find techniques that are alternative to cross-validation (which often relies on the absence of temporal relationships in the data). We focus on tree models and use the dispersion of
feature importance as a criterion for model selection. We show how this new method can help us choose models with a better generalisation more efficiently.
Table of contents

List of figures 11

List of tables 13

1 Introduction 15
   1.1 Related Work ................................................. 16
   1.2 Contributions .................................................. 17
      1.2.1 Survey on existing cryptocurrency trading techniques and theories 19
      1.2.2 Order book signs prediction with deep learning techniques .......... 19
      1.2.3 Dispersion of feature importance in selecting tree models .......... 19
   1.3 Thesis Outline ................................................. 20

2 A comprehensive survey on cryptocurrency trading 21
   2.1 Introduction .................................................... 21
   2.2 Cryptocurrency Trading ......................................... 23
      2.2.1 Blockchain .................................................. 23
      2.2.2 Introduction of cryptocurrency market .......................... 25
      2.2.3 Cryptocurrency Trading ..................................... 29
   2.3 Cryptocurrency Trading Strategy ................................. 31
      2.3.1 Cryptocurrency Trading Software System ........................ 31
      2.3.2 Systematic Trading ......................................... 32
      2.3.3 Tools for building automated trading systems .................... 32
      2.3.4 Portfolio Research .......................................... 34
      2.3.5 Market Condition Research .................................. 35
   2.4 Paper Collection and Review Schema ............................ 35
      2.4.1 Survey Scope ................................................ 35
      2.4.2 Paper Collection Methodology ................................ 36
      2.4.3 Collection Results .......................................... 37
      2.4.4 Survey Organisation ......................................... 39
   2.5 Cryptocurrency Trading Software Systems ........................ 40
Table of contents

2.5.1 Trading Infrastructure Systems ........................................ 40
2.5.2 Real-time Cryptocurrency Trading Systems ......................... 43
2.5.3 Turtle trading system in Cryptocurrency market ....................... 43
2.5.4 Arbitrage Trading Systems for Cryptocurrencies ..................... 43
2.5.5 Characteristics of three cryptocurrency trading systems ............. 44
2.6 Systematic Trading ..................................................... 44
   2.6.1 Technical Analysis .................................................. 44
   2.6.2 Pairs Trading ....................................................... 46
   2.6.3 Applicability of using equity trading strategies in cryptocurrency . 46
   2.6.4 Price formation in trading strategies ................................ 48
   2.6.5 Others ............................................................. 50
2.7 Emergent Trading Technologies ............................................ 51
   2.7.1 Econometrics on cryptocurrency ..................................... 51
   2.7.2 Machine Learning Technology ....................................... 54
   2.7.3 Others ............................................................. 64
2.8 Portfolio, Assets and Market Condition .................................. 65
   2.8.1 Research among cryptocurrency pairs and related factors ........ 65
   2.8.2 Crypto-asset Portfolio Research .................................... 68
   2.8.3 Bubbles and Crash Analysis ......................................... 70
   2.8.4 Extreme condition .................................................. 71
2.9 Cryptocurrency asset pricing .............................................. 72
   2.9.1 Factor pricing model ................................................ 72
   2.9.2 Valuation in pricing and market network ............................ 73
   2.9.3 Risk Factors ....................................................... 74
   2.9.4 Advantages and disadvantages among pricing factors .............. 74
2.10 Other work related to Cryptocurrency Trading .......................... 76
2.11 Summary Analysis of Literature Review ................................ 78
   2.11.1 Timeline .......................................................... 78
   2.11.2 Research Distribution among Properties .......................... 79
   2.11.3 Research Distribution among Categories and Technologies ...... 80
   2.11.4 Datasets used in Cryptocurrency Trading ........................ 83
2.12 Opportunities in Cryptocurrency Trading ................................ 83
2.13 Conclusions ............................................................ 87

3 Ascertaining price formation in cryptocurrency markets with deep learning 89
3.1 Introduction ............................................................ 89
   3.1.1 Roadmap ......................................................... 92
3.2 Our tools ............................................................... 92
Table of contents

3.2.1 Machine Learning ................................................. 92
3.2.2 Limit Order Books .................................................. 94
3.2.3 Data source and overview of the envisioned trading system .... 94
3.3 Experimental Study .................................................. 96
  3.3.1 Objective .............................................................. 96
  3.3.2 Dataset ............................................................... 96
  3.3.3 Methodology ......................................................... 97
  3.3.4 Research Questions ................................................ 105
  3.3.5 Results and Analysis .............................................. 106
3.4 Validity of findings .................................................. 121
3.5 Conclusions ............................................................ 122

4 Dispersion of feature importance in tree models .................... 124
  4.1 Introduction .......................................................... 124
  4.2 Related Work ......................................................... 126
  4.3 Methodology .......................................................... 128
    4.3.1 Feature importance in Tree Models .......................... 128
    4.3.2 Prediction Path based feature importance .................. 128
    4.3.3 Two methods to measure utilisation rate of features ...... 130
    4.3.4 Coefficient of Variation of Feature Importance .......... 132
    4.3.5 Entropy Values of Feature Importance ..................... 134
  4.4 Experimental Study .................................................. 136
    4.4.1 Experimental Setup .............................................. 136
    4.4.2 Datasets ........................................................... 136
    4.4.3 Evaluation Criteria .............................................. 136
    4.4.4 Research Questions .............................................. 137
  4.5 Experimental Results ................................................ 138
    4.5.1 RQ1: What is the relationship between feature contribution dispersion and accuracy matrix in model selection? .......... 138
    4.5.2 RQ2: What is the effect to apply our CV based feature contribution dispersion method in model selection compared to $k$-fold cross validation? .................................................. 139
    4.5.3 RQ3: How the entropy values of feature importance can be applied in narrowing parameter tuning space and selecting tree models efficiently? .................................................. 143
  4.6 Threats to Validity ................................................... 148
  4.7 Conclusions ........................................................... 149
  4.8 Supplement figures .................................................... 149
Table of contents

5  Conclusion 154
   5.1  Summary of Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
   5.2  Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155

References 157
## List of figures

2.1 Cryptocurrency Trading Publications Trends ........................................ 22  
2.2 Workflow of Blockchain transaction .................................................... 24  
2.3 Market Capitalization and Volume of cryptocurrency ............................. 26  
2.4 Percentage of Total Market Capitalisation .......................................... 27  
2.5 Process of machine learning in predicting cryptocurrency ..................... 33  
2.6 Publication Venue Distribution ......................................................... 38  
2.7 Timeline of cryptocurrency trading research ....................................... 79  
2.8 Research distribution among cryptocurrency trading properties ............. 80  
2.9 Research distribution among cryptocurrency trading technologies .......... 81  
3.1 An overview of an LSTM cell .............................................................. 93  
3.2 An overview of a simple trading system ............................................. 95  
3.3 LSTM model architecture ................................................................. 98  
3.4 Distribution of historical price changes .............................................. 100  
3.5 Currency pairs without improvement ............................................... 108  
3.6 Currency pairs with improvement .................................................... 109  
3.7 Box plots of currency pair with and without improvement ................... 109  
3.8 Relationship between accuracy and number of time steps used in training 112  
3.9 Performance decay on the the live data ........................................... 113  
3.10 Architecture of the autoencoder ...................................................... 114  
3.11 Performance of the universal model with autoencoder ....................... 114  
3.12 Predictions distribution and real-time target distribution .................... 115  
3.13 Performance of the universal model with autoencoder ....................... 116  
3.14 Predictions distribution and real-time target distribution .................... 116  
3.15 LSTM model with autoencoder ....................................................... 118  
3.16 Snapshot of comparison between different walkthrough methods .......... 120  
3.17 Dynamic Walkthrough comparison with or without extreme conditions .. 121  
3.18 Comparison of different Walkthrough methods .................................. 122  
4.1 Explanation of how prediction is formed by path-based tree prediction ... 130
4.2 Comparison between cross validation and feature contribution dispersion 131
4.3 Workflow in Model selection .......................... 133
4.4 Process of calculating k-fold feature importance .......................... 135
4.5 Workflow in finding the relationship among two metrics .......................... 135
4.6 Performance of comparison between two metrics .......................... 140
4.7 Relationship between entropy values and model generalisation .......................... 143
4.8 Density Plot of entropy values of feature importance .......................... 144
4.9 Density Plot of Train accuracy and Test accuracy .......................... 144
4.10 Scatter Plot of entropy values and train/test accuracy .......................... 145
4.11 Scatter Plot of entropy values and train/test accuracy (Supplement) .......................... 146
4.12 Regional divisions according to entropy values of feature importance .......................... 147
4.13 Process of Pruning models by confidence interval .......................... 147
4.14 Comparison of performance of before/after pruning .......................... 148
4.15 Performance of comparison between two metrics (Supplement1) .......................... 150
4.16 Performance of comparison between two metrics (Supplement2) .......................... 151
4.17 Performance of comparison between two metrics (Supplement3) .......................... 152
4.18 Performance of comparison between two metrics (Supplement4) .......................... 153
List of tables

1.1 A snapshot of this thesis .............................................. 18
2.1 Cryptocurrency exchanges Lists ..................................... 28
2.2 Comparison among Different machine learning methods ......... 34
2.3 Survey scope table .................................................... 36
2.4 Paper query results ................................................... 37
2.5 Review Schema ......................................................... 39
2.6 Comparison of existing cryptocurrency trading systems .......... 42
2.7 Comparison among five classical technical trading strategies .......... 45
2.8 Search hits of research distribution in all trading areas .......... 82
2.9 Datasets (1/3): Market Data ........................................... 84
2.10 Datasets (2/3): Sentiment-based data ................................. 85
2.11 Datasets (3/3): Others ................................................ 86
3.1 Comparison between our research and Easley et al. ............... 91
3.2 Amount of data collected ............................................. 96
3.3 Data statistics .......................................................... 97
3.4 Feature Set ............................................................. 98
3.5 Multi-label prediction .................................................. 100
3.6 Classification report in Logistic Regression ......................... 102
3.7 Classification report in Decision Tree ............................... 103
3.8 Classification report in Random Forest .............................. 103
3.9 Classification report in support vector machine .................... 104
3.10 Classification report in XGBoost .................................... 104
3.11 Out-of-sample accuracy with respect to training sample sizes .. 107
3.12 Out-of-sample Precision with respect to training sample sizes .. 107
3.13 Out-of-sample F1 with respect to training sample sizes .......... 108
3.14 Models’ performance with different sample sizes used in training .. 110
3.15 Performance of the models for different time steps using F1-score .. 111
3.16 Coefficients of model F1-score in OLS regression ................ 112
### List of tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.17</td>
<td>Classification report</td>
<td>118</td>
</tr>
<tr>
<td>3.18</td>
<td>Classification report 2</td>
<td>119</td>
</tr>
<tr>
<td>4.1</td>
<td>Dataset Description</td>
<td>137</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of performance of model selection</td>
<td>142</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In recent years, new technology seeks to improve and automate financial services, leading to the development of Financial Technology (Fintech). Fintech is applied in a wide range of finance applications including automated portfolio manager, the development and trading of cryptocurrencies, algorithmic trading, and credit risk control. Machine Learning technology, as a kind of data-driven approach, is widely used in the creation of predictive signals that can lead to prediction power in trading systems [6]. Machine learning structure fits the characteristics of non-linearity and non-stationary of return time series, which break the assumption of traditional stochastic econometric models [95]. Machine learning techniques have been successfully applied and gave good predictive models from large data sets in trade execution and the generation of alpha [246, 224].

This thesis points to two major applications in using novel machine learning pipelines. First, we use cryptocurrency as an underlying market to explore possible directions in financial applications of machine learning techniques. In this research topic, a novel survey on cryptocurrency trading is carried out, which includes classical cryptocurrency trading software systems, systematic trading methods, emergent trading techniques, portfolios and market conditions in the cryptocurrency market. To the best of our knowledge, our survey is the first academic effort to systematically analyse cryptocurrency trading techniques and summarise them in groups. During the survey process, we were impressed by how widespread modern machine learning has become in the trading process and in investment decisions. Inspired by Sirignano’s research paper [381], we apply machine learning techniques with complex structures (deep neural networks with auto-encoder) in high-frequency limit order book direction prediction in the real cryptocurrency market. Some practical research questions are presented and discussed in experiments, including multi-label limit order book signs, prediction performance decay in real data, comparison of different retraining algorithms in deep neural networks, etc. Second, we focus on a niche but important direction in machine learning pipelines – model selection based on
1.1 Related Work

A Cryptocurrency is a peer-to-peer digital exchange system in which cryptography is used to generate and distribute currency units [325]. The research of cryptocurrency has been widely followed by academic researchers, especially its unique decentralised characteristic and its expected return by active trading. Research also suggests that cryptocurrency returns are driven and can be predicted by factors that are specific to cryptocurrency markets [287]. As an asset that can be traded on the open exchange, trading strategies of cryptocurrency could be driven by econometrics methods (eg. time-series analysis, pairs trading, etc.), data-mining methods (eg. machine learning), recurrent arbitrage opportunities across exchanges [302]. As a result of the wide variety of trading strategies published by academics and practitioners applied to cryptocurrency trading, it is even more critical to go for an aggregated report of all cryptocurrency trading strategies. The thesis summarising contributes to the area by available cryptocurrency trading strategies and analysing the techniques used in trading strategies. Problems in the area of cryptocurrency trading strategies and related research directions based on practical research and trading experience are pointed out in this thesis.

In the area of high-frequency trading, order book dynamics is a key component in finding the regularities of price formation from the ultra-microscopic view of modern automated transaction markets. From the perspective of the high-frequency markets, all feedback effects in the jump intensity are linear and quadratic in past returns [60], which promotes the use of the point processes in analysing the high-frequency time series. Empirical research shows that the total volume of metaorder (the collection of all individual orders belonging to the same trading decision) affects the price by an approximate square root. The impact of small and large orders on order book liquidity leads to dynamical studies of high-frequency order flow liquidity/imbalance [65, 92, 83] and agent-based latent order book liquidity [396, 312]. Rama Cont proposes the simulation model of the
order book in a liquid market from reduced-form representation and constructs the net order flow process [129, 128, 127, 126]. With the development of machine learning techniques, time series forecasting models with memory tuples have been attempted in high-frequency order sign prediction [382]. Using deep learning models with high-frequency order book dynamics/imbalance features has been validated to have a high degree of accuracy and facilitates the generation of high-frequency trading strategies. Our contribution to this research topics is discussed in Chapter 3.

Machine learning is an efficient tool to analyse potential non-linear relationships among data. Normally, we will organise the datasets columned by features before putting them into machine learning models, especially when we apply machine learning in financial applications which often has a huge number of features in the dataset. Complex machine learning models are made up mostly of black-box models with poor interpretability [73]. But black-box machine learning models have wide applications in finance, including signal extraction for excess alpha, time-series modelling, default risk analysis etc. Selecting the best performing pipeline (a way to codify and automate the workflow it takes to produce a machine learning model) from a portfolio of models consisting of a large number of black-box models is very important. Validate process is designed for selecting optimum models to fit the dataset, including cross validation and batch validation [76]. But common validation methods are computation-expensive, especially when no prior knowledge of suitable models is found or application scenarios have high runtime requirements (eg. high-frequency trading). This is the research topic we study in this thesis. Specifically, we limit our available models to tree-structure models.

1.2 Contributions

We summarise two major applications in using novel machine learning pipelines into three chapters, see snapshot Table 1.1. In this table, we list research questions, hypotheses, theory, data, empirical methods and key findings of our work.
### Table 1.1 A snapshot of this thesis

<table>
<thead>
<tr>
<th>Research questions</th>
<th>Hypotheses</th>
<th>Theory</th>
<th>Data</th>
<th>Empirical methods</th>
<th>Key findings</th>
</tr>
</thead>
</table>
| Classification of existing cryptocurrency trading techniques and theories and their trends | Not applicable           | Not applicable                   | 146 research papers and resources available | Snowballing in paper collection                        | 1) A comprehensive survey of cryptocurrency trading  
2) Dataset, research trends, research directions and opportunities are analysed                                    |
| 1) How well does a universal machine learning model perform?                      | 1) Previous research could be extended to cryptocurrency market  
2) Normal deep learning model has to be optimised work on real data  
3) Binary signs could be extended to multi-label with a refresh of prediction model | Limit Order Book Dynamics  
LSTM  
Autoencoder  
Maximum-Minimum Drawdown | Live data through GDAX | 1) Deep learning optimised by the autoencoder for real data  
2) Dynamic Retraining of deep learning model considering the entry of new information  
3) Test multi-label signs prediction compared to binary signs | 1) Our model achieves 78% F1-score in prediction  
2) Stable frequency retraining and dynamic frequency retraining could improve stability |
| 1) What is the relationship between feature contribution dispersion in model selection? | The degree of utilisation rate of features, reflected in the dispersion of features’ contribution or the characteristics of features may affect the performance of tree models’ selection when we tune the hyperparameters to optimise model parameters | Coefficient of Variation  
Model Selection  
Entropy Theory  
Feature Importance | OpenML Dataset | 1) Compare the Coefficient of Variation and entropy values in selecting tree models  
2) Compare the performance of our model with cross-validation in selecting models | 1) Feature contribution dispersion-based model selection would be impacted by features selection  
2) Our method performs better or similar to the model selection performance while our method has a large efficiency improvement in model selection execution time |
1.2 Contributions

1.2.1 Survey on existing cryptocurrency trading techniques and theories

As can be seen in Chapter 2, our survey makes the following contributions:

Definition. The survey defines cryptocurrency trading and categorises it into: cryptocurrency markets, cryptocurrency trading models, and cryptocurrency trading strategies. The core content of the survey is trading strategies for cryptocurrencies while we cover all aspects of it.

Multidisciplinary Survey. The survey provides a comprehensive overview of 146 cryptocurrency trading papers, across different academic disciplines such as finance and economics, artificial intelligence and computer science. Some papers may cover multiple aspects and will be surveyed for each category.

Analysis. The survey analyses the research distribution, datasets and trends that characterise the cryptocurrency trading literature.

Horizons. The survey identifies challenges, promising research directions in cryptocurrency trading, aimed to promote and facilitate further research.

1.2.2 Order book signs prediction with deep learning techniques

In this research, we propose to adopt a machine learning approach to reveal useful patterns from limit order books. Specifically, we show that there are universal features amongst cryptocurrencies that can improve the predictive power of machine learning models, as there are in the case of equities [382]. We also show that feeding more data to train our deep neural network fails to improve the model performances; simpler single-dimensional models are preferred. Thirdly, we test the model on live data for different periods of varying length, which bears conceptual as well as technical challenges. We show in this paper that, certain known architectures can meet both requirements, when using a novel training method that we call Walkthrough Training. Finally, we explore the problem of multi-label classification, by predicting "small" or “large” increase/decrease of the mid-price; we analyze the trade-off between performance and retrain frequency of Walkthrough Training in this context. Ultimately, our findings pave the way to the design of novel trading strategies and market estimators.

1.2.3 Dispersion of feature importance in selecting tree models

This research aims at revealing the characteristics of utilisation rate of features using feature importance dispersion when we perform hyperparameter tuning in tree models.
1.3 Thesis Outline

This novel conceptual contribution connects common feature weighting techniques with the dispersion degree matrix. The subtle effect of features on the generalisation of the model shows that there is a strong relationship between the distribution of feature importance and the classification prediction performance of the model. This research uses Feature Contribution Dispersion notion to define a new pipeline (two measurements to calculate the dispersion extent) for model selection as opposed to cross validation method. The highlight of Feature Contribution Dispersion method is that the optimised pipeline can generally maintain the performance of cross validation (in terms of test accuracy as generalisation) whilst reducing computation time by at least a third.

1.3 Thesis Outline

The remainder of this thesis is structured as follows. In Chapter 2, we survey existing cryptocurrency trading techniques and theorems in groups and summarise this area by research distributions, datasets statistics, and opportunities. Chapter 3 extends existing research using deep learning in price formation to more stable complex machine learning structures in predicting high-frequency signs of continuous order book in the cryptocurrency market. Chapter 4 describes the research of a brand new evaluation criterion – “dispersion of feature importance” in selecting tree models in machine learning pipelines. Finally, we conclude our findings in Chapter 5 and consider extensions and potential future work.
Chapter 2

A comprehensive survey on cryptocurrency trading

2.1 Introduction

Cryptocurrencies have experienced broad market acceptance and fast development despite their recent conception. Many hedge funds and asset managers have begun to include cryptocurrency-related assets into their portfolios and trading strategies. The academic community has similarly spent considerable efforts in researching cryptocurrency trading. This section seeks to provide a comprehensive survey of the research on cryptocurrency trading, by which we mean any study aimed at facilitating and building strategies to trade cryptocurrencies.

As an emerging market and research direction, cryptocurrencies and cryptocurrency trading have seen considerable progress and a notable upturn in interest and activity [164]. From Figure 2.1, we observe over 85% of papers have appeared since 2018, demonstrating the emergence of cryptocurrency trading as a new research area in financial trading. The sampling interval of this survey is from 2013 to 2022.

The literature is organised according to six distinct aspects of cryptocurrency trading:

- Cryptocurrency trading software systems (i.e., real-time trading systems, turtle trading systems, arbitrage trading systems);
- Systematic trading including technical analysis, pairs trading and other systematic trading methods;
- Emergent trading technologies including econometric methods, machine learning technology and other emergent trading methods;
2.1 Introduction

- Portfolio and cryptocurrency assets including research among cryptocurrency co-movements and crypto-asset portfolio research;
- Market condition research including bubbles [172] or crash analysis and extreme conditions;
- Other Miscellaneous cryptocurrency trading research.

In this survey we aim at compiling the most relevant research in these areas and extract a set of descriptive indicators that can give an idea of the level of maturity research in this area has achieved.

![Fig. 2.1 Cryptocurrency Trading Publications (cumulative) during 2013-2022(December 2022)](image)

We also summarise research distribution (among research properties and categories/research technologies). The distribution among properties defines the classification of research objectives and content. The distribution among technologies defines the classification of methods or technological approaches to the study of cryptocurrency trading. Specifically, we subdivide research distribution among categories/technologies into statistical methods and machine learning technologies. Moreover, We identify datasets and opportunities (potential research directions) that have appeared in the cryptocurrency trading area. To ensure that our survey is self-contained, we aim to provide sufficient material to adequately guide financial trading researchers who are interested in cryptocurrency trading.
2.2 Cryptocurrency Trading

There has been related work that discussed or partially surveyed the literature related to cryptocurrency trading. Kyriazis et al. [271] investigated the efficiency and profitable trading opportunities in the cryptocurrency market. Ahamad et al. [7] and Sharma et al. [373] gave a brief survey on cryptocurrencies, merits of cryptocurrencies compared to fiat currencies and compared different cryptocurrencies that are proposed in the literature. Ujan et al. [325] gave a brief survey of cryptocurrency systems. Ignasi et al. [318] performed a bibliometric analysis of bitcoin literature. The outcomes of this related work focused on specific area in cryptocurrency, including cryptocurrencies and cryptocurrency market introduction, cryptocurrency systems / platforms, bitcoin literature review, etc. To the best of our knowledge, no previous work has provided a comprehensive survey particularly focused on cryptocurrency trading.

In summary, the chapter makes the following contributions:

**Definition.** This section defines cryptocurrency trading and categorises it into: cryptocurrency markets, cryptocurrency trading models, and cryptocurrency trading strategies. The core content of this survey is trading strategies for cryptocurrencies while we cover all aspects of it.

**Multidisciplinary Survey.** The section provides a comprehensive survey of 177 cryptocurrency trading papers, across different academic disciplines such as finance and economics, artificial intelligence and computer science. Some papers may cover multiple aspects and will be surveyed for each category.

**Analysis.** The section analyses the research distribution, datasets and trends that characterise the cryptocurrency trading literature.

**Horizons.** The section identifies challenges, promising research directions in cryptocurrency trading, aimed to promote and facilitate further research.

2.2 Cryptocurrency Trading

This section provides an introduction to cryptocurrency trading. We will discuss **Blockchain**, as the enabling technology, **cryptocurrency markets** and **cryptocurrency trading strategies**.

2.2.1 Blockchain

**Blockchain Technology Introduction**

**Blockchain** is a digital ledger of economic transactions that can be used to record not just financial transactions, but any object with an intrinsic value. [393]. In its simplest form, a
2.2 Cryptocurrency Trading

Blockchain is a series of immutable data records with timestamps, which are managed by a cluster of machines that do not belong to any single entity. Each of these data blocks is protected by cryptographic principle and bound to each other in a chain (cf. Figure 2.2 for the workflow).

Cryptocurrencies like Bitcoin are conducted on a peer-to-peer network structure. Each peer has a complete history of all transactions, thus recording the balance of each account. For example, a transaction is a file that says “A pays X Bitcoins to B” that is signed by A using its private key. This is basic public-key cryptography, but also the building block on which cryptocurrencies are based. After being signed, the transaction is broadcast on the network. When a peer discovers a new transaction, it checks to make sure that the signature is valid (this is equivalent to using the signer’s public key, denoted as the algorithm in Figure 2.2). If the verification is valid then the block is added to the chain; all other blocks added after it will “confirm” that transaction. For example, if a transaction is contained in block 502 and the length of the blockchain is 507 blocks, it means that the transaction has 5 confirmations (507-502) [365].

![Fig. 2.2 Workflow of Blockchain transaction](Image)

**From Blockchain to cryptocurrencies**

Confirmation is a critical concept in cryptocurrencies; only miners can confirm transactions. Miners add blocks to the Blockchain; they retrieve transactions in the previous block and combine it with the hash of the preceding block to obtain its hash, and then store the derived hash into the current block. Miners in Blockchain accept transactions, mark them as legitimate and broadcast them across the network. After the miner confirms the transaction, each node must add it to its database. In layman terms, it has become part of the Blockchain and miners undertake this work to obtain cryptocurrency tokens, such as Bitcoin. In contrast to Blockchain, cryptocurrencies are related to the use of tokens based on distributed ledger technology. Any transaction involving purchase, sale, investment, etc. involves a Blockchain native token or sub-token.

Blockchain is a platform that drives cryptocurrency and is a technology that acts as a distributed ledger for the network. The network creates a means of transaction and enables the transfer of value and information. Cryptocurrencies are the tokens used in these networks to send value and pay for these transactions. They can be thought of as
tools on the Blockchain, and in some cases can also function as resources or utilities. In other instances, they are used to digitise the value of assets.

In summary, cryptocurrencies are part of an ecosystem based on Blockchain technology.

### 2.2.2 Introduction of cryptocurrency market

**What is cryptocurrency?**

*Cryptocurrency* is a decentralised medium of exchange which uses cryptographic functions to conduct financial transactions [145]. Cryptocurrencies leverage the Blockchain technology to gain decentralisation, transparency, and immutability [319]. In the above, we have discussed how Blockchain technology is implemented for cryptocurrencies.

In general, the security of cryptocurrencies is built on cryptography, neither by people nor on trust [329]. For example, Bitcoin uses a method called “Elliptic Curve Cryptography” to ensure that transactions involving Bitcoin are secure [417]. Elliptic curve cryptography is a type of public-key cryptography that relies on mathematics to ensure the security of transactions. When someone attempts to circumvent the aforesaid encryption scheme by brute force, it takes them one-tenth the age of the universe to find a value match when trying 250 billion possibilities every second [193]. Regarding its use as a currency, cryptocurrency has properties similar to fiat currencies. It has a controlled supply. Most cryptocurrencies limit the availability of their currency volumes. E.g. for Bitcoin, the supply will decrease over time and will reach its final quantity sometime around 2,140. All cryptocurrencies control the supply of tokens through a timetable encoded in the Blockchain.

One of the most important features of cryptocurrencies is the exclusion of financial institution intermediaries [205]. The absence of a “middleman” lowers transaction costs for traders. For comparison, if a bank’s database is hacked or damaged, the bank will rely entirely on its backup to recover any information that is lost or compromised. With cryptocurrencies, even if part of the network is compromised, the rest will continue to be able to verify transactions correctly. Cryptocurrencies also have the important feature of not being controlled by any central authority [363]: the decentralised nature of the Blockchain ensures cryptocurrencies are theoretically immune to government control and interference.

The pure digital asset is anything that exists in a digital format and carries with it the right to use it. Currently, digital assets include digital documents, motion picture and so on; the market for digital assets has in fact evolved since its inception in 2009, with the first digital asset “Bitcoin” [231]. For this reason, we call the cryptocurrency the “first pure digital asset”.

As of December 20, 2019, there exist 4,950 cryptocurrencies and 20,325 cryptocurrency markets; the market cap is around 190 billion dollars [121]. Figure 2.3 shows historical data
2.2 Cryptocurrency Trading

on global market capitalisation and 24-hour trading volume [402]. The blue line is the total cryptocurrency market capitalization and green/red histogram is the total cryptocurrency market volume. The total market cap is calculated by aggregating the dollar market cap of all cryptocurrencies. From the figure, we can observe how cryptocurrencies experience exponential growth in 2017 and a large bubble burst in early 2018. In the wake of the pandemic, cryptocurrencies raised dramatically in value in 2020. In 2023, the market value of cryptocurrencies has been very volatile but consistently at historically high levels.

![Fig. 2.3 Total Market Capitalization and Volume of cryptocurrency market, USD [402]](image)

There are three mainstream cryptocurrencies [135]: Bitcoin (BTC), Ethereum (ETH), and Litecoin (LTC). Bitcoin was created in 2009 and garnered massive popularity. On October 31, 2008, an individual or group of individuals operating under the pseudonym Satoshi Nakamoto released the Bitcoin white paper and described it as: ”A pure peer-to-peer version of electronic cash that can be sent online for payment from one party to another without going through a counterparty, ie. a financial institution.” [328] Launched by Vitalik Buterin in 2015, Ethereum is a special Blockchain with a special token called Ether (ETH symbol in exchanges). A very important feature of Ethereum is the ability to create new tokens on the Ethereum Blockchain. The Ethereum network went live on July 30, 2015, and pre-mined 72 million Ethereum. Litecoin is a peer-to-peer cryptocurrency created by Charlie Lee. It was created according to the Bitcoin protocol, but it uses a different hashing algorithm. Litecoin uses a memory-intensive proof-of-work algorithm, Scrypt.

Figure 2.4 shows percentages of total cryptocurrency market capitalisation; Bitcoin and Ethereum account for the majority of the total market capitalisation (data collected on 23 March 2023).
2.2 Cryptocurrency Trading

Cryptocurrency Exchanges

A cryptocurrency exchange or digital currency exchange (DCE) is a business that allows customers to trade cryptocurrencies. Cryptocurrency exchanges can be market makers, usually using the bid-ask spread as a commission for services, or as a matching platform, by simply charging fees. A cryptocurrency exchange or digital currency exchange (DCE) is a place that allows customers to trade cryptocurrencies. Cryptocurrency exchanges can be market makers (usually using the bid-ask spread as a commission for services) or a matching platform (simply charging fees).

Table 2.1 shows the top or classical cryptocurrency exchanges according to the rank list, by volume, compiled on “nomics” website [334]. Chicago Mercantile Exchange (CME), Chicago Board Options Exchange (CBOE) as well as BAKKT (backed by New York Stock Exchange) are regulated cryptocurrency exchanges. Fiat currency data also comes from “nomics” website [334]. Regulatory authority and supported currencies of listed exchanges are collected from official websites or blogs.

Spot and Future in Cryptocurrency

The current state of spot and futures trading in cryptocurrencies is highly active and growing rapidly. Spot trading involves buying and selling cryptocurrencies on an exchange at the current market price, while futures trading allows traders to speculate on the future price of a cryptocurrency by buying and selling contracts that are settled at a future date. In recent years, the cryptocurrency market has seen a surge in spot trading activity, as more individuals and institutional investors have entered the market. Many cryptocurrency
exchanges now offer a wide variety of spot trading pairs, allowing users to trade between various cryptocurrencies and fiat currencies. Futures trading in cryptocurrencies has also grown significantly in recent years, with major exchanges such as CME [361] and Binance offering futures contracts for Bitcoin and other cryptocurrencies [61]. Futures trading can provide traders with a way to manage risk and hedge their positions in the volatile cryptocurrency market. Following the introduction of futures, there is evidence of a significant increase in cross-exchange price synchronisation for BTC. The significant increase in BTC-USD price synchronisation relative to other exchange rate pairs is evidenced by an increase in price correlation and a reduction in arbitrage opportunities. The experiment also shows support for increased price efficiency, market quality and liquidity. [25] There is evidence that the introduction of bitcoin futures has improved the information efficiency of the bitcoin spot market, but the second largest cryptocurrency, Ethereum, has seen no signs of improved information efficiency following the introduction of bitcoin futures. [377] As well as giving institutional traders access to regulated cryptocurrency products, futures offer a way to improve market efficiency by shorting bitcoin. And leveraged currency traders tend to hold the largest positions, net shorts, and their trading behaviour plays a key role in the bitcoin futures market. [39]

Perpetual futures, as swaps that never expire, are by far the most popular derivatives traded in the cryptocurrency market, with over 100 billion dollars in daily trading volume. Perpetual contracts provide investors with leveraged exposure to cryptocurrencies without the need to roll over or hold them outright. Deviations of crypto perpetual futures from unarbitrageable prices are much larger than those recorded in traditional cryptocurrency markets. These deviations span cryptocurrencies and decrease over time as crypto markets develop and become more efficient. [206] Although there are no opening and closing times, perpetual futures exhibit multiple ‘U-shaped’ curves, seasonality effects and opening effects. Researchers have found suggestive evidence of spillover effects between perpetual and quarterly futures contracts. [139]

Table 2.1 Cryptocurrency exchanges Lists

<table>
<thead>
<tr>
<th>Exchanges</th>
<th>Category</th>
<th>Supported currencies</th>
<th>Fiat Currency</th>
<th>Registration country</th>
<th>Regulatory authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>CME</td>
<td>Derivatives</td>
<td>BTC and Ethereum [114]</td>
<td>USD</td>
<td>USA [116]</td>
<td>CFTC [115]</td>
</tr>
<tr>
<td>CBOE</td>
<td>Derivatives</td>
<td>BTC [97]</td>
<td>USD</td>
<td>USA [96]</td>
<td>CFTC [98]</td>
</tr>
<tr>
<td>BAKKT (NYSE)</td>
<td>Derivatives</td>
<td>BTC [31]</td>
<td>USD</td>
<td>USA [32]</td>
<td>CFTC [31]</td>
</tr>
<tr>
<td>BitMEX</td>
<td>Derivatives/Spot</td>
<td>12 cryptocurrencies [54]</td>
<td>USD</td>
<td>Seychelles [55]</td>
<td>CFTC</td>
</tr>
<tr>
<td>Bitstamp</td>
<td>Spot/Derivatives</td>
<td>65+ cryptocurrencies [56]</td>
<td>EUR, USD</td>
<td>Luxembourg [57]</td>
<td>CSSF [58]</td>
</tr>
<tr>
<td>Poloniex</td>
<td>Spot/Derivatives</td>
<td>350+ cryptocurrencies [351]</td>
<td>USD</td>
<td>USA [151]</td>
<td></td>
</tr>
</tbody>
</table>
2.2 Cryptocurrency Trading

2.2.3 Cryptocurrency Trading

Definition

First we give a definition of cryptocurrency trading.

**Definition 1.** Cryptocurrency trading is the act of buying and selling of cryptocurrencies with the intention of making a profit.

The definition of cryptocurrency trading can be broken down into three aspects: object, operation mode and trading strategy. The object of cryptocurrency trading is the asset being traded, which is “cryptocurrency”. The operation mode of cryptocurrency trading depends on the means of transaction in the cryptocurrency market, which can be classified into “trading of cryptocurrency Contract for Differences (CFD)” (The contract between the two parties, often referred to as the “buyer” and “seller”, stipulates that the buyer will pay the seller the difference between themselves when the position closes [26]) and “buying and selling cryptocurrencies via an exchange”. A trading strategy in cryptocurrency trading, formulated by an investor, is an algorithm that defines a set of predefined rules to buy and sell on cryptocurrency markets.

Advantages of Trading Cryptocurrency

The benefits of cryptocurrency trading include:

**Drastic fluctuations.** The volatility of cryptocurrencies are often likely to attract speculative interest and investors. The rapid fluctuations of intraday prices can provide traders with great money-earning opportunities, but it also includes more risk.

**24-hour market.** The cryptocurrency market is available 24 hours a day, 7 days a week because it is a decentralised market. Unlike buying and selling stocks and commodities, the cryptocurrency market is not traded physically from a single location. Cryptocurrency transactions can take place between individuals, in different venues across the world.

**Near Anonymity.** Buying goods and services using cryptocurrencies is done online and does not require to make one’s own identity public. With increasing concerns over identity theft and privacy, cryptocurrencies can thus provide users with some advantages regarding privacy. Different exchanges have specific Know-Your-Customer (KYC) measures for identifying users or customers [5]. The KYC undertook in the exchanges allows financial institutions to reduce the financial risk while maximising the wallet owner’s anonymity.
Peer-to-peer transactions. One of the biggest benefits of cryptocurrencies is that they do not involve financial institution intermediaries. As mentioned above, this can reduce transaction costs. Moreover, this feature might appeal to users who distrust traditional systems. Over-the-counter (OTC) cryptocurrency markets offer, in this context, peer-to-peer transactions on the Blockchain. The most famous cryptocurrency OTC market is “LocalBitcoin [293]”.

Programmable “smart” capabilities. Some cryptocurrencies can bring other benefits to holders, including limited ownership and voting rights. Cryptocurrencies may also include a partial ownership interest in physical assets such as artwork or real estate.

Disadvantages of Trading Cryptocurrency

The disadvantages of cryptocurrency trading include:

Scalability Problem. Before the massive expansion of the technology infrastructure, the number of transactions and the speed of transactions cannot compete with traditional currency trading. Scalability issues led to a multi-day trading backlog in March 2020, affecting traders looking to move cryptocurrencies from their personal wallets to exchanges [175].

Cybersecurity Issues. As a digital technology, cryptocurrencies are subject to cyber security breaches and can fall into the hands of hackers. Recently, over $600 million of ethereum and other cryptocurrencies were stolen in August 2021 in blockchain-based platform Poly Network [177]. Mitigating this situation requires ongoing maintenance of the security infrastructure and the use of enhanced cyber security measures that go beyond those used in traditional banking [260].

Regulations. Authorities around the world face challenging questions about the nature and regulation of cryptocurrency as some parts of the system and its associated risks are largely unknown. There are currently three types of regulatory systems used to control digital currencies, they include: closed system for the Chinese market, open and liberal for the Swiss market, and open and strict system for the US market [407]. At the same time, we notice that some countries such as India is not at par in using the cryptocurrency. As Buffett said, “It doesn’t make sense. This thing is not regulated. It’s not under control. It’s not under the supervision of […] United States Federal Reserve or any other central bank [174]."
2.3 Cryptocurrency Trading Strategy

Cryptocurrency trading strategy is the main focus of this survey. There are many trading strategies, which can be broadly divided into two main categories: technical and fundamental. Technical and fundamental trading are two main trading analysis thoughts when it comes to analyzing the financial markets. Most traders use these two analysis methods or both [337]. From a survey on stock prediction, we in fact know that 66% of the relevant research work was based on technical analysis; while 23% and 11% were based on fundamental and technical analysis, respectively [336]. Cryptocurrency trading can draw on the experience of stock market trading in most scenarios. So we divide trading strategies into two main categories: technical and fundamental trading.

They are similar in the sense that they both rely on quantifiable information that can be backtested against historical data to verify their performance. In recent years, a third kind of trading strategy, which we call programmatic trading, has received increasing attention. Such a trading strategy is similar to a technical trading strategy because it uses trading activity information on the exchange to make buying or selling decisions. Programmatic traders build trading strategies with quantitative data, which is mainly derived from price, volume, technical indicators or ratios to take advantage of inefficiencies in the market and are executed automatically by trading software. Cryptocurrency market is different from traditional markets as there are more arbitrage opportunities, higher fluctuation and transparency. Due to these characteristics, most traders and analysts prefer using programmatic trading in cryptocurrency markets.

2.3.1 Cryptocurrency Trading Software System

Software trading systems allow international transactions, process customer accounts and information, and accept and execute transaction orders [86]. A cryptocurrency trading system is a set of principles and procedures that are pre-programmed to allow trade between cryptocurrencies and between fiat currencies and cryptocurrencies. Cryptocurrency trading systems are built to overcome price manipulation, cybercriminal activities and transaction delays [40]. When developing a cryptocurrency trading system, we must consider the capital market, base asset, investment plan and strategies [322]. Strategies are the most important part of an effective cryptocurrency trading system and they will be introduced below. There exist several cryptocurrency trading systems that are available commercially, for example, Capfolio, 3Commas, CCXT, Freqtrade and Ctubio. From these cryptocurrency trading systems, investors can obtain professional trading strategy support, fairness and transparency from the professional third-party consulting companies and fast customer services.
2.3 Cryptocurrency Trading Strategy

2.3.2 Systematic Trading

**Systematic Trading** is a way to define trading goals, risk controls and rules. In general, systematic trading includes high frequency trading and slower investment types like systematic trend tracking. In this survey, we divide systematic cryptocurrency trading into technical analysis, pairs trading and others. Technical analysis in cryptocurrency trading is the act of using historical patterns of transaction data to assist a trader in assessing current and projecting future market conditions for the purpose of making profitable trades. Price and volume charts summarise all trading activity made by market participants in an exchange and affect their decisions. Some experiments showed that the use of specific technical trading rules allows generating excess returns, which is useful to cryptocurrency traders and investors in making optimal trading and investment decisions [189]. Pairs trading is a systematic trading strategy that considers two similar assets with slightly different spreads. If the spread widens, short the high cryptocurrencies and buy the low cryptocurrencies. When the spread narrows again to a certain equilibrium value, a profit is generated [152]. Papers shown in this section involve the analysis and comparison of technical indicators, pairs and informed trading, amongst other strategies.

2.3.3 Tools for building automated trading systems

Tools for building automated trading systems in cryptocurrency market are those emergent trading strategies for cryptocurrency. These include strategies that are based on econometrics and machine learning technologies.

**Econometrics on Cryptocurrency**

Econometric methods apply a combination of statistical and economic theories to estimate economic variables and predict their values [413]. **Statistical models** use mathematical equations to encode information extracted from the data [245]. In some cases, statistical modeling techniques can quickly provide sufficiently accurate models [44]. Other methods might be used, such as sentiment-based prediction and long-and-short-term volatility classification based prediction [103]. The prediction of volatility can be used to judge the price fluctuation of cryptocurrencies, which is also valuable for the pricing of cryptocurrency-related derivatives [240].

When studying cryptocurrency trading using econometrics, researchers apply statistical models on time-series data like generalised autoregressive conditional heteroskedasticity (GARCH) and BEKK (named after Baba, Engle, Kraft and Kroner, 1995 [154]) models to evaluate the fluctuation of cryptocurrencies [91]. A **linear statistical model** is a method to evaluate the linear relationship between prices and an explanatory variable [331]. When
there exists more than one explanatory variable, we can model the linear relationship between explanatory (independent) and response (dependent) variables with multiple linear models. The common linear statistical model used in the time-series analysis is the autoregressive moving average (ARMA) model [111].

**Machine Learning Technology**

Machine learning is an efficient tool for developing Bitcoin and other cryptocurrency trading strategies [316] because it can infer data relationships that are often not directly observable by humans. From the most basic perspective, Machine Learning relies on the definition of two main components: input features and objective function. The definition of Input Features (data sources) is where knowledge of fundamental and technical analysis comes into play. We may divide the input into several groups of features, for example, those based on Economic indicators (such as, gross domestic product indicator, interest rates, etc.), Social indicators (Google Trends, Twitter, etc.), Technical indicators (price, volume, etc.) and other Seasonal indicators (time of day, day of the week, etc.). The objective function defines the fitness criteria one uses to judge if the Machine Learning model has learnt the task at hand. Typical predictive models try to anticipate numeric (e.g., price) or categorical (e.g., trend) unseen outcomes. The machine learning model is trained by using historic input data (sometimes called in-sample) to generalise patterns therein to unseen (out-of-sample) data to (approximately) achieve the goal defined by the objective function. Clearly, in the case of trading, the goal is to infer trading signals from market indicators which help to anticipate asset future returns.

Generalisation error is a pervasive concern in the application of Machine Learning to real applications, and of utmost importance in Financial applications. We need to use statistical approaches, such as cross validation, to validate the model before we actually use it to make predictions. In machine learning, this is typically called “validation”. The process of using machine learning technology to predict cryptocurrency is shown in Figure 2.5.

![Fig. 2.5 Process of machine learning in predicting cryptocurrency](image-url)
2.3 Cryptocurrency Trading Strategy

Depending on the formulation of the main learning loop, we can classify Machine Learning approaches into three categories: Supervised learning, Unsupervised learning and Reinforcement learning. We list a general comparison [221] among these three machine learning methods in Table 2.2. **Supervised learning** is used to derive a predictive function from labeled training data. Labeled training data means that each training instance includes inputs and expected outputs. Usually, these expected outputs are produced by a supervisor and represent the expected behaviour of the model. The most used labels in trading are derived from in sample future returns of assets. **Unsupervised learning** tries to infer structure from unlabeled training data and it can be used during exploratory data analysis to discover hidden patterns or to group data according to any pre-defined similarity metrics. **Reinforcement learning** utilises software agents trained to maximise a utility function, which defines their objective; this is flexible enough to allow agents to exchange short term returns for future ones. In the financial sector, some trading challenges can be expressed as a game in which an agent aims at maximising the return at the end of the period.

Table 2.2 Comparison among Different machine learning methods

<table>
<thead>
<tr>
<th></th>
<th>Supervised Learning</th>
<th>Unsupervised Learning</th>
<th>Reinforcement Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition</strong></td>
<td>The machine learns by using labeled data</td>
<td>Machine training through unlabelled data without any guidance</td>
<td>Agents interact with their environment by performing actions and learning from errors or rewards</td>
</tr>
<tr>
<td><strong>Type of problems</strong></td>
<td>Regression or classification</td>
<td>Association or clustering</td>
<td>Reward-based</td>
</tr>
<tr>
<td><strong>Type of data</strong></td>
<td>Labeled data</td>
<td>Unlabeled data</td>
<td>No predefined data</td>
</tr>
<tr>
<td><strong>Training</strong></td>
<td>External supervision</td>
<td>No supervision</td>
<td>No supervision</td>
</tr>
<tr>
<td><strong>Approach</strong></td>
<td>Mapping tagged inputs to unknown outputs</td>
<td>Understanding patterns or finding outputs</td>
<td>Follow the trail-and-error method</td>
</tr>
</tbody>
</table>

The use of machine learning in cryptocurrency trading research encompasses the connection between data sources’ understanding and machine learning model research. Further concrete examples are shown in a later section.

2.3.4 Portfolio Research

**Portfolio theory** advocates diversification of investments to maximize returns for a given level of risk by allocating assets strategically. The celebrated mean-variance optimisation is a prominent example of this approach [310]. Generally, **crypto asset** denotes a digital asset (i.e., cryptocurrencies and derivatives). There are some common ways to build a diversified portfolio in crypto assets. The first method is to diversify across markets, which is to mix a wide variety of investments within a portfolio of the cryptocurrency market. The second method is to consider the industry sector, which is to avoid investing too much money in any one category. Diversified investment of portfolio in the cryptocurrency
market includes portfolio across cryptocurrencies [285] and portfolio across the global market including stocks and futures [232].

2.3.5 Market Condition Research

Market condition research appears especially important for cryptocurrencies. A financial bubble is a significant increase in the price of an asset without changes in its intrinsic value [81, 262]. Many experts pinpoint a cryptocurrency bubble in 2017 when the prices of cryptocurrencies grew by 900%. In 2018, Bitcoin faced a collapse in its value. This significant fluctuation inspired researchers to study bubbles and extreme conditions in cryptocurrency trading. The cryptocurrency market has experienced a near continuous bull market since the fall of 2020, with the value of Bitcoin soaring from $10,645 on October 7, 2020 to an all-time high of $63,346 on April 15, 2021. This represents a gain of approximately +600% in just six months [176]. Some experts believe that the extreme volatility of exchange rates means that cryptocurrency exposure should be kept at a low percentage of your portfolio. “I understand if you want to buy it because you believe the price will rise, but make sure it’s only a small part of your portfolio, maybe 1% or 2%!” says Thanos Papasavvas, founder of research group ABP Invest, who has a 20-year background in asset management [183]. In any case, bubbles and crash analysis is an important researching area in cryptocurrency trading.

2.4 Paper Collection and Review Schema

The section introduces the scope and approach of our paper collection, a basic analysis, and the structure of our survey.

2.4.1 Survey Scope

We adopt a bottom-up approach to the research in cryptocurrency trading, starting from the systems up to risk management techniques. For the underlying trading system, the focus is on the optimisation of trading platforms structure and improvements of computer science technologies.

At a higher level, researchers focus on the design of models to predict return or volatility in cryptocurrency markets. These techniques become useful to the generation of trading signals. on the next level above predictive models, researchers discuss technical trading methods to trade in real cryptocurrency markets. Bubbles and extreme conditions are hot topics in cryptocurrency trading because, as discussed above, these markets have shown to be highly volatile (whilst volatility went down after crashes). Portfolio and cryptocurrency
2.4 Paper Collection and Review Schema

asset management are effective methods to control risk. We group these two areas in risk management research. Other papers included in this survey include topics like pricing rules, dynamic market analysis, regulatory implications, and so on. Table 2.3 shows the general scope of cryptocurrency trading included in this survey.

Since many trading strategies and methods in cryptocurrency trading are closely related to stock trading, some researchers migrate or use the research results for the latter to the former. When conducting this research, we only consider those papers whose research focuses on cryptocurrency markets or a comparison of trading in those and other financial markets.

Table 2.3 Survey scope table

<table>
<thead>
<tr>
<th>Trading (bottom up)</th>
<th>Trading System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prediction (return)</td>
</tr>
<tr>
<td></td>
<td>Prediction (volatility)</td>
</tr>
<tr>
<td>Risk management</td>
<td>Technical trading methods</td>
</tr>
<tr>
<td></td>
<td>Bubble and extreme condition</td>
</tr>
<tr>
<td>Others</td>
<td>Portfolio and Cryptocurrency asset</td>
</tr>
</tbody>
</table>

Specifically, we apply the following criteria when collecting papers related to cryptocurrency trading:

1. The paper introduces or discusses the general idea of cryptocurrency trading or one of the related aspects of cryptocurrency trading.

2. The paper proposes an approach, study or framework that targets optimised efficiency or accuracy of cryptocurrency trading.

3. The paper compares different approaches or perspectives in trading cryptocurrency.

By “cryptocurrency trading” here, we mean one of the terms listed in Table 2.3 and discussed above.

Some researchers gave a brief survey of cryptocurrency [7, 373], cryptocurrency systems [325] and cryptocurrency trading opportunities [271]. These surveys are rather limited in scope as compared to ours, which also includes a discussion on the latest papers in the area; we want to remark that this is a fast-moving research field.

2.4.2 Paper Collection Methodology

To collect the papers in different areas or platforms, we used keyword searches on Google Scholar and arXiv, two of the most popular scientific databases. We also choose other public repositories like SSRN but we find that almost all academic papers in these platforms
can also be retrieved via Google Scholar; consequently, in our statistical analysis, we count those as Google Scholar hits. We choose arXiv as another source since it allows this survey to be contemporary with all the most recent findings in the area. The interested reader is warned that these papers have not undergone formal peer review. The keywords used for searching and collecting are listed below. [Crypto] means the cryptocurrency market, which is our research interest because methods might be different among different markets. We conducted 6 searches across the two repositories until December 30, 2022.

- [Crypto] + Trading
- [Crypto] + Trading system
- [Crypto] + Prediction
- [Crypto] + Trading strategy
- [Crypto] + Risk Management
- [Crypto] + Portfolio

To ensure high coverage, we adopted the so-called snowballing [421] method on each paper found through these keywords. We checked papers added from snowballing methods that satisfy the criteria introduced above until we reached closure.

### 2.4.3 Collection Results

Table 2.4 shows the details of the results from our paper collection. Keyword searches and snowballing resulted in 177 papers across the six research areas of interest in Section 2.4.1.

Table 2.4 Paper query results. #Hits, #Title, and #Body denote the number of papers returned by the search, left after title filtering, and left after body filtering, respectively.

<table>
<thead>
<tr>
<th>Key Words</th>
<th>#Hits</th>
<th>#Title</th>
<th>#Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Crypto] + Trading</td>
<td>612</td>
<td>60</td>
<td>72</td>
</tr>
<tr>
<td>[Crypto] + Trading System</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>[Crypto] + Prediction</td>
<td>40</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>[Crypto] + Trading Strategy</td>
<td>23</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>[Crypto] + Risk Management / [Crypto] + Portfolio</td>
<td>128</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>Query</td>
<td>-</td>
<td>-</td>
<td>117</td>
</tr>
<tr>
<td>Snowball</td>
<td>-</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td>Overall</td>
<td>-</td>
<td>-</td>
<td>177</td>
</tr>
</tbody>
</table>
Figure 2.6 shows the distribution of papers published at different research sites. Among all the papers, 48.63% papers are published in Finance and Economics venues such as Journal of Financial Economics (JFE), Cambridge Centre for Alternative Finance (CCAF), Finance Research Letters, Centre for Economic Policy Research (CEPR), Finance Research Letters (FRL), Journal of Risk and Financial Management (JRFM) and some other high impact financial journals; 4.79% papers are published in Science venues such as Public Library Of Science one (PLOS one), Royal Society open science and SAGE; 14.38% papers are published in Intelligent Engineering and Data Mining venues such as Symposium Series on Computational Intelligence (SSCI), Intelligent Systems Conference (IntelliSys), Intelligent Data Engineering and Automated Learning (IDEAL) and International Conference on Data Mining (ICDM); 4.79% papers are published in Physics / Physicians venues (mostly in Physics venue) such as Physica A and Maths venue like Journal of Mathematics; 10.96% papers are published in AI and complex system venues such as Complexity and International Federation for Information Processing (IFIP); 15.07% papers are published in Others venues which contains independently published papers and dissertations; 1.37% papers are published on arXiv. The distribution of different venues shows that cryptocurrency trading is mostly published in Finance and Economics venues, but with a wide diversity otherwise.

Fig. 2.6 Publication Venue Distribution
2.4 Paper Collection and Review Schema

2.4.4 Survey Organisation

We discuss the contributions of the collected papers and a statistical analysis of these papers in the remainder of the paper, according to Table 2.5.

Table 2.5 Review Schema

<table>
<thead>
<tr>
<th>Classification</th>
<th>Sec</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cryptocurrency Trading Software System</td>
<td>2.5.1</td>
<td>Trading Infrastructure System</td>
</tr>
<tr>
<td></td>
<td>2.5.2</td>
<td>Real-time Cryptocurrency Trading System</td>
</tr>
<tr>
<td></td>
<td>2.5.3</td>
<td>Turtle trading system in Cryptocurrency market</td>
</tr>
<tr>
<td></td>
<td>2.5.4</td>
<td>Arbitrage Trading Systems for Cryptocurrencies</td>
</tr>
<tr>
<td></td>
<td>2.5.5</td>
<td>Comparison of three cryptocurrency trading systems</td>
</tr>
<tr>
<td>Systematic Trading</td>
<td>2.6.1</td>
<td>Technical Analysis</td>
</tr>
<tr>
<td></td>
<td>2.6.2</td>
<td>Pairs Trading</td>
</tr>
<tr>
<td></td>
<td>2.6.3</td>
<td>Applicability of using equity trading strategies in cryptocurrency</td>
</tr>
<tr>
<td></td>
<td>2.6.5</td>
<td>Others</td>
</tr>
<tr>
<td>Emergent Trading Technologies</td>
<td>2.7.1</td>
<td>Econometrics on cryptocurrency</td>
</tr>
<tr>
<td></td>
<td>2.7.2</td>
<td>Machine learning technology</td>
</tr>
<tr>
<td></td>
<td>2.7.3</td>
<td>Others</td>
</tr>
<tr>
<td>Portfolio, Cryptocurrency Assets and Market condition</td>
<td>2.8.1</td>
<td>Research among cryptocurrency pairs and related factors</td>
</tr>
<tr>
<td></td>
<td>2.8.2</td>
<td>Crypto-asset portfolio research</td>
</tr>
<tr>
<td></td>
<td>2.8.3</td>
<td>Bubbles and crash analysis</td>
</tr>
<tr>
<td></td>
<td>2.8.4</td>
<td>Extreme condition</td>
</tr>
<tr>
<td>Cryptocurrency asset pricing</td>
<td>2.9.1</td>
<td>Factor pricing model</td>
</tr>
<tr>
<td></td>
<td>2.9.2</td>
<td>Valuation in pricing and market network</td>
</tr>
<tr>
<td></td>
<td>2.9.3</td>
<td>Risk Factors</td>
</tr>
<tr>
<td></td>
<td>2.9.4</td>
<td>Advantages and disadvantages among pricing factors</td>
</tr>
<tr>
<td>Others</td>
<td>2.10</td>
<td>Others related to Cryptocurrency Trading</td>
</tr>
<tr>
<td>Summary Analysis of Literature Review</td>
<td>2.11.1</td>
<td>Timeline</td>
</tr>
<tr>
<td></td>
<td>2.11.2</td>
<td>Research distribution among properties</td>
</tr>
<tr>
<td></td>
<td>2.11.3</td>
<td>Research distribution among categories and technologies</td>
</tr>
<tr>
<td></td>
<td>2.11.4</td>
<td>Datasets used in cryptocurrency trading</td>
</tr>
</tbody>
</table>

The papers in our collection are organised and presented from six angles. We introduce the work about several different cryptocurrency trading software systems in Section 2.5. Section 2.6 introduces systematic trading applied to cryptocurrency trading. In Section 2.7, we introduce some emergent trading technologies including econometrics on cryptocurrencies, machine learning technologies and other emergent trading technologies in the cryptocurrency market. Section 2.8 introduces research on cryptocurrency pairs and related factors and crypto-asset portfolios research. In Section 2.8.3 and Section 2.8.4 we discuss cryptocurrency market condition research, including bubbles, crash analysis, and extreme conditions. Section 2.10 introduces other research included in cryptocurrency trading not covered above.

We would like to emphasize that the six headings above focus on a particular aspect of cryptocurrency trading; we give a complete organisation of the papers collected under each heading. This implies that those papers covering more than one aspect will be discussed in different sections, once from each angle.

We analyse and compare the number of research papers on different cryptocurrency trading properties and technologies in Section 2.11, where we also summarise the datasets and the timeline of research in cryptocurrency trading.
We build upon this review to conclude in Section 2.12 with some opportunities for future research.

### 2.5 Cryptocurrency Trading Software Systems

#### 2.5.1 Trading Infrastructure Systems

Following the development of computer science and cryptocurrency trading, many cryptocurrency trading systems/bots have been developed. Table 2.6 compares the cryptocurrency trading systems existing in the market. The table is sorted based on URL types (GitHub or Official website) and GitHub stars (if appropriate).

**Capfolio** is a proprietary payable cryptocurrency trading system which is a professional analysis platform and has an advanced backtesting engine [87]. It supports five different cryptocurrency exchanges.

**3 Commas** is a proprietary payable cryptocurrency trading system platform that can take profit and stop-loss orders at the same time [1]. Twelve different cryptocurrency exchanges are compatible with this system.

**CCXT** is a cryptocurrency trading system with a unified API out of the box and optional normalized data and supports many Bitcoin / Ether / Altcoin exchange markets and merchant APIs. Any trader or developer can create a trading strategy based on this data and access public transactions through the APIs [99]. The CCXT library is used to connect and trade with cryptocurrency exchanges and payment processing services worldwide. It provides quick access to market data for storage, analysis, visualisation, indicator development, algorithmic trading, strategy backtesting, automated code generation and related software engineering. It is designed for coders, skilled traders, data scientists and financial analysts to build trading algorithms. Current CCXT features include:

- Support for many cryptocurrency exchanges;
- Fully implemented public and private APIs;
- Optional normalized data for cross-exchange analysis and arbitrage;
- Out-of-the-box unified API, very easy to integrate.

**Blackbird** Bitcoin Arbitrage is a C++ trading system that automatically executes long / short arbitrage between Bitcoin exchanges. It can generate market-neutral strategies that do not transfer funds between exchanges [59]. The motivation behind Blackbird is to naturally profit from these temporary price differences between different exchanges while being market neutral. Unlike other Bitcoin arbitrage systems, Blackbird does not sell...
2.5 Cryptocurrency Trading Software Systems

but actually short sells Bitcoin on the short exchange. This feature offers two important advantages. Firstly, the strategy is always market agnostic: fluctuations (rising or falling) in the Bitcoin market will not affect the strategy returns. This eliminates the huge risks of this strategy. Secondly, this strategy does not require transferring funds (USD or BTC) between Bitcoin exchanges. Buy and sell transactions are conducted in parallel on two different exchanges. There is no need to deal with transmission delays.

**StockSharp** is an open-source trading platform for trading at any market of the world including 48 cryptocurrency exchanges [385]. It has a free C# library and free trading charting application. Manual or automatic trading (algorithmic trading robot, regular or HFT) can be run on this platform. StockSharp consists of five components that offer different features:

- S#.Designer - Free universal algorithm strategy app, easy to create strategies;
- S#.Data - free software that can automatically load and store market data;
- S#.Terminal - free trading chart application (trading terminal);
- S#.Shell - ready-made graphics framework that can be changed according to needs and has a fully open source in C#;
- S#.API - a free C# library for programmers using Visual Studio. Any trading strategies can be created in S#.API.

**Freqtrade** is a free and open-source cryptocurrency trading robot system written in Python. It is designed to support all major exchanges and is controlled by telegram. It contains backtesting, mapping and money management tools, and strategy optimization through machine learning [178]. Freqtrade has the following features:

- Persistence: Persistence is achieved through SQLite technology;
- Strategy optimization through machine learning: Use machine learning to optimize your trading strategy parameters with real trading data;
- Marginal Position Size: Calculates winning rate, risk-return ratio, optimal stop loss and adjusts position size, and then trades positions for each specific market;
- Telegram management: use telegram to manage the robot.
- Dry run: Run the robot without spending money;

**CryptoSignal** is a professional technical analysis cryptocurrency trading system [137]. Investors can track over 500 coins of Bittrex, Bitfinex, GDAX, Gemini and more. Automated technical analysis includes momentum, RSI, Ichimoku Cloud, MACD, etc. The
system gives alerts including Email, Slack, Telegram, etc. CryptoSignal has two primary features. First of all, it offers modular code for easy implementation of trading strategies; Secondly, it is easy to install with Docker.

**Ctubio** is a C++ based low latency (high frequency) cryptocurrency trading system [138]. This trading system can place or cancel orders through supported cryptocurrency exchanges in less than a few milliseconds. Moreover, it provides a charting system that can visualise the trading account status including trades completed, target position for fiat currency, etc.

**Catalyst** is an analysis and visualization of the cryptocurrency trading system [94]. It makes trading strategies easy to express and backtest them on historical data (daily and minute resolution), providing analysis and insights into the performance of specific strategies. Catalyst allows users to share and organise data and build profitable, data-driven investment strategies. Catalyst not only supports the trading execution but also offers historical price data of all crypto assets (from minute to daily resolution). Catalyst also has backtesting and real-time trading capabilities, which enables users to seamlessly transit between the two different trading modes. Lastly, Catalyst integrates statistics and machine learning libraries (such as matplotlib, scipy, statsmodels and sklearn) to support the development, analysis and visualization of the latest trading systems.

**Golang Crypto Trading Bot** is a Go based cryptocurrency trading system [191]. Users can test the strategy in sandbox environment simulation. If simulation mode is enabled, a fake balance for each coin must be specified for each exchange.

<table>
<thead>
<tr>
<th>Name</th>
<th>Features</th>
<th>#Exchange</th>
<th>Language</th>
<th>Open-Source</th>
<th>URL</th>
<th>#Popularity</th>
</tr>
</thead>
</table>
| Capfolio      | Professional analysis platform, Advanced backtesting engine                | 5         | Not mentioned | No          | Official website [87] | Advanced backtesting engine 3 Commas Simultaneous take profit and stop loss orders | 12         | Not mentioned | No          | Official website [1] | CCXT An out of the box unified API, optional normalized data | 10 | JavaScript / Python / PHP | Yes | GitHub [99] | 13k BlackBird Strategy is market-neutral strategy, not transfer funds between exchanges | 8 | C++ | Yes | GitHub [59] | 4.7k StockSharp Free C# library, free trading charting application | 48 | C# | Yes | GitHub [385] | 2.6k Freqtrade Strategy Optimization by machine learning, Calculate edge position sizing | 2 | Python | Yes | GitHub [178] | 2.4k CryptoSignal Technical analysis trading system | 4 | Python | Yes | GitHub [137] | 1.9k Cubit Low latency | 1 | C++ | Yes | GitHub [138] | 1.7k Catalyst Analysis and visualization of system seamless transition between live and back-testing | 4 | Python | Yes | GitHub [94] | 1.7k Golang Sandbox environment simulation | 7 | Go | Yes | GitHub [191] | 277
2.5 Cryptocurrency Trading Software Systems

2.5.2 Real-time Cryptocurrency Trading Systems

Amit et al. [40] developed a real-time Cryptocurrency Trading System. A real-time cryptocurrency trading system is composed of clients, servers and databases. Traders use a web-application to login to the server to buy/sell crypto assets. The server collects cryptocurrency market data by creating a script that uses the Coinmarket API. Finally, the database collects balances, trades and order book information from the server. The authors tested the system with an experiment that demonstrates user-friendly and secure experiences for traders in the cryptocurrency exchange platform.

2.5.3 Turtle trading system in Cryptocurrency market

The original Turtle Trading system is a trend following trading system developed in the 1970s. The idea is to generate buy and sell signals on stock for short-term and long-term breakouts and its cut-loss condition which is measured by Average true range (ATR) [237]. The trading system will adjust the size of assets based on their volatility. Essentially, if a turtle accumulates a position in a highly volatile market, it will be offset by a low volatility position. Extended Turtle Trading system is improved with smaller time interval spans and introduces a new rule by using exponential moving average (EMA). Three EMA values are used to trigger the “buy” signal: 30EMA (Fast), 60EMA (Slow), 100EMA (Long). The author of [237] performed backtesting and comparing both trading systems (Original Turtle and Extended Turtle) on 8 prominent cryptocurrencies. Through the experiment, Original Turtle Trading System achieved an 18.59% average net profit margin (percentage of net profit over total revenue) and 35.94% average profitability (percentage of winning trades over total numbers of trades) in 87 trades through nearly one year. Extended Turtle Trading System achieved 114.41% average net profit margin and 52.75% average profitability in 41 trades through the same time interval. This research showed how Extended Turtle Trading System compared can improve over Original Turtle Trading System in trading cryptocurrencies.

2.5.4 Arbitrage Trading Systems for Cryptocurrencies

Christian [343] introduced arbitrage trading systems for cryptocurrencies. Arbitrage trading aims to spot the differences in price that can occur when there are discrepancies in the levels of supply and demand across multiple exchanges. As a result, a trader could realise a quick and low-risk profit by buying from one exchange and selling at a higher price on a different exchange. Arbitrage trading signals are caught by automated trading software. The technical differences between data sources impose a server process to be organised for each data source. Relational databases and SQL are reliable solution due
2.6 Systematic Trading

to the large amounts of relational data. The author used the system to catch arbitrage opportunities on 25 May 2018 among 787 cryptocurrencies on 7 different exchanges. The research paper [343] listed the best ten trading signals made by this system from 186 available found signals. The results showed that the system caught the trading signal of “BTG-BTC” to get a profit of up to 495.44% when arbitraging to buy in Cryptopia exchange and sell in Binance exchange. Another three well-traded arbitrage signals (profit expectation around 20% mentioned by the author) were found on 25 May 2018. Arbitrage Trading Software System introduced in that paper presented general principles and implementation of arbitrage trading system in the cryptocurrency market.

2.5.5 Characteristics of three cryptocurrency trading systems

Real-time trading systems use real-time functions to collect data and generate trading algorithms. Turtle trading system and arbitrage trading system have shown a sharp contrast in their profit and risk behaviour. Using Turtle trading system in cryptocurrency markets got high returns with high risk. Arbitrage trading system is inferior in terms of revenue but also has a lower risk. One feature that turtle trading system and arbitrage trading system have in common is they performed well in capturing alpha.

2.6 Systematic Trading

2.6.1 Technical Analysis

Many researchers have focused on technical indicators (patterns) analysis for trading on cryptocurrency markets. Examples of studies with this approach include “Turtle Soup pattern strategy” [397], “Nem (XEM) strategy” [400], “Amazing Gann Box strategy” [398], “Busted Double Top Pattern strategy” [399], and “Bottom Rotation Trading strategy” [401]. Table 2.7 shows the comparison among these five classical technical trading strategies using technical indicators. “Turtle soup pattern strategy” [397] used a 2-day breakout of price in predicting price trends of cryptocurrencies. This strategy is a kind of chart trading pattern. “Nem (XEM) strategy” combined Rate of Change (ROC) indicator and Relative Strength Index (RSI) in predicting price trends [400]. “Amazing Gann Box” predicted exact points of increase and decrease in Gann Box which are used to catch explosive trends of cryptocurrency price [398]. Technical analysis tools such as candlestick and box charts with Fibonacci Retracement based on golden ratio are used in this technical analysis. Fibonacci Retracement uses horizontal lines to indicate where possible support and resistance levels are in the market. “Busted Double Top Pattern” used a Bearish reversal trading pattern which generates a sell signal to predict price trends [399]. “Bottom
Rotation Trading” is a technical analysis method that picks the bottom before the reversal happens. This strategy used a price chart pattern and box chart as technical analysis tools.

Table 2.7 Comparison among five classical technical trading strategies

<table>
<thead>
<tr>
<th>Technical trading strategy</th>
<th>Core Methods</th>
<th>Technical tools/patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turtle Soup pattern [397]</td>
<td>2-daybreakout of price</td>
<td>Chart trading patterns</td>
</tr>
<tr>
<td>Nem (XEM) [400]</td>
<td>Price trends combined ROC &amp; RSI</td>
<td>Rate of Change indicator (ROC)</td>
</tr>
<tr>
<td>Amazing Gann Box [398]</td>
<td>Predict exact points of rises and falls in Gann Box (catch explosive trends)</td>
<td>Relative strength index (RSI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Candlestick, boxcharts with</td>
</tr>
<tr>
<td>Busted Double Top Pattern [399]</td>
<td>Bearish reversal trading pattern that generates a sell signal</td>
<td>Fibonacci Retracement</td>
</tr>
<tr>
<td>Bottom Rotation Trading [401]</td>
<td>Pick the bottom before the reversal happens</td>
<td>Price chart pattern</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Price chart pattern, box chart</td>
</tr>
</tbody>
</table>

Sungjoo et al. [202] investigated using genetic programming (GP) to find attractive technical patterns in the cryptocurrency market. Over 12 technical indicators including Moving Average (MA) and Stochastic oscillator were used in experiments; adjusted gain, match count, relative market pressure and diversity measures have been used to quantify the attractiveness of technical patterns. With extended experiments, the GP system is shown to find successfully attractive technical patterns, which are useful for portfolio optimization. Hudson et al. [216] applied almost 15,000 to technical trading rules (classified into MA rules, filter rules, support resistance rules, oscillator rules and channel breakout rules). This comprehensive study found that technical trading rules provide investors with significant predictive power and profitability. Corbet et al. [130] analysed various technical trading rules in the form of the moving average-oscillator and trading range break-out strategies to generate higher returns in cryptocurrency markets. By using one-minute dollar-denominated Bitcoin close-price data, the backtest showed variable-length moving average (VMA) rule performs best considering it generates the most useful signals in high frequency trading.

Grovys et al. [196] examined a simple moving average trading strategy using daily price data for the 11 most traded cryptocurrencies over the period 2016-2018. The results showed that, excluding Bitcoin, technical trading rules produced an annualised excess return of 8.76% after controlling for average market returns. The analysis also suggests that cryptocurrency markets are inefficient. AI-Yahyaee et al. [10] examined multiple fractals, long memory processes and efficiency assumptions of major cryptocurrencies using Hurst exponents, time-rolling MF-DFA and quantile regression methods. The results showed that all markets provide evidence of long-term memory properties and multiple fractals. Furthermore, the inefficiency of cryptocurrency markets is time-varying. The researchers concluded that high liquidity with low volatility facilitates arbitrage opportunities for active traders.
2.6 Systematic Trading

2.6.2 Pairs Trading

Pairs trading is a trading strategy that attempts to exploit the mean-reversion between the prices of certain securities. Miroslav [168] investigated the applicability of standard pairs trading approaches on cryptocurrency data with the benchmarks of Gatev et al. [187]. The pairs trading strategy is constructed in two steps. Firstly, suitable pairs with a stable long-run relationship are identified. Secondly, the long-run equilibrium is calculated and pairs trading strategy is defined by the spread based on the values. The research also extended intra-day pairs trading using high frequency data. Overall, the model was able to achieve a 3% monthly profit in Miroslav’s experiments [168]. Broek [408] applied pairs trading based on cointegration in cryptocurrency trading and 31 pairs were found to be significantly cointegrated (within sector and cross-sector). By selecting four pairs and testing over a 60-day trading period, the pairs trading strategy got its profitability from arbitrage opportunities, which rejected the Efficient-market hypothesis (EMH) for the cryptocurrency market. Lintihac et al [282] proposed an optimal dynamic pair trading strategy model for a portfolio of assets. The experiment used stochastic control techniques to calculate optimal portfolio weights and correlated the results with several other strategies commonly used by practitioners including static dual-threshold strategies. Thomas et al. [279] proposed a pairwise trading model incorporating time-varying volatility with constant elasticity of variance type. The experiment calculated the best pair strategy by using a finite difference method and estimated parameters by generalised moment method.

2.6.3 Applicability of using equity trading strategies in cryptocurrency

Trading strategies used in the equity space can also be applied to cryptocurrency assets to some extent. 

**Trend following.** One of the most popular trading strategies in the equity space is trend following. This strategy involves identifying and following trends in the market, and making trades accordingly. This strategy can also be applied to cryptocurrencies, as they are known to have high volatility and exhibit trends that can be identified and followed. Rohrbach et al. [362] used the geometric Brownian motion of the dynamics of financial instrument returns to explain extensively the motivation and context behind each step of a momentum trading strategy. Trading signals are calculated as a mixture of exponential moving averages with different time horizons and the methodology is tested on the global currency markets. When considering a time series-based momentum strategy, this strategy is most suitable for traditional fiat currencies. For cryptocurrencies, the cross-sectional approach is more appropriate and as a result, emerging market currencies and
cryptocurrencies outperform the G10 currencies. Zoicas-lenciu et al. [437] documented the impact of stock volatility on trend-tracking profits for a global sample of 1618 blue-chip stocks, using the price sensitivity of trend signals (i.e. signal volatility) to isolate the adverse effects of high stock volatility exhibited through excessive/inefficient trading. The results show that investors can use ex-post signal volatility estimates as an effective criterion for selecting potential trading rules, providing a good reference strategy for high-volatility markets like cryptocurrencies markets. Gwilym et al. [19] examined the performance of the momentum and time-trading approaches in 32 international equity markets and find that the use of global portfolios to discover momentum strategies is profitable. Although the excess returns of momentum strategies have diminished over the past two decades, the authors find that the trend-following approach significantly reduces the volatility of international equities and provides superior risk smoothing.

**Mean reversion.** Another strategy commonly used in the equity market is mean reversion. This strategy involves buying assets that are underpriced and selling assets that are overpriced, with the expectation that the prices will eventually revert to their mean. This strategy can also be applied to cryptocurrencies, as they often experience significant price fluctuations that may present opportunities for mean reversion trading. Corbet et al. [131] explored whether cryptocurrencies, represented by Bitcoin, regress and conduct experiments on similar asymmetric regression patterns for minute, hourly, daily and weekly returns. The evidence suggested that negative price returns have stronger recovery behaviour in terms of speed and magnitude of recovery compared to positive returns, as well as evidence of positive serial correlation with prior positive price returns. The authors investigated the asymmetry in the persistence of bitcoin price return series using a higher-order ANAR model and found evidence of higher persistence of positive returns than negative returns. Balvers et al. [105] used a combination of momentum and inverse strategies to select from 18 developed equity markets at a monthly frequency and outperform both pure momentum and pure inverse strategies. The results hold after correcting for factor sensitivities and transaction costs. They reveal the importance of controlling for mean reversion when utilising momentum and vice versa. Waser [107] used a momentum threshold autoregressive model, experiments show that there is asymmetric mean-reversion behaviour in return on equity (ROE). The results suggest that ROE adjusts to its long-run mean more slowly when ROE rises than when ROE falls.

**Technical analysis.** Technical analysis is commonly used in both equity and cryptocurrency trading. Technical analysis involves using charts and other tools to analyze past price movements and identify patterns that can be used to predict future price movements. However, it is important to note that cryptocurrencies are often more volatile and subject to rapid price fluctuations than traditional equities, which can make technical analysis more challenging. Gerritsen et al. [190] applied seven trend-following indicators commonly
used in the stock market to assess the profitability of technical trading rules in the Bitcoin market. The results showed that specific technical analysis trading rules, primarily trading range breaks, contain significant predictive power for bitcoin prices, allowing for better performance of buy-and-hold strategies via Sharpe ratios calculated via bootstrap methods. Analysis shows that the added value of the trading range breakout rule performs well in strong markets. Shynkevich’s application [376] of extensive technical trading rules to a set of technology sector and small-cap sector portfolios over the 1995-2010 period yielded superior predictability after adjusting for data snooping bias in the first half of the sample period and statistically significant profits for some portfolios when assuming small to moderate trading costs. This fact suggests that technical analyses are unlikely to have generated abnormal returns for any other segment of the domestic equity market over the past decade. Marshall’s research [311] shows that candlestick charts, the oldest known form of technical analysis, were not profitable in the Japanese stock market during the period 1975-2004. Candlestick technical analysis was developed in Japan in the 1600s and is very popular in Japan as it is closely related to Japanese culture. However, there is no evidence that the candlestick technical trading strategy has added value over the entire 30-year period, over three 10-year subperiods, or in either bull or bear markets.

2.6.4 Price formation in trading strategies

In high-frequency trading (HFT), price formation is based on the continuous matching of buy and sell orders in electronic markets. The process of price formation is influenced by a number of factors, including market liquidity, supply and demand, and news and information about the security being traded.

Since the birth of the market, traders have been trying to find accurate models to use to make a profit. Many studies and experiments have been conducted based on statistical modelling of the stock price data. Some studies attempted to model the limit order book by using statistical approaches, such as using Poisson Processes and Hawkes Processes to estimate the next coming order and to model the state of the limit order book [3, 395]. Brooks et al. [78] pointed out that financial data science and econometrics are highly complementary. The new research paradigm financial data science brings new opportunities for academic research in finance.

Others have used machine learning approaches to estimate the upcoming market condition by applying different machine learning models, such as support vector machine (SVM) [249], convolutional neural network (CNN) [406], Random Forest [149] and recurrent network such as Long-Short-Term-Memory (LSTM) [143]. These studies show that it is possible to use a data-driven approach to discover hidden patterns within the market. In particular, Kercheval and Zhang [249] modelled the high-frequency limit order book
dynamics by using SVM. They discovered that some of the essential features of the order book lie on fundamental features, such as price and volume, and time-insensitive features like mid-price and bid-ask spread. Nousi et al. [335] provided extensive study in high frequency limit order book information in predicting mid-price movements. Support vector machine (SVM), Single layer Feedforward Network (SLFN) and Multilayer Perceptron (MLP) are compared in examining whether the classifier learns the general trends and trends of the stock market by learning from some stocks and applying this knowledge to invisible stocks. The research evaluated these models in solving high speed, variance, quantity of limit order book data and showed that the feature extraction model can discover potential auxiliary knowledge [335]. As from above, Easley et al. [149] investigated price dynamics in future contract markets using random forests. Mäkinen et al. [303] proposed an approach with attention forecasting jump arrivals one-minute ahead in stock prices. This mechanism, convolutional neural network and Long Short-Term memory model are compared in their experiments. Tran et al. [403] considered a neural network Layer structure combining the idea of bilinear projection and enabling this layer to detect and focus on critical time information in financial time-series forecasting. Barbon [33] proposed an encoder-decoder neural network augmented with an attention-based mechanism in predicting future transaction prices in NASDAQ. The results showed the model’s behavior prefers liquidity provision rather than front-running strategies. Gu et al. [197] compared different machine learning methods for the canonical problem of empirical asset pricing. Tree and neural networks are best-performing methods in measuring asset risk premiums from their research. Verstyuk [409] took a range of small to large-scale Long Short-Term Memory recurrent neural networks and compared those to the VAR method. These methods are used to model multivariate time series such as GDP growth, inflation, commodity prices etc. The results showed that the neural networks used may also be a useful tool for policy simulation under actual relevant economic conditions, which can also discover different macroeconomic regimes. Finally, Sirignano and Cont [382] suggested that there might be some universal features on the stock market’s limit order book that have a non-linear relationship to the price change. They tried to predict the mid-price movement of the next tick by training a neural network using a significant amount of stock data. Their findings suggest that instead of building a stock-specific model, a universal model for all kinds of stock could be built.

Most of the studies in the area focus on the traditional stock market like NYSE and NASDAQ [201]. Many researchers have studied these exchanges for many years. The quality of data and the market environment are more desirable than those of the cryptocurrency market. Although the traditional stock market may provide a less volatile and more regulated environment for traders, the high volatility of the cryptocurrency market may provide a higher potential return.
Understanding price formation through mid-price for cryptocurrencies becomes even harder due to the fact that they are distinct from traditional fiat-based currencies. The latter are usually issued by banks or governments. The only way to create Bitcoins, the currently dominant cryptocurrency, is to run a computationally intensive algorithm to add new blocks to the blockchain. People who participate in this processing will verify transactions on the blockchain, and try to earn Bitcoins as the reward of adding new blocks. These people are usually referred to as Bitcoins miners. The protocol of the Bitcoin fixed its total supply at 21 million [326]. Every transaction on the blockchain is protected by a cryptographic hash algorithm called SHA-256. It is a computational intensive hash algorithm that is implemented to verify blocks on the blockchain. For instance, if a counterfeiter wants to forge a block on the blockchain, they will also need to redo all the hashing before that block. This property provides a trustless foundation for Bitcoin because neither an individual nor an institution can counterfeit the currency or the transaction unless it has a computational power in excess of the majority of the network [326].

Multi-label prediction is widely used in image processing, character recognition and forecasting of decisions or time series. Complex trading strategies might require more than binary classification. One may use the status box method to measure different stock statuses such as turning point, flat box and up-down box [432] in order to reflect the relative position of the stock and classify whether the state coincides with the stock price trend.

Marcus [309] points critical aspects on the application of machine learning, especially deep learning, which provide insights for finance. First, machine learning models are data-hungry. Marcus gives an example in his research: one cannot rely on millions of training examples to represent abstract relationships between similar algebraic variables. Accordingly, in financial predictions, machine learning models might be misaligned due to limited training examples. Second, the knowledge gathered by deep learning systems is primarily concerned with correlations between features, rather than abstractions like quantified statements. For these reasons, machine learning is not yet able to reach human-level cognitive flexibility. Ultimately, a machine learning algorithm applied to finance needs to improve adaptability when facing a new market or order structure.

2.6.5 Others

Other systematic trading methods in cryptocurrency trading mainly include informed trading. Using USD / BTC exchange rate trading data, Feng et al. [167] found evidence of informed trading in the Bitcoin market in those quantiles of the order sizes of buyer-initiated (seller-initiated) orders are abnormally high before large positive (negative) events, compared to the quantiles of seller-initiated (buyer-initiated) orders; this study adopts a new indicator inspired by the volume imbalance indicator [150]. The evidence of informed
trading in the Bitcoin market suggests that investors profit on their private information when they get information before it is widely available.

2.7 Emergent Trading Technologies

2.7.1 Econometrics on cryptocurrency

Copula-quantile causality analysis and Granger-causality analysis are methods to investigate causality in cryptocurrency trading analysis. Bouri et al. [68] applied a copula-quantile causality approach on volatility in the cryptocurrency market. The approach of the experiment extended the Copula-Granger-causality in distribution (CGCD) method of Lee and Yang [276] in 2014. The experiment constructed two tests of CGCD using copula functions. The parametric test employed six parametric copula functions to discover dependency density between variables. The performance matrix of these functions varies with independent copula density. Three distribution regions are the focus of this research: left tail (1%, 5%, 10% quantile), central region (40%, 60% quantile and median) and right tail (90%, 95%, 99% quantile). The study provided significant evidence of Granger causality from trading volume to the returns of seven large cryptocurrencies on both left and right tails. Elie et al. [69] examined the causal linkages among the volatility of leading cryptocurrencies via the frequency-domain test of Bodart and Candelon [63] and distinguished between temporary and permanent causation. The results showed that permanent shocks are more important in explaining Granger causality whereas transient shocks dominate the causality of smaller cryptocurrencies in the long term. Badenhorst [28] attempted to reveal whether spot and derivative market volumes affect Bitcoin price volatility with the Granger-causality method and ARCH (1,1). The result shows spot trading volumes have a significant positive effect on price volatility while the relationship between cryptocurrency volatility and the derivative market is uncertain. Elie et al. [72] used a dynamic equicorrelation (DECO) model and reported evidence that the average earnings equilibrium correlation changes over time between the 12 leading cryptocurrencies. The results showed increased cryptocurrency market consolidation despite significant price declined in 2018. Furthermore, measurement of trading volume and uncertainty are key determinants of integration.

Several econometrics methods in time-series research, such as GARCH and BEKK, have been used in the literature on cryptocurrency trading. Conrad et al. [125] used the GARCH-MIDAS model to extract long and short-term volatility components of the Bitcoin market. The technical details of this model decomposed the conditional variance into the low-frequency and high-frequency components. The results identified that S&P 500 realized volatility has a negative and highly significant effect on long-term Bitcoin
2.7 Emergent Trading Technologies

volatility and S&P 500 volatility risk premium has a significantly positive effect on long-term Bitcoin volatility. Ardia et al. [20] used the Markov Switching GARCH (MSGARCH) model to test the existence of institutional changes in the GARCH volatility dynamics of Bitcoin’s logarithmic returns. Moreover, a Bayesian method was used for estimating model parameters and calculating VaR prediction. The results showed that MSGARCH models clearly outperform single-regime GARCH for Value-at-Risk forecasting. Troster et al. [404] performed general GARCH and GAS (Generalized Auto-regressive Score) analysis to model and predict Bitcoin’s returns and risks. The experiment found that the GAS model with heavy-tailed distribution can provide the best out-of-sample prediction and goodness-of-fit attributes for Bitcoin’s return and risk modeling. The results also illustrated the importance of modeling excess kurtosis for Bitcoin returns.

Charles et al. [104] studied four cryptocurrency markets including Bitcoin, Dash, Litecoin and Ripple. Results showed cryptocurrency returns are strongly characterised by the presence of jumps as well as structural breaks except the Dash market. Four GARCH-type models (i.e., GARCH, APARCH, IGARCH and FIGARCH) and three return types with structural breaks (original returns, jump-filtered returns, and jump-filtered returns) are considered. The research indicated the importance of jumps in cryptocurrency volatility and structural breakthroughs. Malladi et al. [304] examined the time series analysis of Bitcoin and Ripple’s returns and volatility to examine the dependence of their prices in part on global equity indices, gold prices and fear indicators such as volatility indices and US economic policy uncertainty indices. Autoregressive-moving-average model with exogenous inputs model (ARMAX), GARCH, VAR and Granger causality tests are used in the experiments. The results showed that there is no causal relationship between global stock market and gold returns on bitcoin returns, but a causal relationship between ripple returns on bitcoin prices is found.

Some researchers focused on long memory methods for volatility in cryptocurrency markets. Long memory methods focused on long-range dependence and significant long-term correlations among fluctuations on markets. Chaim et al. [101] estimated a multivariate stochastic volatility model with discontinuous jumps in cryptocurrency markets. The results showed that permanent volatility appears to be driven by major market developments and popular interest levels. Caporale et al. [88] examined persistence in the cryptocurrency market by Rescaled range (R/S) analysis and fractional integration. The results of the study indicated that the market is persistent (there is a positive correlation between its past and future values) and that its level changes over time. Khuntin et al. [251] applied the adaptive market hypothesis (AMH) in the predictability of Bitcoin evolving returns. The consistent test of Dominguez and Lobato [144], generalized spectral (GS) of Escanciano and Velasco [157] are applied in capturing time-varying linear and nonlinear dependence in bitcoin returns. The results verified Evolving Efficiency in Bitcoin price.
changes and evidence of dynamic efficiency in line with AMH’s claims. Gradojevic et al. [192] examined volatility cascades across multiple trading ranges in the cryptocurrency market. Using a wavelet Hidden Markov Tree model, authors estimated the transition probability of propagating high or low volatility at one time scale (range) to high or low volatility at the next time scale. The results showed that the volatility cascade tends to be symmetrical when moving from long to short term. In contrast, when moving from short to long term, the volatility cascade is very asymmetric.

Nikolova et al. [333] provided a new method to calculate the probability of volatility clusters, especially for cryptocurrencies (high volatility of their exchange rates). The authors used the FD4 method to calculate the Hurst index of a volatility series and describe explicit criteria for determining the existence of fixed size volatility clusters by calculation. The results showed that the volatility of cryptocurrencies changes more rapidly than that of traditional assets, and much more rapidly than that of Bitcoin/USD, Ethereum/USD, and Ripple/USD pairs. Ma et al. [300] investigated whether a new Markov Regime Transformation Mixed Data Sampling (MRS-MIADS) model can improve the prediction accuracy of Bitcoin’s Realised Variance (RV). The results showed that the proposed new MRS-MIDAS model exhibits statistically significant improvements in predicting the RV of Bitcoin. At the same time, the occurrence of jumps significantly increases the persistence of high volatility and switches between high and low volatility.

Katsiampa et al. [243] applied three pair-wise bivariate BEKK models to examine the conditional volatility dynamics along with interlinkages and conditional correlations between three pairs of cryptocurrencies in 2018. More specifically, the BEKK-MGARCH methodology also captured cross-market effects of shocks and volatility, which are also known as shock transmission effects and volatility spillover effects. The experiment found evidence of bi-directional shock transmission effects between Bitcoin and both Ether and Litecoin. In particular, bi-directional shock spillover effects are identified between three pairs (Bitcoin, Ether and Litecoin) and time-varying conditional correlations exist with positive correlations mostly prevailing. In 2019, Katsiampa [242] further researched an asymmetric diagonal BEKK model to examine conditional variances of five cryptocurrencies that are significantly affected by both previous squared errors and past conditional volatility. The experiment tested the null hypothesis of the unit root against the stationarity hypothesis. Once stationarity is ensured, ARCH LM is tested for ARCH effects to examine the requirement of volatility modeling in return series. Moreover, volatility co-movements among cryptocurrency pairs are also tested by the multivariate GARCH model. The results confirmed the non-normality and heteroskedasticity of price returns in cryptocurrency markets. The finding also identified the effects of cryptocurrencies’ volatility dynamics due to major news.
2.7 Emergent Trading Technologies

Hultman [217] set out to examine GARCH (1,1), bivariate-BEKK (1,1) and a standard stochastic model to forecast the volatility of Bitcoin. A rolling window approach is used in these experiments. Mean absolute error (MAE), Mean squared error (MSE) and Root-mean-square deviation (RMSE) are three loss criteria adopted to evaluate the degree of error between predicted and true values. The result shows the following rank of loss functions: GARCH (1,1) > bivariate-BEKK (1,1) > Standard stochastic for all the three different loss criteria; in other words, GARCH(1,1) appeared best in predicting the volatility of Bitcoin. Wavelet time-scale persistence analysis is also applied in the prediction and research of volatility in cryptocurrency markets [340]. The results showed that information efficiency (efficiency) and volatility persistence in the cryptocurrency market are highly sensitive to time scales, measures of returns and volatility, and institutional changes. Adjepong et al. [340] connected with similar research by Corbet et al. [134] and showed that GARCH is quicker than BEKK to absorb new information regarding the data.

Zhang et al. [431] examined how to price exceptional volatility in a cross-section of cryptocurrency returns. Using portfolio-level analysis and Fama-MacBeth regression analysis, the authors demonstrated that idiosyncratic volatility is positively correlated with expected returns on cryptocurrencies.

2.7.2 Machine Learning Technology

As we have previously stated, Machine learning technology constructs computer algorithms that automatically improve themselves by finding patterns in existing data without explicit instructions [210]. The rapid development of machine learning in recent years has promoted its application to cryptocurrency trading, especially in the prediction of cryptocurrency returns. Some ML algorithms solve both classification and regression problems from a methodological point of view. For clearer classification, we focus on the application of these ML algorithms in cryptocurrency trading. For example, Decision Tree (DT) can solve both classification and regression problems. But in cryptocurrency trading, researchers focus more on using DT in solving classification problems. Here we classify DT as “Classification Algorithms”.

Common Machine Learning Technology in this survey

Several machine learning technologies are applied in cryptocurrency trading. We distinguish these by the objective set to the algorithm: classification, clustering, regression, reinforcement learning. We have separated a section specifically on deep learning due to its intrinsic variation of techniques and wide adoption.

Classification Algorithms. Classification in machine learning has the objective of categorising incoming objects into different categories as needed, where we can assign
labels to each category (e.g., up and down). Naive Bayes (NB) [360], Support Vector Machine (SVM) [418], K-Nearest Neighbours (KNN) [418], Decision Tree (DT) [179], Random Forest (RF) [281] and Gradient Boosting (GB) [182] algorithms have been used in cryptocurrency trading based on papers we collected. NB is a probabilistic classifier based on Bayes’ theorem with strong (naive) conditional independence assumptions between features [360]. SVM is a supervised learning model that aims at achieving high margin classifiers connecting to learning bounds theory [428]. SVMs assign new examples to one category or another, making it a non-probabilistic binary linear classifier [418], although some corrections can make a probabilistic interpretation of their output [247]. KNN is a memory-based or lazy learning algorithm, where the function is only approximated locally, and all calculations are being postponed to inference time [418]. DT is a decision support tool algorithm that uses a tree-like decision graph or model to segment input patterns into regions to then assign an associated label to each region [179, 161]. RF is an ensemble learning method. The algorithm operates by constructing a large number of decision trees during training and outputting the average consensus as predicted class in the case of classification or mean prediction value in the case of regression [281]. GB produces a prediction model in the form of an ensemble of weak prediction models [182].

**Clustering Algorithms.** Clustering is a machine learning technique that involves grouping data points in a way that each group shows some regularity [228]. K-Means is a vector quantization used for clustering analysis in data mining. K-means stores the k-centroids used to define the clusters; a point is considered to be in a particular cluster if it is closer to the cluster’s centroid than any other centroid [414]. K-Means is one of the most used clustering algorithms used in cryptocurrency trading according to the papers we collected. Clustering algorithms have been successfully applied in many financial applications, such as fraud detection, rejection inference and credit assessment. Automated detection clusters are critical as they help to understand sub-patterns of data that can be used to infer user behaviour and identify potential risks [278, 261].

**Regression Algorithms.** We have defined regression as any statistical technique that aims at estimating a continuous value [269]. Linear Regression (LR) and Scatterplot Smoothing are common techniques used in solving regression problems in cryptocurrency trading. LR is a linear method used to model the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables) [269]. Scatterplot Smoothing is a technology to fit functions through scatter plots to best represent relationships between variables [181].

**Deep Learning Algorithms.** Deep learning is a modern take on artificial neural networks (ANNs) [430], made possible by the advances in computational power. An ANN is a computational system inspired by the natural neural networks that make up the animal’s brain. The system “learns” to perform tasks including the prediction by
considering examples. Deep learning’s superior accuracy comes from high computational complexity cost. Deep learning algorithms are currently the basis for many modern artificial intelligence applications [392]. Convolutional neural networks (CNNs) [274], Recurrent neural networks (RNNs) [320], Gated recurrent units (GRUs) [113], Multilayer perceptron (MLP) and Long short-term memory (LSTM) [109] networks are the most common deep learning technologies used in cryptocurrency trading. A CNN is a specific type of neural network layer commonly used for supervised learning. CNNs have found their best success in image processing and natural language processing problems. An attempt to use CNNs in cryptocurrency can be shown in [235].

An RNN is a type of artificial neural network in which connections between nodes form a directed graph with possible loops. This structure of RNNs makes them suitable for processing time-series data [320] due to the introduction of memory in the recurrent connections. They face nevertheless for the vanishing gradients problem [341] and so different variations have been recently proposed. LSTM [109] is a particular RNN architecture widely used. LSTMs have shown to be superior to nongated RNNs on financial time-series problems because they have the ability to selectively remember patterns for a long time. A GRU [113] is another gated version of the standard RNN which has been used in crypto trading [147]. Another deep learning technology used in cryptocurrency trading is Seq2seq, which is a specific implementation of the Encoder–Decoder architecture [422]. Seq2seq was first aimed at solving natural language processing problems but has been also applied it in cryptocurrency trend predictions in [384].

**Reinforcement Learning Algorithms.** Reinforcement learning (RL) is an area of machine learning leveraging the idea that software agents act in the environment to maximize a cumulative reward [391]. Deep Q-Learning (DQN) [198] and Deep Boltzmann Machine (DBM) [366] are common technologies used in cryptocurrency trading using RL. Deep Q learning uses neural networks to approximate Q-value functions. A state is given as input, and Q values for all possible actions are generated as outputs [198]. DBM is a type of binary paired Markov random field (undirected probability graphical model) with multiple layers of hidden random variables [366]. It is a network of randomly coupled random binary units.

**Research on Machine Learning Models**

In the development of machine learning trading signals, technical indicators have usually been used as input features. Nakano et al. [328] explored Bitcoin intraday technical trading based on ANNs for return prediction. The experiment obtained medium frequency price and volume data (time interval of data is 15min) of Bitcoin from a cryptocurrency exchange. An ANN predicts the price trends (up and down) in the next period from the
input data. Data is preprocessed to construct a training dataset that contains a matrix of technical patterns including EMA, Emerging Markets Small Cap (EMSD), relative strength index (RSI), etc. Their numerical experiments contain different research aspects including base ANN research, effects of different layers, effects of different activation functions, different outputs, different inputs and effects of additional technical indicators. The results have shown that the use of various technical indicators possibly prevents over-fitting in the classification of non-stationary financial time-series data, which enhances trading performance compared to the primitive technical trading strategy. (Buy-and-Hold is the benchmark strategy in this experiment.)

Some classification and regression machine learning models are applied in cryptocurrency trading by predicting price trends. Most researchers have focused on the comparison of different classification and regression machine learning methods. Sun et al. [388] used random forests (RFs) with factors in Alpha01 [233] (capturing features from the history of the cryptocurrency market) to build a prediction model. The experiment collected data from API in cryptocurrency exchanges and selected 5-minute frequency data for backtesting. The results showed that the performances are proportional to the amount of data (more data, more accurate) and the factors used in the RF model appear to have different importance. For example, “Alpha024” and “Alpha032” features appeared as the most important in the model adopted. (The alpha features come from paper “101 Formulaic Alphas” [233].) Vo et al. [412] applied RFs in High-Frequency cryptocurrency Trading (HFT) and compared it with deep learning models. Minute-level data is collected when utilising a forward fill imputation method to replace the NULL value (i.e., a missing value). Different periods and RF trees are tested in the experiments. The authors also compared F-1 precision and recall metrics between RF and Deep Learning (DL). The results showed that RF is effective despite multicollinearity occurring in ML features, the lack of model identification also potentially leading to model identification issues; this research also attempted to create an HFT strategy for Bitcoin using RF.

Maryna et al. [438] investigated the profitability of an algorithmic trading strategy based on training an SVM model to identify cryptocurrencies with high or low predicted returns. The results showed that the performance of the SVM strategy was the fourth being better only than S&P B&H strategy, which simply buys-and-hold the S&P index. (There are other 4 benchmark strategies in this research.) The authors observed that SVM needs a large number of parameters and so is very prone to overfitting, which caused its bad performance. Barnwal et al. [35] used generative and discriminative classifiers to create a stacking model, particularly 3 generative and 6 discriminative classifiers combined by a one-layer Neural Network, to predict the direction of cryptocurrency price. A discriminative classifier directly models the relationship between unknown and known data, while generative classifiers model the prediction indirectly through the data generation distribution [332].
Technical indicators including trend, momentum, volume and volatility, are collected as features of the model. The authors discussed how different classifiers and features affect the prediction. Attanasio et al. [24] compared a variety of classification algorithms including SVM, NB and RF in predicting next-day price trends of a given cryptocurrency. The results showed that due to the heterogeneity and volatility of cryptocurrencies’ financial instruments, forecasting models based on a series of forecasts appeared better than a single classification technology in trading cryptocurrencies. Madan et al. [301] modeled the Bitcoin price prediction problem as a binomial classification task, experimenting with a custom algorithm that leverages both random forests and generalized linear models. Daily data, 10-minute data and 10-second data are used in the experiments. The experiments showed that 10-minute data gave a better sensitivity and specificity ratio than 10-second data (10-second prediction achieved around 10% accuracy). Considering predictive trading, 10-minute data helped show clearer trends in the experiment compared to 10-second backtesting. Similarly, Virk [411] compared RF, SVM, GB and LR to predict the price of Bitcoin. The results showed that SVM achieved the highest accuracy of 62.31% and precision value 0.77 among binomial classification machine learning algorithms.

Different deep learning models have been used in finding patterns of price movements in cryptocurrency markets. Zhengy et al. [435] implemented two machine learning models, fully-connected ANN and LSTM to predict cryptocurrency price dynamics. The results showed that ANN, in general, outperforms LSTM although theoretically, LSTM is more suitable than ANN in terms of modeling time series dynamics; the performance measures considered are MAE and RMSE in joint prediction (five cryptocurrencies daily prices prediction). The findings show that the future state of a time series for cryptocurrencies is highly dependent on its historic evolution. Kwon et al. [270] used an LSTM model, with a three-dimensional price tensor representing the past price changes of cryptocurrencies as input. This model outperforms the GB model in terms of F1-score. Specifically, it has a performance improvement of about 7% over the GB model in 10-minute price prediction. In particular, the experiments showed that LSTM is more suitable when classifying cryptocurrency data with high volatility.

Alessandretti et al. [11] tested Gradient boosting decision trees (including single regression and XGBoost-augmented regression) and the LSTM model on forecasting daily cryptocurrency prices. They found methods based on gradient boosting decision trees worked best when predictions were based on short-term windows of 5/10 days while LSTM worked best when predictions were based on 50 days of data. The relative importance of the features in both models are compared and an optimised portfolio composition (based on geometric mean return and Sharpe ratio) is discussed in this paper. Phaladisailoed et al. [345] chose regression models (Theil-Sen Regression and Huber Regression) and deep learning-based models (LSTM and GRU) to compare the performance of predicting
the rise and fall of Bitcoin price. In terms of two common measure metrics, MSE and R-Square ($R^2$), GRU shows the best accuracy.

Researchers have also focused on comparing classical statistical models and machine/deep learning models. Rane et al. [355] described classical time series prediction methods and machine learning algorithms used for predicting Bitcoin price. Statistical models such as Autoregressive Integrated Moving Average models (ARIMA), Binomial Generalized Linear Model and GARCH are compared with machine learning models such as SVM, LSTM and Non-linear Auto-Regressive with Exogenous Input Model (NARX). The observation and results showed that the NARX model is the best model with nearly 52% predicting accuracy based on 10 seconds interval. Rebane et al. [357] compared traditional models like ARIMA with a modern popular model like seq2seq in predicting cryptocurrency returns. The result showed that the seq2seq model exhibited demonstrable improvement over the ARIMA model for Bitcoin-USD prediction but the seq2seq model showed very poor performance in extreme cases. The authors proposed performing additional investigations, such as the use of LSTM instead of GRU units to improve the performance. Similar models were also compared by Stuerner et al. [387] who explored the superiority of automated investment approach in trend following and technical analysis in cryptocurrency trading. Samuel et al. [344] explored the vector autoregressive model (VAR model), a more complex RNN, and a hybrid of the two in residual recurrent neural networks (R2N2) in predicting cryptocurrency returns. The RNN with ten hidden layers is optimised for the setting and the neural network augmented by VAR allows the network to be shallower, quicker and to have a better prediction than an RNN. RNN, VAR and R2N2 models are compared. The results showed that the VAR model has phenomenal test period performance and thus props up the R2N2 model, while the RNN performs poorly. This research is an attempt at optimisation of model design and applying to the prediction on cryptocurrency returns.

**Deep Neural Network**

Deep Neural Network architectures play important roles in forecasting. Researchers had applied many advanced deep neural network models in cryptocurrency trading like stacking (CNN + RNN) and Autoencoder-Decoder. In this subsection, we describe the cutting edge Deep Neural Network researches in cryptocurrency trading. Recent studies show the productivity of using models based on such architectures for modeling and forecasting financial time series, including cryptocurrencies. Livieris et al. [291] proposed model called CNN-LSTM for accurate prediction of gold prices and movements. The first component of the model consists of a convolutional layer and a pooling layer, where complex mathematical operations are performed to develop the features of the input data.
2.7 Emergent Trading Technologies

The second component uses the generated LSTM and the features of the dense layer. The results show that due to the sensitivity of the various hyperparameters of the proposed CNN-LSTM and its high complexity, additional optimisation configurations and major feature engineering have the potential to further improve the predictive power. More Intelligent Evolutionary Optimisation (IEO) for hyperparameter optimisation is a core problem when tuning the overall optimization process of machine learning models [214]. Lu et al. [295] proposed a CNN-LSTM based method for stock price prediction. In terms of MAE, RMSE and $R^2$ metrics, the experimental results showed that CNN-LSTM has the highest prediction accuracy and the best performance compared with MLP, CNN, RNN, LSTM, and CNN-RNN.

Fan et al. [160] applied an autoencoder-augmented LSTM structure in predicting the mid-price of 8 cryptocurrency pairs. Level-2 limit order book live data is collected and the experiment achieved 78% accuracy of price movements prediction in high frequency trading (tick level). This research improved and verified the view of Sirignano et al. [381] that universal models have better performance than currency-pair specific models for cryptocurrency markets. Moreover, “Walkthrough” (i.e., retrain the original deep learning model itself when it appears to no longer be valid) is proposed as a method to optimise the training of a deep learning model and shown to significantly improve the prediction accuracy. Yao et al. [424] proposed a new method for predicting cryptocurrency prices based on deep learning techniques such as RNN and LSTM, taking into account various factors such as market capitalization, trading volume, circulating supply and maximum supply. The experimental results showed that the model performs well for a certain size of dataset. Livieris et al. [292] combined three of the most widely used integration learning strategies: integrated averaging, bagging and stacking, with advanced deep learning models for predicting hourly prices of major cryptocurrencies. The proposed integrated model is evaluated using a state-of-the-art deep learning model as a component learner, which consists of a combination of LSTM, bidirectional LSTM and convolutional layers. The authors’ detailed experimental analysis shows that integrated learning and deep learning can effectively reinforce each other to develop robust, stable and reliable predictive models. Kumar et al. [267] analyzed how deep learning techniques such as MLP and LSTM can help predict the price trend of Ethereum. By applying day/hour/minute historical data, the LSTM model is more robust and accurate to long-term dependencies than the MLP while LSTM outperformed the MLP marginally but not very significantly.
Sentiment Analysis

Sentiment analysis, a popular research topic in the age of social media, has also been adopted to improve predictions for cryptocurrency trading. This data source typically has to be combined with Machine Learning for the generation of trading signals.

Lamon et al. [272] used daily news and social media data labeled on actual price changes, rather than on positive and negative sentiment. By this approach, the prediction on price is replaced with positive and negative sentiment. The experiment acquired cryptocurrency-related news article headlines from the website like “cryptocoinsnews” and twitter API. Weights are taken in positive and negative words in the cryptocurrency market. Authors compared Logistic Regression (LR), Linear Support Vector Machine (LSVM) and NB as classifiers and concluded that LR is the best classifier in daily price prediction with 43.9% of price increases correctly predicted and 61.9% of price decreases correctly forecasted. Smuts [383] conducted a similar binary sentiment-based price prediction method with an LSTM model using Google Trends and Telegram sentiment. In detail, the sentiment was extracted from Telegram by using a novel measure called VADER [218]. The backtesting reached 76% accuracy on the test set during the first half of 2018 in predicting hourly prices.

Nasir et al. [330] researched the relationship between cryptocurrency returns and search engines. The experiment employed a rich set of established empirical approaches including VAR framework, copulas approach and non-parametric drawings of time series. The results found that Google searches exert significant influence on Bitcoin returns, especially in the short-term intervals. Kristoufek [266] discussed positive and negative feedback on Google trends or daily views on Wikipedia. The author mentioned different methods including Cointegration, Vector autoregression and Vector error-correction model to find causal relationships between prices and searched terms in the cryptocurrency market. The results indicated that search trends and cryptocurrency prices are connected. There is also a clear asymmetry between the effects of increased interest in currencies above or below their trend values from the experiment. Young et al. [253] analysed user comments and replies in online communities and their connection with cryptocurrency volatility. After crawling comments and replies in online communities, authors tagged the extent of positive and negative topics. Then the relationship between price and the number of transactions of cryptocurrency is tested according to comments and replies to selected data. At last, a prediction model using machine learning based on selected data is created to predict fluctuations in the cryptocurrency market. The results show the amount of accumulated data and animated community activities exerted a direct effect on fluctuation in the price and volume of a cryptocurrency.
2.7 Emergent Trading Technologies

Phillips et al. [350] applied dynamic topic modeling and Hawkes model to decipher relationships between topics and cryptocurrency price movements. The authors used Latent Dirichlet allocation (LDA) model for topic modeling, which assumes each document contains multiple topics to different extents. The experiment showed that particular topics tend to precede certain types of price movements in the cryptocurrency market and the authors proposed the relationships could be built into real-time cryptocurrency trading. Li et al. [280] analysed Twitter sentiment and trading volume and an Extreme Gradient Boosting Regression Tree Model in the prediction of ZClassic (ZCL) cryptocurrency market. Sentiment analysis using natural language processing from the Python package “Textblob” assigns impactful words a polarity value. Values of weighted and unweighted sentiment indices are calculated on an hourly basis by summing weights of coinciding tweets, which makes us compare this index to ZCL price data. The model achieved a Pearson correlation of 0.806 when applied to test data, yielding a statistical significance at the \( p < 0.0001 \) level. Flori [173] relied on a Bayesian framework that combines market-neutral information with subjective beliefs to construct diversified investment strategies in the Bitcoin market. The result shows that news and media attention seem to contribute to influence the demand for Bitcoin and enlarge the perimeter of the potential investors, probably stimulating price euphoria and upwards-downwards market dynamics. The authors’ research highlighted the importance of news in guiding portfolio re-balancing.

Elie et al. [66] compared the ability of newspaper-based metrics and internet search-based uncertainty metrics in predicting bitcoin returns. The predictive power of Internet-based economic uncertainty-related query indices is statistically stronger than that of newspapers in predicting bitcoin returns.

Similarly, Colianni et al. [123], Garcia et al. [185], Zamuda et al. [426] et al. used sentiment analysis technology applying it in the cryptocurrency trading area and had similar results. Colianni et al. [123] cleaned data and applied supervised machine learning algorithms such as logistic regression, Naive Bayes and support vector machines, etc. on Twitter Sentiment Analysis for cryptocurrency trading. Garcia et al. [185] applied multidimensional analysis and impulse analysis in social signals of sentiment effects and algorithmic trading of Bitcoin. The results verified the long-standing assumption that transaction-based social media sentiment has the potential to generate a positive return on investment. Zamuda et al. [426] adopted new sentiment analysis indicators and used multi-target portfolio selection to avoid risks in cryptocurrency trading. The perspective is rationalized based on the elastic demand for computing resources of the cloud infrastructure. A general model evaluating the influence between user’s network Action-Reaction-Influence-Model (ARIM) is mentioned in this research. Bartolucci et al. [37] researched cryptocurrency prices with the “Butterfly effect”, which means “issues” of the open-source project provides insights to improve prediction of cryptocurrency
2.7 Emergent Trading Technologies

prices. Sentiment, politeness, emotions analysis of GitHub comments are applied in Ethereum and Bitcoin markets. The results showed that these metrics have predictive power on cryptocurrency prices. Johannes et al. [223] applied deep learning machine learning models to improve bitcoin price prediction and trading by integrating unstructured information from financial news. The authors first extracted structured information from financial news using LSTM networks, and then apply the machine learning model with structured inputs from financial news to predict the bitcoin price. The prediction system achieved a significantly higher expiration return than the buy-and-hold strategy.

Reinforcement Learning

Deep reinforcement algorithms bypass prediction and go straight to market management actions to achieve high cumulated profit [207, 283]. Bu et al. [82] proposed a combination of double Q-network and unsupervised pre-training using DBM to generate and enhance the optimal Q-function in cryptocurrency trading. The trading model contains agents in series in the form of two neural networks, unsupervised learning modules and environments. The input market state connects an encoding network which includes spectral feature extraction (convolution-pooling module) and temporal feature extraction (LSTM module). A double-Q network follows the encoding network and actions are generated from this network. Compared to existing deep learning models (LSTM, CNN, MLP, etc.), this model achieved the highest profit even facing an extreme market situation (recorded 24% of the profit while cryptocurrency market price drops by -64%). Juchli [229] applied two implementations of reinforcement learning agents, a Q-Learning agent, which serves as the learner when no market variables are provided, and a DQN agent which was developed to handle the features previously mentioned. The DQN agent was backtested under the application of two different neural network architectures. The results showed that the DQN-CNN agent (convolutional neural network) is superior to the DQN-MLP agent (multilayer perceptron) in backtesting prediction. Lucarelli et al. [296] focused on improving automated cryptocurrency trading with a deep reinforcement learning approach. Double and Dueling double deep Q-learning networks are compared for 4 years. By setting rewards functions as Sharpe ratio and profit, the double Q-learning method demonstrated to be the most profitable approach in trading cryptocurrency. Sattarov et al. [367] applied deep reinforcement learning and used historical data from BTC, LTC and ETH to observe historical price movements and acted on real-time prices. The model proposed by the authors helped traders to correctly choose one of the following three actions: buy, sell and hold stocks and get advice on the correct option. Experiments applying BTC via deep reinforcement learning showed that investors made a net profit of 14.4% in one month. Similarly, tests on LTC and ETH ended with 74% and 41% profits respectively. Koker et
2.7 Emergent Trading Technologies

al. [256] pointed out direct reinforcement (DR) based model for active trading. Within the model, the authors attempt to estimate the parameters of the non-linear autoregressive model to achieve maximum risk-adjusted returns. Traders can take long or short positions in each of our sampled cryptocurrency markets, establish or hold them at the end of time interval \( t \), and re-evaluate at the end of \( t + 1 \). The results provide some preliminary evidence that cryptocurrency prices may not follow a purely random wandering process.

2.7.3 Others

Atsalakis et al. [23] proposes a computational intelligence technique that uses a hybrid Neuro-Fuzzy controller, namely PATSOS, to forecast the direction in the change of the daily price of Bitcoin. The proposed methodology outperforms two other computational intelligence models, the first being developed with a simpler neuro-fuzzy approach, and the second being developed with artificial neural networks. According to the signals of the proposed model, the investment return obtained through trading simulation is 71.21% higher than the investment return obtained through a simple buy and hold strategy. This application is proposed for the first time in the forecasting of Bitcoin price movements.

Topological data analysis is applied to forecasting price trends of cryptocurrency markets in [252]. The approach is to harness topological features of attractors of dynamical systems for arbitrary temporal data. The results showed that the method can effectively separate important topological patterns and sampling noise (like bid–ask bounces, discreteness of price changes, differences in trade sizes or informational content of price changes, etc.) by providing theoretical results. Kurbucz [268] designed a complex method consisting of single-hidden layer feedforward neural networks (SLFNs) to (i) determine the predictive power of the most frequent edges of the transaction network (a public ledger that records all Bitcoin transactions) on the future price of Bitcoin; and, (ii) to provide an efficient technique for applying this untapped dataset in day trading. The research found a significantly high accuracy (60.05%) for the price movement classifications base on information that can be obtained using a small subset of edges (approximately 0.45% of all unique edges). It is worth noting that, Kondor et al. [257, 259] firstly published some papers giving analysis on transaction networks on cryptocurrency markets and applied related research in identifying Bitcoin users [230].

Abay et al. [2] attempted to understand the network dynamics behind the Blockchain graphs using topological features. The results showed that standard graph features such as the degree distribution of transaction graphs may not be sufficient to capture network dynamics and their potential impact on Bitcoin price fluctuations. Maurice et al [340] applied wavelet time-scale persistence in analysing returns and volatility in cryptocurrency markets. The experiment examined the long-memory and market efficiency characteristics.
in cryptocurrency markets using daily data for more than two years. The authors employed a log-periodogram regression method in researching stationarity in the cryptocurrency market and used ARFIMA-FIGARCH class of models in examining long-memory behaviour of cryptocurrencies across time and scale. In general, experiments indicated that heterogeneous memory behaviour existed in eight cryptocurrency markets using daily data over the full-time period and across scales (August 25, 2015 to March 13, 2018).

2.8 Portfolio, Cryptocurrency Assets and Market Condition Research

2.8.1 Research among cryptocurrency pairs and related factors

Ji et al. [226] examined connectedness via return and volatility spillovers across six large cryptocurrencies (collected from coinmarketcap lists from August 7 2015 to February 22 2018) and found Litecoin and Bitcoin to have the most effect on other cryptocurrencies. The authors followed methods of Diebold et al. [142] and built positive/negative returns and volatility connectedness networks. Furthermore, the regression model is used to identify drivers of various cryptocurrency integration levels. Further analysis revealed that the relationship between each cryptocurrency in terms of return and volatility is not necessarily due to its market size. Adjepong et al. [339] explored market coherence and volatility causal linkages of seven leading cryptocurrencies. Wavelet-based methods are used to examine market connectedness. Parametric and nonparametric tests are employed to investigate directions of volatility spillovers of the assets. Experiments revealed from diversification benefits to linkages of connectedness and volatility in cryptocurrency markets. Elie et al. [70] found the presence of jumps was detected in a series of 12 cryptocurrency returns, and significant jumping activity was found in all cases. More results underscore the importance of the jump in trading volume for the formation of cryptocurrency leapfrogging. Stanislaw et al. [146] examined the correlation of daily exchange rate fluctuations within a basket of the 100 highest market capitalization cryptocurrencies for the period October 1, 2015 to March 31, 2019. The corresponding dynamics mainly involve one of the leading eigenvalues of the correlation matrix, while the others are mainly consistent with the eigenvalues of the Wishart random matrix. The study shows that Bitcoin (BTC) was dominant during the period under consideration, signalling exchange rate dynamics at least as influential as the US dollar (USD).

Some researchers explored the relationship between cryptocurrency and different factors, including futures, gold, etc. Hale et al. [203] suggested that Bitcoin prices rise and fall rapidly after CME issues futures consistent with pricing dynamics. Specifically, the
authors pointed out that the rapid rise and subsequent decline in prices after the introduction of futures is consistent with trading behaviour in the cryptocurrency market. Werner et al. [265] focused on the asymmetric interrelationships between major currencies and cryptocurrencies. The results of multiple fractal asymmetric de-trending cross-correlation analysis show evidence of significant persistence and asymmetric multiplicity in the cross-correlation between most cryptocurrency pairs and ETF pairs. Bai et al. [29] studied a trading algorithm for foreign exchange on a cryptocurrency Market using the Automated Triangular Arbitrage method. Implementing a pricing strategy, implementing trading algorithms and developing a given trading simulation are three problems solved by this research. Kang et al. [239] examined the hedging and diversification properties of gold futures versus Bitcoin prices by using dynamic conditional correlations (DCCs) and wavelet coherence. DCC-GARCH model [153] is used to estimate the time-varying correlation between Bitcoin and gold futures by modeling the variance and the co-variance but also this two flexibility. Wavelet coherence method focused more on co-movement between Bitcoin and gold futures. From experiments, the wavelet coherence results indicated volatility persistence, causality and phase difference between Bitcoin and gold. Qiao et al [353] used wavelet coherence and relevance networks to investigate synergistic motion between Bitcoin and other cryptocurrencies. The authors then tested the hedging effect of bitcoin on others at different time frequencies by risk reduction and downside risk reduction. The empirical results provide evidence of linkage and hedging effects. Bitcoin’s returns and volatility are ahead of other cryptocurrencies at low frequencies from the analysis, and in the long run, Bitcoin has a more pronounced hedging effect on other cryptocurrencies. Dyhrberg et al [148] applied the GARCH model and the exponential GARCH model in analysing similarities between Bitcoin, gold and the US dollar. The experiments showed that Bitcoin, gold and the US dollar have similarities with the variables of the GARCH model, have similar hedging capabilities and react symmetrically to good and bad news. The authors observed that Bitcoin can combine some advantages of commodities and currencies in financial markets to be a tool for portfolio management. Alexander et al. [15] used a set of model-free smile-implied and other smile-adjusted deltas to analyze robust dynamic delta hedging for bitcoin options. An analysis of unique data on hourly historical bitcoin option prices showed that the bitcoin implied volatility curve behaves very differently from the curve of stock index options. In certain periods, the use of smiling implied hedge ratios could significantly outperform simple Black-Scholes delta hedging, especially when using perpetual swaps as a hedging instrument, where out-of-the-money puts can have efficiency gains in excess of 30%.

Baur et al. [38] extended the research of Dyhrberg et al.; the same data and sample periods are tested [148] with GARCH and EGARCH-(1,1) models but the experiments reached different conclusions. Baur et al. found that Bitcoin has unique risk-return
2.8 Portfolio, Assets and Market Condition

characteristics compared with other assets. They noticed that Bitcoin excess returns and volatility resemble a rather highly speculative asset with respect to gold or the US dollar. Bouri et al. [67] studied the relationship between Bitcoin and energy commodities by applying DCCs and GARCH (1,1) models. In particular, the results showed that Bitcoin is a strong hedge and safe haven for energy commodities. Kakushadze [234] proposed factor models for the cross-section of daily cryptoasset returns and provided source code for data downloads, computing risk factors and backtesting for all cryptocurrencies and a host of various other digital assets. The results showed that cross-sectional statistical arbitrage trading may be possible for cryptoassets subject to efficient executions and shorting. Beneki et al. [45] tested hedging abilities between Bitcoin and Ethereum by a multivariate BEKK-GARCH methodology and impulse response analysis within VAR model. The results indicated a volatility transaction from Ethereum to Bitcoin, which implied possible profitable trading strategies on the cryptocurrency derivatives market. Guglielmo et al. [90] examined the week effect in cryptocurrency markets and explored the feasibility of this indicator in trading practice. Student $t$-test, ANOVA, Kruskal–Wallis and Mann–Whitney tests were carried out for cryptocurrency data in order to compare time periods that may be characterised by anomalies with other time periods. When an anomaly is detected, an algorithm was established to exploit profit opportunities (MetaTrader terminal in MQL4 is mentioned in this research). The results showed evidence of anomaly (abnormal positive returns on Mondays) in the Bitcoin market by backtesting in 2013-2016.

A number of special research methods have proven to be relevant to cryptocurrency pairs, which is reflected in cryptocurrency trading. Delfabbro et al. [140] pointed out that cryprocurrency trading have similarities to gambling. Decisions are often based on limited information, short-term profit motives, and highly volatile and uncertain outcomes. The authors examined whether gambling and problem gambling are reliable predictors of reported cryptocurrency trading strength. Results showed that problem gambling scores (PGSI) and engaging in stock trading were significantly correlated with measures of cryptocurrency trading intensity based on time spent per day, number of trades and level of expenditure. In further research, Delfabbro et al. [141] reviewed the specific structural features of cryptocurrency trading and its potential to give rise to excessive or harmful behaviour, including over-spending and compulsive checking. There are some similarities noted between online sports betting and day trading, but there are also some important differences. These include the 24/7 nature of trading, the global nature of the market and the powerful role of social media, social influences and non-balance sheet related events as determinants of price movement. Cheng et al. [108] investigated whether the economic policy uncertainty (EPU) index provided by Baker et al. [30] can predict the returns of cryptocurrencies. The results suggest that China’s EPU Index can predict monthly returns for Bitcoin, whereas the EPU Index for the US or other Asian countries has no
predictive power. In addition, China’s ban on cryptocurrency trading only affects bitcoin returns among major cryptocurrencies. Leirvik [277] analysed the relationship between the particular volatility of market liquidity and the returns of the five largest cryptocurrencies by market capitalisation. The results showed that in general there is a positive correlation between the volatility of liquidity and the returns of large-cap cryptocurrencies. For the most liquid and popular cryptocurrencies, this effect does not exist: Bitcoin. Moreover, the liquidity of cryptocurrencies increases over time, but varies greatly over time.

2.8.2 Crypto-asset Portfolio Research

Some researchers applied portfolio theory to crypto assets. Corbet et al. [132] gave a systematic analysis of cryptocurrencies as financial assets. Brauneis et al. [75] applied the Markowitz mean-variance framework in order to assess the risk-return benefits of cryptocurrency portfolios. In an out-of-sample analysis accounting for transaction cost, they found that combining cryptocurrencies enriches the set of ‘low’-risk cryptocurrency investment opportunities. In terms of the Sharpe ratio and certainty equivalent returns, the $1/N$-portfolio (i.e., “naive” strategies, such as equally dividing amongst asset classes) outperformed single cryptocurrencies and more than 75% in terms of the Sharpe ratio and certainty equivalent returns of mean-variance optimal portfolios. Castro et al. [93] developed a portfolio optimisation model based on the Omega measure which is more comprehensive than the Markowitz model and applied this to four crypto-asset investment portfolios by means of a numerical application. Experiments showed crypto-assets improves the return of the portfolios, but on the other hand, also increase the risk exposure. Alexander et al. [13] evaluated the forecasting accuracy of a simple parametric RiskMetrics TM-type volatility and covariance model, focusing on ad hoc parameter selection rather than a data-intensive calibration procedure. The results showed that simpler models in the exponentially weighted moving average (EWMA) class are as accurate as the VaR and ES predicted GARCH models, provided they capture asymmetric volatility responses and heavy-tailed return distributions.

Bedi et al. [42] examined diversification capabilities of Bitcoin for a global portfolio spread across six asset classes from the standpoint of investors dealing in five major fiat currencies, namely US Dollar, Great Britain Pound, Euro, Japanese Yen and Chinese Yuan. They employed modified Conditional Value-at-Risk and standard deviation as measures of risk to perform portfolio optimisations across three asset allocation strategies and provided insights into the sharp disparity in Bitcoin trading volumes across national currencies from a portfolio theory perspective. Similar research has been done by Antipova et al. [18], which explored the possibility of establishing and optimizing a global portfolio by diversifying investments using one or more cryptocurrencies, and assessing returns to
investors in terms of risks and returns. Fantazzini et al. [163] proposed a set of models that can be used to estimate the market risk for a portfolio of crypto-currencies, and simultaneously estimate their credit risk using the Zero Price Probability (ZPP) model. The results revealed the superiority of the t-copula/skewed-t GARCH model for market risk, and the ZPP-based models for credit risk. Qiang et al. [225] examined the common dynamics of bitcoin exchanges. Using a connectivity metric based on the actual daily volatility of the bitcoin price, they found that Coinbase is undoubtedly the market leader, while Binance performance is surprisingly weak. The results also suggested that safer asset extraction is more important for volatility linkages between Bitcoin exchanges relative to trading volumes. Fasanya et al. [165] quantified returns and volatility transmission between cryptocurrency portfolios by using a spillover approach and rolling sample analysis. The results showed that there is a significant difference between the behaviour of cryptocurrency portfolio returns and the volatility spillover index over time. Given the spillover index, the authors found evidence of interdependence between cryptocurrency portfolios, with the spillover index showing an increased degree of integration between cryptocurrency portfolios.

Trucios et al. [405] proposed a methodology based on vine copulas and robust volatility models to estimate the Value-at-Risk (VaR) and Expected Shortfall (ES) of cryptocurrency portfolios. The proposed algorithm displayed good performance in estimating both VaR and ES. Hrytsiuk et al. [213] showed that the cryptocurrency returns can be described by the Cauchy distribution and obtained the analytical expressions for VaR risk measures and performed calculations accordingly. As a result of the optimisation, the sets of optimal cryptocurrency portfolios were built in their experiments.

Jiang et al. [227] proposed a two-hidden-layer CNN that takes the historical price of a group of cryptocurrency assets as an input and outputs the weight of the group of cryptocurrency assets. This research focused on portfolio research in cryptocurrency assets using emerging technologies like CNN. Training is conducted in an intensive manner to maximise cumulative returns, which is considered a reward function of the CNN network. The performance of the CNN strategy is compared with the three benchmarks and the other three portfolio management algorithms (buy and hold strategy, Uniform Constant Rebalanced Portfolio and Universal Portfolio with Online Newton Step and Passive Aggressive Mean Reversion); the results are positive in that the model is only second to the Passive Aggressive Mean Reversion algorithm (PAMR). Estalayo et al. [158] reported initial findings around the combination of DL models and Multi-Objective Evolutionary Algorithms (MOEAs) for allocating cryptocurrency portfolios. Technical rationale and details were given on the design of a stacked DL recurrent neural network, and how its predictive power can be exploited for yielding accurate ex-ante estimates of the return and risk of the portfolio. Results obtained for a set of experiments carried out with real
2.8 Portfolio, Assets and Market Condition

cryptocurrency data have verified the superior performance of their designed deep learning model with respect to other regression techniques.

2.8.3 Bubbles and Crash Analysis

Bubbles and crash analysis is an important researching area in cryptocurrency trading. Phillips and Yu proposed a methodology to test for the presence of cryptocurrency bubble [110], which is extended by Shaen et al. [133]. The method is based on supremum Augmented Dickey–Fuller (SADF) to test for the bubble through the inclusion of a sequence of forwarding recursive right-tailed ADF unit root tests. An extended methodology generalised SADF (GSAFD), is also tested for bubbles within cryptocurrency data. The research concluded that there is no clear evidence of a persistent bubble in cryptocurrency markets including Bitcoin or Ethereum. Bouri et al. [71] date-stamped price explosiveness in seven large cryptocurrencies and revealed evidence of multiple periods of explosivity in all cases. GSADF is used to identify multiple explosiveness periods and logistic regression is employed to uncover evidence of co-explosivity across cryptocurrencies. The results showed that the likelihood of explosive periods in one cryptocurrency generally depends on the presence of explosivity in other cryptocurrencies and points toward a contemporaneous co-explosivity that does not necessarily depend on the size of each cryptocurrency.

Extended research by Phillips et al. [346, 347] (who applied a recursive augmented Dickey-Fuller algorithm, which is called PSY test) and Landsnes et al. [155] studied possible predictors of bubble periods of certain cryptocurrencies. The evaluation includes multiple bubble periods in all cryptocurrencies. The result shows that higher volatility and trading volume is positively associated with the presence of bubbles across cryptocurrencies. In terms of bubble prediction, the authors found the probit model to perform better than the linear models.

Phillips et al. [348] used Hidden Markov Model (HMM) and Superiority and Inferiority Ranking (SIR) method to identify bubble-like behaviour in cryptocurrency time series. Considering HMM and SIR method, an epidemic detection mechanism is used in social media to predict cryptocurrency price bubbles, which classify bubbles through epidemic and non-epidemic labels. Experiments have demonstrated a strong relationship between Reddit usage and cryptocurrency prices. This work also provides some empirical evidence that bubbles mirror the social epidemic-like spread of an investment idea. Guglielmo et al. [89] examined the price overreactions in the case of cryptocurrency trading. Some parametric and non-parametric tests confirmed the presence of price patterns after overreactions, which identified that the next-day price changes in both directions are bigger than after “normal” days. The results also showed that the overreaction detected in the cryptocurrency market would not give available profit opportunities (possibly due to transaction costs)
that cannot be considered as evidence of the EMH. Chaim et al. [100] analysed the high unconditional volatility of cryptocurrency from a standard log-normal stochastic volatility model to discontinuous jumps of volatility and returns. The experiment indicated the importance of incorporating permanent jumps to volatility in cryptocurrency markets.

J.L. Cross et al. [136] investigated the existence and nature of the interdependence of bitcoin, ethereum, litecoin and ripple during the cryptocurrency bubble of 2017-18. A generalized time-varying asset pricing model approach is proposed. The results showed that the negative news impact of the boom period in 2017 for LiteCoin and Ripple, which incurred a risk premium for investors, could explain the returns of cryptocurrencies during the 2018 crash. Martin et al. [415] presented an important extension to Conditional Value at Risk (CoVaR), a popular measure of systematic risk, and investigate its properties in the cryptocurrency market. VCoVaR relaxed the assumption of normality and is estimated by copula, analysing how different adverse events in cryptocurrencies affect each other’s risk levels. The results showed that Litecoin has the largest impact on Bitcoin.

2.8.4 Extreme condition

Differently from traditional fiat currencies, cryptocurrencies are risky and exhibit heavier tail behaviour. Paraskevi et al. [244] found extreme dependence between returns and trading volumes. Evidence of asymmetric return-volume relationship in the cryptocurrency market was also found by the experiment, as a result of discrepancies in the correlation between positive and negative return exceedances across all the cryptocurrencies.

There has been a price crash in late 2017 to early 2018 in cryptocurrency [425]. Yaya et al. [425] researched the persistence and dependence of Bitcoin on other popular alternative coins before and after the 2017/18 crash in cryptocurrency markets. The result showed that higher persistence of shocks is expected after the crash due to speculations in the mind of cryptocurrency traders, and more evidence of non-mean reversions, implying chances of further price fall in cryptocurrencies.

Manahov [306] obtained millisecond data for major cryptocurrencies as well as the cryptocurrency indices Cryptocurrency IndeX (CRIX) and Cryptocurrencies Index 30 (CCI30) to investigate the relationship between cryptocurrency liquidity, herding behaviour and profitability during extreme price movements (EPM). Millisecond data was obtained for major cryptocurrencies as well as the cryptocurrency indices CRIX and CCI30 to investigate the relationship between cryptocurrency liquidity, herding behaviour and profitability during EPM. The experiments demonstrate that cryptocurrency traders (CTs) can promote EPM and demand liquidity even during periods of maximum EPM. The authors’ robustness checks suggest that herding behaviour follows a dynamic pattern with decreasing magnitude over time. Shahzad et al. [369] investigated the interdependence of
median-based and tail-based returns between cryptocurrencies under normal and extreme market conditions. The experiment used daily data and combines LASSO techniques with quantile regression within a network analysis framework. The main results showed that the interdependence of the tails is higher than the median, especially in the right tail. Fluctuations in market, size and momentum drive return connectivity and clustering coefficients under both normal and extreme market conditions. Chan et al. [102] examined the extreme dependence and correlation between high-frequency cryptocurrency (Bitcoin and Ethereum, relative to the euro and the US dollar) returns and trading volumes in the extreme tails associated with booms and busts in cryptocurrency markets. Experiments with extreme value theory methods highlight how these results can help traders and practitioners who rely on technical indicators in their trading strategies - especially in times of extreme market volatility or irrational market booms.

2.9 Cryptocurrency asset pricing

2.9.1 Factor pricing model

In traditional financial markets, there is a school of literature that has developed well-known factors such as the ‘small - large’ (SMB) factor, the ‘high - low’ (HML) value and the ‘strong - weak’ (RMW) profitability investment model factor [159]. Many factor model research papers have been published in the cryptocurrency trading area. Shen et al. [374] proposed a three-factor pricing model, containing value-weight return, size and reversal factors in constructing a three-factor model. The results show that the model strongly outperforms the cryptocurrency-CAPM model and that its performance is robust to different factor constructs. Shahzad et al [370] calculated zero-cost portfolios by buying winners and selling losers, market factors or more traditional market risk premiums, and formed six value-weighted portfolios based on prior single-day returns and market values. Liu et al. [289] found that three factors - cryptocurrency market, size and momentum captured the cross-sectional expected returns of cryptocurrencies. For the factor premium, the authors provide evidence that the cryptocurrency size factor is related to the liquidity effect and that the size premium is consistent with the trade-off between capital gains and convenience gains. Liu et al [286] presented three common factors specific to the cryptocurrency market, namely market factors and factors related to size and momentum, to examine the average returns of cryptocurrencies. The authors used cross-sectional analysis methods for the anomalous variables and time series regression methods for the risk factors. The results also showed that the scale and momentum effects are strong in the cryptocurrency market. Cakici et al. [85] used a series of machine learning models to explore the predictability of cross-sectional returns in the cryptocurrency market.
2.9 Cryptocurrency asset pricing

The predictability of returns were mainly derived from simple characteristics such as exceptional volatility, past alpha or maximum daily returns, and is likely to be driven by mispricing. Abnormal returns were mainly derived from short positions, concentrated in assets that are difficult to arbitrage and decline over time with profits heavily dependent on shorting small cryptocurrencies. Biais et al. [46] provided a general equilibrium analysis of cryptocurrency pricing, where in addition to the fundamentals, i.e. the net trading revenue stream, the equilibrium price reflects sunspots, which in turn implies multiple equilibria and extrinsic volatility. The authors argued that cryptocurrency prices will fluctuate even if fundamentals remain unchanged.

2.9.2 Valuation in pricing and market network

The value of cryptocurrencies such as Bitcoin, Ethereum, and Litecoin can be highly volatile, meaning they can fluctuate rapidly in response to market forces such as investor sentiment, news events, and regulatory changes. Factors that can affect the value of cryptocurrencies include adoption by mainstream businesses, government regulations, mining difficulty, security concerns, and overall investor confidence in the technology and its potential future use cases. The value of cryptocurrency is priced by many factors, including information disclosure and data on the chain. Liu et al [288] examined the role of information disclosed on the blockchain in the cryptocurrency market and find that blockchain user adoption disclosure, measured by the number of new addresses, is highly correlated with value in the cryptocurrency market. Cong et al. [124] developed a dynamic asset pricing model for cryptocurrencies/tokens. The equilibrium price of the token was determined by aggregating the transaction demands of different users, rather than discounting the cash flows as in standard valuation models. By comparing platforms with and without tokens, the authors showed that the introduction of tokens reduces the effective cost of ownership for conducting platform transactions, thereby accelerating the adoption of production platforms. Zhang [433] provided a critical review of pricing issues related to cryptocurrencies and NFT from a marketing perspective. The author showed that the value of cryptocurrencies like Bitcoin depends on the market’s collective belief in its long-term viability as a store of value and unit of exchange, a belief that may lead to excessive pricing equilibrium for Bitcoin and is also influenced by many demand factors. Barth et al [36] investigated the extent to which ethical considerations associated with the use of cryptocurrencies affect the valuation of such currencies. The experiment involved measuring the extent to which ethical and unethical terms were used in Bitcoin-related discussions on Twitter, and the results showed that the frequency of unethical discussions of Bitcoin was negatively correlated with its price. Liu et al. [290] proposed a new market-to-fundamental ratio, the
price-to-utility (PU) ratio, using a unique blockchain accounting approach as a cryptocurrency asset fundamental indicator. Experimentally validated PU ratio valuation through unsupervised and supervised machine learning, valuation methods can inform investment returns and effectively predict bull markets. The authors proposed an automated trading strategy suggested by the PU ratio that outperforms traditional buy-and-hold and market timing strategies.

### 2.9.3 Risk Factors

The risk factor refers to any element or circumstance that may potentially result in loss or harm to an investor or user. Borri et al. [64] studied the risk premium of cryptocurrencies and characterize the stochastic discount factor according to potential factors to obtain risk premium estimates for a large number of observable factors that are robust to omitted variables and measurement errors. The authors found that macroeconomic risk is priced into cryptocurrencies, specifically in the case of higher geopolitical risk or negative macro shocks, where coins with lower returns are riskier and have higher returns. Alexander et al. [14] obtained a unique dataset of high frequency trading prices for bitcoin calls and puts from the Deribit cryptocurrency derivatives exchange and use these prices to construct the term structure of the bitcoin implied volatility index. These indices provided indicative fair values for bitcoin variance swaps traded on-chain, with realized volatilities monitored at 15-minute, hourly and daily frequencies to check bitcoin variance risk premiums at different maturities. Mamun et al [9] investigated the impact of geopolitical risk, global and US economic policy uncertainty on bitcoin investments. The authors found that geopolitical risk and uncertainty in global economic policy require a risk premium. Furthermore, in times of high policy uncertainty and deteriorating economic conditions, bitcoin investors could only hedge their portfolios with gold and not with other financial assets. Shi [375] conducted an empirical analysis of the Bitcoin futures risk premium. The experimental findings showed that speculators’ and retailers’ trading activity and extreme sentiment have significant predictive power for subsequent changes in bitcoin futures prices over different time horizons. Moreover, the return impact of changes in hedgers’ net positions is likely to be influenced by extremes in macroeconomic variables.

### 2.9.4 Advantages and disadvantages among pricing factors

Pricing factors in investing or trading refer to the various elements that affect the price or value of a financial asset like cryptocurrency. These factors can be internal or external to the asset or market, and they can change over time. By analyzing pricing factors, investors and traders can identify trends, risks, and opportunities in the market and adjust their
strategies accordingly. But when designing trading strategies according to asset pricing factors, we observe advantages and disadvantages in applying them.

There are advantages to designing trading strategies based on pricing factors. Firstly, pricing-factors-based strategies could increase accuracy. Pricing factors provide traders with reliable and accurate data that they can use to make informed decisions. By analyzing these factors, traders can identify patterns and trends in the market, which can help them predict future price movements with greater accuracy. Secondly, pricing-factors-based strategies could improve profitability. Trading strategies based on pricing factors can help traders maximize their profits by enabling them to enter and exit positions at the most opportune times. By taking advantage of price movements in the market, traders can increase their profits and reduce their losses. Thirdly, pricing-factors-based strategies could reduce risk. By using pricing factors to inform their trading decisions, traders can reduce their risk exposure. By avoiding trades that have a higher likelihood of resulting in losses, traders can protect their capital and minimize their risk of suffering significant losses. Fourthly, pricing-factors-based strategies could increase efficiency. Trading strategies based on pricing factors can help traders make more efficient use of their time and resources. By automating certain aspects of their trading process, traders can focus on other important tasks, such as research and analysis, which can help them make better trading decisions over the long term. Fifthly, pricing-factors-based strategies could increase versatility. Pricing factors can be used to design trading strategies for a wide range of cryptocurrency markets and assets. Whether a trader is interested in Bitcoin, Ethereum, or any other cryptocurrency, pricing factors can be used to analyze market trends and develop effective trading strategies that work across different markets and assets.

There are several disadvantages to designing trading strategies based solely on pricing factors in cryptocurrency trading. Firstly, cryptocurrencies are highly volatile, and their prices can fluctuate significantly within a short period. This makes it difficult to predict future prices accurately, which can lead to losses if trades are not timed correctly. Secondly, cryptocurrency markets are not as regulated as traditional financial markets, making them vulnerable to manipulation and insider trading. This can lead to sudden price changes that are not based on market fundamentals, making it challenging to design effective trading strategies. Thirdly, cryptocurrencies are a relatively new asset class, and there is limited historical data available for analysis. This makes it difficult to identify patterns and trends that can be used to inform trading decisions. Fourthly, cryptocurrencies are subject to market sentiment and hype, which can cause prices to rise or fall rapidly based on rumours or news. This can make it difficult to predict future price movements accurately. Fifthly, some cryptocurrencies have low trading volumes and liquidity, making it difficult to enter and exit trades quickly. This can lead to slippage and other execution issues, which can impact trading strategies’ effectiveness.
2.10 Other work related to Cryptocurrency Trading

Some other research papers related to cryptocurrency trading treat market behaviour, regulatory mechanisms and benchmarks.

Krafft et al. [264] and Yang [423] analysed market dynamics and behavioural anomalies respectively to understand effects of market behaviour in the cryptocurrency market. Krafft et al. discussed potential ultimate causes, potential behavioural mechanisms and potential moderating contextual factors to enumerate possible influence of GUI and API on cryptocurrency markets. Then they highlighted the potential social and economic impact of human-computer interaction in digital agency design. Yang, on the other hand, applied behavioural theories of asset pricing anomalies in testing 20 market anomalies using cryptocurrency trading data. The results showed that anomaly research focused more on the role of speculators, which gave a new idea to research the momentum and reversal in the cryptocurrency market. Cocco et al. [118] implemented a mechanism to form a Bitcoin price and specific behaviour for each type of trader including the initial wealth distribution following Pareto’s law, order-based transaction and price settlement mechanism. Specifically, the model reproduced the unit root attributes of the price series, the fat tail phenomenon, the volatility clustering of price returns, the generation of Bitcoins, hashing power and power consumption. Vladimir et al. [211] proposed a discrete price model based on a mixture of a double Poisson distribution with dynamic volatility and a dynamic proportion of agent types and estimated by the maximum likelihood method. Experiments found that higher instantaneous volatility leads to weaker price aggregation at UHF.

Leclair [275] and Vidal-Thomás et al. [410] analysed the existence of herding in the cryptocurrency market. Leclair applied herding methods of Huang and Salmon [219] in estimating the market herd dynamics in the CAPM framework. Vidal-Thomás et al. analyse the existence of herds in the cryptocurrency market by returning the cross-sectional standard (absolute) deviations. Both their findings showed significant evidence of market herding in the cryptocurrency market. Makarov et al. [302] studied price impact and arbitrage dynamics in the cryptocurrency market and found that 85% of the variations in bitcoin returns and the idiosyncratic components of order flow [284] play an important role in explaining the size of the arbitrage spreads between exchanges. King et al. [254] examined the extent to which herding and feedback trading behaviour drive the price dynamics of nine major cryptocurrencies. The study documented heterogeneity in the types of feedback trading strategies used by investors in different markets and evidence of herding or “trend chasing” behaviour in some cryptocurrency markets.

In November 2019, Griffin et al. put forward a paper on the thesis of unsupported digital money inflating cryptocurrency prices [195], which caused a great stir in the
academic circle and public opinion. Using algorithms to analyse Blockchain data, they found that purchases with Tether are timed following market downturns and result in sizeable increases in Bitcoin prices. By mapping the blockchains of Bitcoin and Tether, they were able to establish that one large player on Bitfinex uses Tether to purchase large amounts of Bitcoin when prices are falling and following the prod of Tether.

More researches involved benchmark and development in cryptocurrency market [208, 436], regulatory framework analysis [371, 166], data mining technology in cryptocurrency trading [342], application of efficient market hypothesis in the cryptocurrency market [379], Decentralized Exchanges (DEXs) and artificial financial markets for studying a cryptocurrency market [117]. Hileman et al. [208] segmented the cryptocurrency industry into four key sectors: exchanges, wallets, payments and mining. They gave a benchmarking study of individuals, data, regulation, compliance practices, costs of firms and a global map of mining in the cryptocurrency market in 2017. Zhou et al. [436] discussed the status and future of computer trading in the largest group of Asia-Pacific economies and then considered algorithmic and high frequency trading in cryptocurrency markets as well. Shanaev et al. [371] used data on 120 regulatory events to study the implications of cryptocurrency regulation and the results showed that stricter regulation of cryptocurrency is not desirable. Feinstein et al. [166] collected raw data on global cryptocurrency regulations and used them to empirically test the trading activity of many exchanges against key regulatory announcements. No systematic evidence has been found that regulatory measures cause traders to flee or enter the affected regional jurisdictions according to authors’ analysis. Akhilesh et al. [342] used the average absolute error calculated between the actual and predicted values of the market sentiment of different cryptocurrencies on that day as a method for quantifying the uncertainty. They used the comparison of uncertainty quantification methods and opinion mining to analyse current market conditions. Sigaki et al. [379] used permutation entropy and statistical complexity on the sliding time window returned by the price log to quantify the dynamic efficiency of more than four hundred cryptocurrencies. As a result, the cryptocurrency market showed significant compliance with efficient market assumptions. Aspris et al. [22] surveyed the rapid rise of DEXs, including automated market makers. The study demonstrated the significant differences in the listing and trading characteristics of these tokens compared to their centralised equivalents. Cocco et al. [117] described an agent-based artificial cryptocurrency market in which heterogeneous agents buy or sell cryptocurrencies. The proposed simulator is able to reproduce some real statistical properties of price returns observed in the Bitcoin real market. Marko [338] considered the future use of cryptocurrencies as money based on the long-term value of cryptocurrencies. Neil et al. [184] analysed the influence of network effect on the competition of new cryptocurrency markets. Bariviera and Merediz-Sola [34] gave a survey based on hybrid analysis, which proposed...
2.11 Summary Analysis of Literature Review

A methodological hybrid method for a comprehensive literature review and provided the latest technology in the cryptocurrency economics literature.

There also exists some research and papers introducing the basic process and rules of cryptocurrency trading including findings of Hansel et al. [204], Kate [241], Garza et al. [186], Ward et al. [419] and Fantazzini et al. [162]. Hansel et al. [204] introduced the basics of cryptocurrency, Bitcoin and Blockchain, ways to identify the profitable trends in the market, ways to use Altcoin trading platforms such as GDAX and Coinbase, methods of using a crypto wallet to store and protect the coins in their book. Kate et al. [241] set six steps to show how to start an investment without any technical skills in the cryptocurrency market. This book is an entry-level trading manual for starters learning cryptocurrency trading. Garza et al. [186] simulated an automatic cryptocurrency trading system, which helps investors limit systemic risks and improve market returns. This paper is an example to start designing an automatic cryptocurrency trading system. Ward et al. [419] discussed algorithmic cryptocurrency trading using several general algorithms, and modifications thereof including adjusting the parameters used in each strategy, as well as mixing multiple strategies or dynamically changing between strategies. This paper is an example to start algorithmic trading in cryptocurrency market. Fantazzini et al. [162] introduced the R packages Bitcoin-Finance and bubble, including financial analysis of cryptocurrency markets including Bitcoin.

A community resource, that is, a platform for scholarly communication, about cryptocurrencies and Blockchains is “Blockchain research network”, see [358].

2.11 Summary Analysis of Literature Review

This section analyses the timeline, the research distribution among technology and methods, the research distribution among properties. It also summarises the datasets that have been used in cryptocurrency trading research.

2.11.1 Timeline

Figure 2.7 shows several major events in cryptocurrency trading. The timeline contains milestone events in cryptocurrency trading and important scientific breakthroughs in this area.

As early as 2009, Satoshi Nakamoto proposed and invented the first decentralised cryptocurrency, Bitcoin [327]. It is considered to be the start of cryptocurrency. In 2010, the first cryptocurrency exchange was founded, which means cryptocurrency would not be an OTC market but traded on exchanges based on an auction market system.
2.11 Summary Analysis of Literature Review

In 2013, Kristoufek [266] concluded that there is a strong correlation between Bitcoin price and the frequency of “Bitcoin” search queries in Google Trends and Wikipedia. In 2014, Lee and Yang [276] firstly proposed to check causality from copula-based causality in the quantile method from trading volumes of seven major cryptocurrencies to returns and volatility.

In 2015, Cheah et al. [106] discussed the bubble and speculation of Bitcoin and cryptocurrencies. In 2016, Dyhrberg explored Bitcoin volatility using GARCH models combined with gold and US dollars [148].

From late 2016 to 2017, machine learning and deep learning technology were applied in the prediction of cryptocurrency return. In 2016, McNally et al. [315] predicted Bitcoin price using the LSTM algorithm. Bell and Zbikowski et al. [43, 427] applied SVM algorithm to predict trends of cryptocurrency price. In 2017, Jiang et al. [227] used double Q-network and pre-trained it using DBM for the prediction of cryptocurrencies portfolio weights.

From 2019 to 2020, several research directions including cross asset portfolios [42, 93, 75], transaction network applications [268, 71], machine learning optimisation [355, 23, 435] have been considered in the cryptocurrency trading area.

In 2021, more regulation issues were put out the stage. On 18 May 2021, China banned financial institutions and payment companies from providing services related to cryptocurrency transactions, which led to a sharp drop in the price of bitcoin [359]. In June 2021, El Salvador becomes the first country to accept Bitcoin as legal tender [317].

2.11.2 Research Distribution among Properties

We counted the number of papers covering different aspects of cryptocurrency trading. Figure 2.8 shows the result. The attributes in the legend are ranked according to the number of papers that specifically test the attribute.
2.11 Summary Analysis of Literature Review

Over one-third (37.67%) of the papers research prediction of returns. Another one-third of papers focus on researching bubbles and extreme conditions and the relationship between pairs and portfolios in cryptocurrency trading. The remaining researching topics (prediction of volatility, trading system, technical trading and others) have roughly one-third share.

![Fig. 2.8 Research distribution among cryptocurrency trading properties](image)

2.11.3 Research Distribution among Categories and Technologies

This section introduces and compares categories and technologies in cryptocurrency trading. When papers cover multiple technologies or compare different methods, we draw statistics from different technical perspectives.

Among all the 177 papers, 102 papers (69.86%) cover statistical methods and machine learning categories. These papers basically research technical-level cryptocurrency trading including mathematical modeling and statistics. Other papers related to trading systems on pure technical indicators and introducing the industry and its history are not included in this analysis. Among all 102 papers, 88 papers (86.28%) present statistical methods and technologies in cryptocurrency trading research and 13.72% papers research machine learning applied to cryptocurrency trading (cf. Figure 2.9). It is interesting to mention that, there are 17 papers (16.67%) applying and comparing more than one technique in cryptocurrency trading. More specifically, Bach et al. [27], Alessandretti et al. [11], Vo et
al. [412], Phaladisailoed et al. [345], Siaminos [378], Rane et al. [355] used both statistical methods and machine learning methods in cryptocurrency trading.

Table 2.8 shows the results of search hits in all trading areas (not limited to cryptocurrencies). From the table, we can see that most research findings focused on statistical methods in trading, which means most of the research on traditional markets still focused on using statistical methods for trading. But we observed that machine learning in trading had a higher degree of attention. It might because the traditional technical and fundamental have been arbitraged, so the market has moved in recent years to find new anomalies to exploit. Meanwhile, the results also showed there exist many opportunities for research in the widely studied areas of machine learning applied to trade in cryptocurrency markets (cf. Section 2.12).
Table 2.8 Search hits of research distribution in all trading areas

<table>
<thead>
<tr>
<th>Technology Category</th>
<th>Google Scholar hits</th>
<th>Google hits</th>
<th>Arxiv hits</th>
<th>SCOPUS Indexed citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical methods</td>
<td>1.22M</td>
<td>62M</td>
<td>1240</td>
<td>2790</td>
</tr>
<tr>
<td>Machine learning methods</td>
<td>483K</td>
<td>150M</td>
<td>520</td>
<td>1,754</td>
</tr>
</tbody>
</table>

**Research Distribution among Statistical methods**

As from Figure 2.9, we further classified the papers using statistical methods into 6 categories: (i) basic regression methods; (ii) linear classifiers and clustering; (iii) time-series analysis; (iv) decision trees and probabilistic classifiers; (v) modern portfolios theory; and, (vi) Others.

- **Basic regression methods** include regression methods (Linear Regression), function estimation and CGCD method.  
- **Linear Classifiers and Clustering** include SVM and KNN algorithm.  
- **Time-series analysis** include GARCH model, BEKK model, ARIMA model, Wavelet time-scale method.  
- **Decision Trees and probabilistic classifiers** include Boosting Tree, RF model.  
- **Modern portfolio theory** include Value-at-Risk (VaR) theory, expected-shortfall (ES), Markowitz mean-variance framework.  
- **Others** include industry, market data and research analysis in cryptocurrency market.

The figure shows that basic Regression methods and time-series analysis are the most commonly used methods in this area.

**Research Distribution among Machine Learning Categories**

Papers using machine learning account for 13.7% (c.f Figure 2.9) of the total. We further classified these papers into three categories: (vii) ANNs, (viii) LSTM/RNN/GRUs, and (ix) DL/RL.

The figure also shows that methods based on LSTM, RNN and GRU are the most popular in this subfield.

- **ANNs** contains papers researching ANN applications in cryptocurrency trading such as back propagation (BP) NN.  
- **LSTM/RNN/GRUs** include papers using neural networks that exploit the temporal structure of the data, a technology especially suitable for time series prediction and financial trading.  
- **DL/RL** includes papers using Multilayer Neural Networks and Reinforcement Learning. The difference between ANN and DL is that generally, DL refers to an ANN with multiple hidden layers while ANN refers to simple structure neural network contained input layer, hidden layer (one or multiple), and an output layer.
2.12 Opportunities in Cryptocurrency Trading

2.11.4 Datasets used in Cryptocurrency Trading

Tables 2.9–2.11 show the details for some representative datasets used in cryptocurrency trading research. Table 2.9 shows the market datasets. They mostly include price, trading volume, order-level information, collected from cryptocurrency exchanges. Table 2.10 shows the sentiment-based data. Most of the datasets in this table contain market data and media/Internet data with emotional or statistical labels. Table 2.11 gives two examples of datasets used in the collected papers that are not covered in the first two tables.

The column “Currency” shows the types of cryptocurrencies included; this shows that Bitcoin is the most commonly used currency for cryptocurrency researches. The column “Description” shows a general description and types of datasets. The column “Data Resolution” means latency of the data (e.g., used in the backtest) – this is useful to distinguish between high-frequency trading and low-frequency trading. The column “Time range” shows the time span of datasets used in experiments; this is convenient to distinguish between the current performance in a specific time interval and the long-term effect. We also present how the dataset has been used (i.e., the task), cf. column “Usage”. “Data Sources” gives details on where the data is retrieved from, including cryptocurrency exchanges, aggregated cryptocurrency index and user forums (for sentiment analysis).

Alexander et al. [12] made an investigation of cryptocurrency data as well. They summarised data collected from 152 published and SSRN discussion papers about cryptocurrencies and analysed their data quality. They found that less than half the cryptocurrency papers published since January 2017 employ correct data.

2.12 Opportunities in Cryptocurrency Trading

This section discusses potential opportunities for future research in cryptocurrency trading.

Sentiment-based research. As discussed above, there is a substantial body of work, which uses natural language processing technology, for sentiment analysis with the ultimate goal of using news and media contents to improve the performance of cryptocurrency trading strategies.

Possible research directions may lie in a larger volume of media input (e.g., adding video sources) in sentiment analysis; updating baseline natural language processing model to perform more robust text preprocessing; applying neural networks in label training; extending samples in terms of holding period; transaction-fees; opinion dynamics [429] and, user reputation research.

Long-and-short term trading research. There are significant differences between long and short time horizons in cryptocurrency trading. In long-term trading, investors might obtain greater profits but have more possibilities to control risk when managing a
<table>
<thead>
<tr>
<th>Research Source</th>
<th>Description</th>
<th>Currency</th>
<th>Data Resolution</th>
<th>Time Range</th>
<th>Usage</th>
<th>Data Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bouri et al. [68]</td>
<td>price, volatility, detrended volume data</td>
<td>Bitcoin, Ethereum, 5 other cryptocurrencies</td>
<td>daily</td>
<td>From: 2013/01/01 To: 2017/12/31</td>
<td>Prediction of volatility/return</td>
<td>CoinMarketCap</td>
</tr>
<tr>
<td>Nakano et al. [328]</td>
<td>high frequency price, volume data</td>
<td>Bitcoin</td>
<td>15min</td>
<td>From: 2016/07/31 To: 2018/01/24</td>
<td>Prediction of return</td>
<td>Poloniex</td>
</tr>
<tr>
<td>Bu et al. [82]</td>
<td>three pieces time slice for different research objectives</td>
<td>Bitcoin and seven altcoins</td>
<td>Not mentioned</td>
<td>From: 2016/05/14 To: 2018/01/31</td>
<td>Maximum profit with DRL</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Sun et al. [388]</td>
<td>price, volatility</td>
<td>ETC-USDT, other 12 cryptocurrencies</td>
<td>1 minute, 5 minutes, 30 minutes, one hour, one day</td>
<td>From: August 2017 To: December 2018</td>
<td>Prediction of return</td>
<td>Binance, Bitfinex</td>
</tr>
<tr>
<td>Guo et al. [200]</td>
<td>volatility, order book data</td>
<td>Bitcoin</td>
<td>hourly volatility observations, order book snapshots</td>
<td>From: September 2015 To: April 2017</td>
<td>Prediction of volatility</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Vo et al. [412]</td>
<td>timestamps, the OHLC prices etc.</td>
<td>Bitcoin</td>
<td>1 minute</td>
<td>From: Starting 2015 To: End 2016</td>
<td>Prediction of return</td>
<td>Bitstamp, Btoe, Btcn, Coinbase, Coincheck, and Kraken</td>
</tr>
<tr>
<td>Ross et al. [348]</td>
<td>price</td>
<td>Bitcoin, other 3 cryptocurrencies</td>
<td>daily</td>
<td>From: April 2015 To: September 2016</td>
<td>Predicting bubbles</td>
<td>CryptoCompare</td>
</tr>
<tr>
<td>Yaya et al. [425]</td>
<td>price</td>
<td>Bitcoin, other 12 cryptocurrencies</td>
<td>daily</td>
<td>From: 2015/08/07 To: 2018/11/28</td>
<td>Bubbles and crashes</td>
<td>Coin Metrics</td>
</tr>
<tr>
<td>Brauneis et al. [75]</td>
<td>individual price, trading volume</td>
<td>500 most capitalized Cryptocurrencies</td>
<td>daily</td>
<td>From: 2015/01/01 To: 2017/12/31</td>
<td>Portfolios management</td>
<td>CoinMarketCap</td>
</tr>
<tr>
<td>Gradojevic et al. [192]</td>
<td>price, volatility</td>
<td>BTC/USD, ETH/USD, XRP/USD</td>
<td>1-Minute, 1-Hour, 1-Day, 1-Month</td>
<td>From: 2015/01/01 To: 2020/10/15</td>
<td>Prediction of volatility</td>
<td>CryptoCompare</td>
</tr>
<tr>
<td>Research Source</td>
<td>Description</td>
<td>Currency</td>
<td>Time range</td>
<td>Usage</td>
<td>Data Sources</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
<td>----------</td>
<td>------------</td>
<td>-------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>Kim et al. [253]</td>
<td>Online cryptocurrency communities data and market data</td>
<td>Bitcoin, Ethereum, Ripple</td>
<td>From: December 2013 To: August, 2016 (Bitcoin) From: August 2015 To: August, 2016 (Ethereum) From: Creation To: August, 2016 (Ripple)</td>
<td>Prediction of fluctuation</td>
<td>Each community’s HTML page</td>
<td></td>
</tr>
<tr>
<td>Smtus [383]</td>
<td>Hourly data on price and trading volume and search terms from Google Trends</td>
<td>Bitcoin, Ethereum and their respective pricedrivers</td>
<td>From: 2017/12/01 To: 2018/06/30</td>
<td>Prediction of price</td>
<td>Google Trends, Telegram chat groups</td>
<td></td>
</tr>
<tr>
<td>Lamon et al. [272]</td>
<td>Daily price data and cryptocurrency related news article headlines</td>
<td>Bitcoin, Ethereum, Litecoin</td>
<td>From: 2017/01/01 To: 2017/11/30</td>
<td>Prediction of price</td>
<td>Kaggle, news headline</td>
<td></td>
</tr>
<tr>
<td>Phillips et al. [349]</td>
<td>Price and social media factors from Reddit</td>
<td>Bitcoin, Ethereum, Monero</td>
<td>From: 2010/09/10 To: 2017/05/31 (Bitcoin) Others can reference the paper</td>
<td>Waveletcoherence analysis of price</td>
<td>BraveNewCoin</td>
<td></td>
</tr>
<tr>
<td>Kang et al. [238]</td>
<td>Market data and posts and comments including metadata</td>
<td>Bitcoin</td>
<td>From: 2009/11/22 To: 2018/02/02</td>
<td>Relationships Between Bitcoin Prices and User Groups in Online Community</td>
<td>Bitcoin forum</td>
<td></td>
</tr>
</tbody>
</table>
2.12 Opportunities in Cryptocurrency Trading

Table 2.11 Datasets (3/3): Others

<table>
<thead>
<tr>
<th>Research Source</th>
<th>Description</th>
<th>Time range</th>
<th>Usage</th>
<th>Data Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kurbucz [268, 258]</td>
<td>Raw and preprocessed data of all Bitcoin transactions and daily returns</td>
<td>From: 2016/11/09 To: 2018/02/05</td>
<td>Predicting the price of Bitcoin with transaction network</td>
<td>Bitcoin network dataset [321]</td>
</tr>
<tr>
<td>Bedi et al. [42]</td>
<td>A diversified portfolio including equity, fixedincome, real estate, alternative investments, commodities and money market</td>
<td>From: July 2010 To: December 2018</td>
<td>Cross-currency including cryptocurrency researching portfolios</td>
<td>Portfolio sources [42]</td>
</tr>
</tbody>
</table>

position for weeks or months. It is mandatory to control for risk on long term strategies due to the increase in the holding period, directly proportional to the risk incurred by the trader. On the other hand, the longer the horizon, the higher the risk and the most important the risk control. The shorter the horizon, the higher the cost and the lower the risk, so cost takes over the design of a strategy. In short-term trading, automated algorithmic trading can be applied when holding periods are less than a week. Researchers can differentiate between long-term and short-term trading in cryptocurrency trading by applying wavelet technology analysing bubble regimes [349] and considering price explosiveness [71] hypotheses for short-term and long-term research.

The existing work is mainly about showing the differences between long and short-term cryptocurrency trading. Long-term trading means less time would cost in trend tracing and simple technical indicators in market analysis. Short-term trading can limit overall risk because small positions are used in every transaction. But market noise (interference) and short transaction time might cause some stress in short term trading. It might also be interesting to explore the extraction of trading signals, time series research, application to portfolio management, the relationship between a huge market crash and small price drop, derivative pricing in cryptocurrency market, etc.

**Correlation between cryptocurrency and others.** By the effects of monetary policy and business cycles that are not controlled by the central bank, cryptocurrency is always negatively correlated with overall financial market trends. There have been some studies discussing correlations between cryptocurrencies and other financial markets [239, 93], which can be used to predict the direction of the cryptocurrency market.

Considering the characteristics of cryptocurrency, the correlation between cryptocurrency and other assets still requires further research. Possible breakthroughs might be achieved with principal component analysis, the relationship between cryptocurrency and other currencies in extreme conditions (i.e., financial collapse).

**Bubbles and crash research.** To discuss the high volatility and return of cryptocurrencies, current research has focussed on bubbles of cryptocurrency markets [110], correlation between cryptocurrency bubbles and indicators like volatility index (VIX) [155] (which is a “panic index” to measure the implied volatility of S&amp;P500 Index Options), spillover effects in cryptocurrency market [299].
2.13 Conclusions

Additional research for bubbles and crashes in cryptocurrency trading could include a connection between the process of bubble generation and financial collapse and conducting a coherent analysis (coherent process analysis from the formation of bubbles to aftermath analysis of bubble burst), analysis of bubble theory by Microeconomics, trying other physical or industrial models in analysing bubbles in cryptocurrency market (i.e., Omori law [420]), discussing the supply and demand relationship of cryptocurrency in bubble analysis (like using supply and demand graph to simulate the generation of bubbles and simulate the bubble burst).

**Game theory and agent-based analysis.** Applying game theory or agent-based modelling in trading is a hot research direction in the traditional financial market. It might also be interesting to apply this method to trading in cryptocurrency markets.

**Public nature of Blockchain technology.** Investigations on the connections between the formation of a given currency’s transaction network and its price has increased rapidly in recent years; the growing attention on user identification [230] also strongly supports this direction. With an in-depth understanding of these networks, we may identify new features in price prediction and may be closer to understanding financial bubbles in cryptocurrency trading.

**Balance between the opening of trading research literature and the fading of alphas.** Mclean et al. [314] pointed out that investors learn about mispricing in stock markets from academic publications. Similarly, cryptocurrency market predictability could also be affected by research papers in the area. A possible attempt is to try new pricing methods by applying real-time market changes. Considering the proportion of informed traders increasing in the cryptocurrency market in the pricing process is another breaking point (looking for a balance between alpha trading and trading research literature).

**2.13 Conclusions**

We provided a comprehensive overview and analysis of the research work on cryptocurrency trading. This survey presented a nomenclature of the definitions and current state of the art. The section provides a comprehensive survey of 177 cryptocurrency trading papers and analyses the research distribution that characterise the cryptocurrency trading literature. Research distribution among properties and categories/technologies are analysed in this survey respectively. We further summarised the datasets used for experiments and analysed the research trends and opportunities in cryptocurrency trading. Future research directions and opportunities are discussed in Section 2.12.

We expect this survey to be beneficial to academics (e.g., finance researchers) and quantitative traders alike. The survey represents a quick way to get familiar with the
literature on cryptocurrency trading and can motivate more researchers to contribute to the pressing problems in the area, for example along the lines we have identified.

By surveying related research topics related to cryptocurrency trading, we can conclude that modern machine learning has become widespread in the literature on trading and investment.

In the remainder of this thesis, we will explore two research gaps identified by this survey. Firstly, there is not a research on the universal features of cryptocurrency markets akin [381]. Secondly, experimenting with model selection and hyperparameter optimisation is a lengthy task that can discourage further advances in the field. In Chapter 3, we study the first question and apply machine learning techniques with complex structures (deep neural networks with auto-encoder) to explore this question and predict the direction of the cryptocurrency markets based on high-frequency limit order books. In Chapter 4, we introduce an optimisation approach to select models while tuning the hyperparameters using the importance of features.
Chapter 3

Ascertaining price formation in cryptocurrency markets with deep learning

3.1 Introduction

In high-frequency trading (HFT), price formation is based on the continuous matching of buy and sell orders in electronic markets. The process of price formation is influenced by a number of factors, including market liquidity, supply and demand, and news and information about the security being traded. HFT firms use a variety of strategies to capture profits from price formation. These strategies include market-making, statistical arbitrage, and high-frequency momentum trading. Market makers use their technology to provide liquidity to the market, buying and selling securities to balance supply and demand. Statistical arbitrage involves exploiting pricing discrepancies between related securities, while high-frequency momentum trading involves capturing short-term price movements in the market.

A powerful yet basic toolkit for algorithmic traders would efficiently predict the direction of price changes for financial assets; machine learning techniques, such as deep neural networks, are known as performant predictors for a variety of tasks and setups. This paper focuses on effectively applying neural networks on cryptocurrency market trading systems. Our objective is to predict the price changes; we consider both binary (up/down) and multi-class (e.g., degrees of increase/decrease) prediction of price changes.

The cryptocurrency market is a huge emerging market [8]. There were over 11,641 exchanges available on the internet as of July 2018 [156]. Most of them are exchanges of small capitalization with low liquidity. Exchanges with the highest 24-hour volume are FCoin, BitMEX, and Binance. Bitcoin, as the pioneer and also the market leader, has
3.1 Introduction

a market capitalization of over 112 billion USD, and a 24-hour volume over 3.8 billion USD in early July 2018. The cryptocurrency market is one of the most rapidly growing markets in the world, and are also considered one of the most volatile markets to trade in. For example, the price of a single Bitcoin increased significantly, from near zero in 2013 to nearly 19,000 USD in 2017. For some alt-coins, the price can increase or fall over 50% within a day. Therefore, having a method to accurately predict these changes is a pervasive task, but one that could achieve a long-term profit for cryptocurrency traders.

There are a number of research papers that studied the structure of the limit order book (i.e., the bids at both sides of the market) and, more generally, the micro-structure of the market by using different methods ranging from stochastic to statistical and machine learning approaches [215, 16, 171, 47, 335]. The Limit-Order Books of cryptocurrency markets share many common characteristics with those of traditional markets, especially at the microstructure level. The main difference is due to the lower average depth of the book in cryptocurrency markets; this leads to other differences related to the way the order book absorbs order flows and trade flow imbalances [380].

The objective of this paper is to understand the way prices at either side of the market move. Motivated by related literature, we focus on a particular measure, the mid-price, which intuitively captures the average difference between the best ask (the lowest price sellers are willing to accept) and best bid (the highest price buyers are willing to pay). Towards this aim we could, for example, use Markov chains to model the limit order book [248]. We could view the limit order book as a queuing system with a random process and use birth-and-death chains to model its behaviour. From this perspective, a natural way to explain the mid-price movement is to consider the value of the mid-price as the state of the chain. This value is controlled by the ratio between the probability of birth transitions \( p \) and the probability of death transitions \( q \). A ratio \( p/q \) greater than 1 within a short interval of time indicates that there is a higher chance for birth transitions to happen (more buyers), and the value of the mid-price is expected to increase. Similarly, if the ratio is smaller than 1, the value of the mid-price is expected to decrease [389]. The problem with this approach is the way it models the order book, namely, is the limit order book a queuing system? If it were, how to correctly simulate the random process and how to accurately estimate \( p \) and \( q \) become vital questions for this approach.

Easley et al. [149] researched similar topic in price formation. They investigated price dynamics in current complex markets using machine learning algorithms. We compare our research design with theirs along several dimensions (cf. Table 3.1). There are several differences. Firstly, our research aims at emerging market – cryptocurrency market – which is characterized by high risk and high returns. Easley et al. focus on future contracts; dollar-volume bars are used in their analysis. The core machine learning model in our research is LSTM whilst they focus on Random Forests. Many researchers found that
LSTM is more suitable than random forest in handling financial time series [169]. The memory cell in LSTM allows the model to remember relevant historical information more clearly. Secondly, our research uses sixteen features including basic market features and order book features while Easley et al. focus on features related to volume. Thirdly, there are differences in the methods used to detect features correlation; we use Principal Component Analysis (PCA) and Autoencoders while Easley et al. use Correlation Coefficient. Although Correlation Coefficient is a good method to find relationship among selected features, it is hard to reduce features’ dimension when some features are strongly related. Combining PCA and Autoencoder could reduce the interference between similar features, as implied by our research. Finally, we design a new retraining method to refresh obsolete predictive machine learning models. Updating the model frequently makes sense in financial prediction, as from our back-testing experiment.

Table 3.1 Comparison between our research and Easley et al. #FDR refers to Feature Dimensionality Reduction

<table>
<thead>
<tr>
<th></th>
<th>Our research</th>
<th>Easley et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Market</td>
<td>Cryptocurrency</td>
<td>Future Contracts</td>
</tr>
<tr>
<td>Data Frequency</td>
<td>Tick-level</td>
<td>Tick-level</td>
</tr>
<tr>
<td>Core ML Model</td>
<td>LSTM</td>
<td>Random Forest</td>
</tr>
<tr>
<td>Features</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td>Methods of #FDR</td>
<td>PCA &amp; Autoencoder</td>
<td>Correlation Coefficient</td>
</tr>
</tbody>
</table>

In this research, we propose to adopt a machine learning approach to reveal useful patterns from limit order books. We provide insights to a number of specific technical questions that arise from this approach. Specifically, we show that there are universal features amongst cryptocurrencies that can improve the predictive power of machine learning models, as there are in the case of equities [382]. The conceptual difference is that Sirignano and Cont focus on equities and our research focused on cryptocurrencies. We also show that feeding more data to train our deep neural network fails to improve the model performances; simpler single-dimensional models are preferred. Thirdly, we test the model on live data for different periods of varying length, which bears conceptual as well as technical challenges. Conceptually, we show that models developed with an ideal condition (carefully selected and split data) hardly perform well on real world cases, often because of sub-optimal accuracy and inefficient running time. This leads to the engineering challenge of designing a lean model which runs fast on live data, including retraining the model when necessary, whilst retaining accurate predictions. We show in this paper that, certain known architectures can meet both requirements, when using a novel training method that we call Walkthrough Training. Finally, we explore the problem of multi-label classification, by predicting “small” or “large” increase/decrease of the mid-price; we
3.2 Our tools

analyze the trade-off between performance and retrain frequency of Walkthrough Training in this context. Ultimately, our findings pave the way to the design of novel trading strategies and market estimators.

3.1.1 Roadmap

The paper is organized as follows. In Section 3.2, we provide a brief overview of the tools adopted, including machine learning, limit order books and data sources we used. In Section 3.3, we design experiments to address our research questions. In Section 3.4, we give a brief discussion of the validity of our findings. Section 3.5 gives a conclusion of this paper.

3.2 Our tools

In this section, we first review the background of machine learning and limit order books, before introducing an overview of the trading system where our prediction model is trained on.

3.2.1 Machine Learning

Artificial neural networks are computational algorithms mimicking biological neural systems, such as human brains. These algorithms are designed to recognize and generalize patterns from the input, and memorize them as weights in the neural network. The basic unit of a neural network is a neuron; a simple neural network, which is a conglomeration of neurons, is called Perceptron.

The neural network used in this paper is a type of recurrent neural network called Long-Short-Term-Memory (LSTM) [209]. This is distinct from the feed-forward neural network such as Perceptrons, since the output of the neural network sends feedback to the input and affects the subsequent output. Therefore, LSTM is better suited for handling sequential data where the previous data can have an impact on subsequent data; this, in principle, works well for time series data for price prediction and forecasting.

An LSTM cell contains a few gates and a cell status to help the LSTM cell decide what information should be kept and what information should be forgotten. As a result, the LSTM cell can recall important features from the previous prediction by having a cell state. An LSTM cell can also be viewed as a combination of a few simple neural networks, each of them serving a different purpose. The first one is the forget gate [209]. The previous output is concatenated with the new input and passed through a sigmoid function. After that, the output of the forget gate, \( f_t \), will perform a Hadamard product
3.2 Our tools

Fig. 3.1 An overview of an LSTM cell

(element-wise product) with the previous cell’s state. Note that $f_t$ is a vector containing elements that have a range from 0 to 1. A number closer to 0 means the LSTM should not recall it, whilst a number closer to 1 means the LSTM should recall and carry on to the next operation. This process helps the LSTM select which elements are to forget and remember, respectively. The second one is the input and activation gates [209]. This process concatenates the previous output with the new input, determines which element should be ignored, and updates the internal cell state. The cell state is then updated by a combination of the output and a transformation of the input. The third one is the output gate [209]. This process helps determine the output of the cell. Finally, the output of the LSTM cell is the Hadamard product of the current internal cell state and the output of the output gate [112, 4].

We use Root Mean Square Propagation (RMSprop) [394] – a stochastic gradient descent optimizer – to train the neural network, with the learning rate divided by the exponentially weighted average. Optimizer, learning rate and loss function are core concepts in machine learning models. Optimizer ties together the loss function and model parameters by updating the model in response to the output of the loss function. Loss function is a method of evaluating how well the algorithm models a given dataset, it tells the optimizer whether it’s moving in the right or wrong direction. The learning rate is a hyperparameter that controls how much the weight values should change in response to the estimated error each time the model’s weights are updated.
In our experiments, we also tested the use of an adaptive moment estimation, Adam in short, as the optimizer. While we observed that Adam helps the neural network to converge faster, we noted a tendency to overfit the data: the validation set has an increasing loss while the training set has a decreasing loss. This motivates our choice of RMSprop as optimizer.

### 3.2.2 Limit Order Books

The limit order book is technically a log file in the exchange showing the queue of the outstanding orders based on their price and arrival time. Let $p_b$ be the highest price at the buy side, which is called the best bid. The best bid is the highest price that a trader is willing to pay to buy the asset. Let $p_a$ be the lowest price at the sell side, which is called the best ask. The best ask is the lowest price a trader is willing to accept for selling the asset.

The mid-price of an asset is the average of the best bid and the best ask of the asset in the market.

$$M_p = \frac{(p_b + p_a)}{2}.$$  

There are other metrics that are also useful for describing the state of the limit order book: Spread, Depth and Slope.

### 3.2.3 Data source and overview of the envisioned trading system

Numerous exchanges provide Application Programming Interface (API) for systematic traders or algorithmic traders to connect to the exchange via software. Usually, an exchange provides two types of API, a RESTful API, and WebSocket API. Some exchanges also provide a Financial Information eXchange (FIX) protocol. In this study, a WebSocket API from an exchange called GDAX (Global Digital Asset Exchange) is used to retrieve the level-2 limit order book live data [188]. The level-2 data provides prices and aggregated depths for top 50 bids and asks. GDAX is one of the largest exchanges in the world owned by the Coinbase company.

Our focus is to design a model that can successfully predict the mid-price movement in the context of cryptocurrencies. Such a model is a component of a trading system, as shown in Figure 3.2. There are a few essential components for the trading system. First of all, the WebSocket is used to subscribe to the exchange and receive live data including tickers, order flows, and the limit order book’s update. Tickers data usually appears when two orders of the opposite side are matched and the opening of a candle on a candlestick...
3.2 Our tools

Fig. 3.2 An overview of a simple trading system

chart. Tickers contain the best bid, best ask, and the price, thus reflecting the change in price in real-time.

The ways the updates to the limit order book are communicated differ. Some exchanges provide a real-time snapshot of the order book. Some exchanges, including GDAX, only provide the update, i.e., updated data of a specific price and volume on the limit order book. Therefore, a local real-time limit order book is required to synchronize with the exchange limit order book. Additionally, we need to store all the data in a database. In this study, a non-relational database called MongoDB has been used to this purpose. Unlike a traditional relational database, MongoDB stores unstructured data in a JSON-like format as a collection of documents. The advantage of using a non-relational database is that data can be stored in a more flexible way.

The local copy of the limit order book is reconstructed by using level-2 limit order book updates. The reconstructed limit order book can provide information on the shape and status of the actual exchange limit order book. This limit order book can be used for calculating order imbalance and can provide quantified features of the limit order book. The input to the model is then finalized by a vectorizer, used as a data parser, combining information and extracting features from the ticker data and the local limit order book. Features are then reshaped into the format that can fit into an LSTM model.

We leave to future research the design and experimentation of a decision maker, which should make use of the prediction given by the trained model and help manage the inventory. If the inventory and certain thresholds are met, the decision-maker would place an order to the exchange based on the prediction from the trained LSTM model through RESTful API.
## 3.3 Experimental Study

### 3.3.1 Objective

The purpose of this research is to process real-time tick data using machine learning neural network approach on cryptocurrency trading system. As a machine learning model based on high-frequency trading, accuracy of prediction and computational efficiency are both important factors to consider in this research; accuracy here refers to the percentage of correct predictions made by the model.

### 3.3.2 Dataset

The data in Table 3.2 is used to perform universal machine learning research question RQ1. The data used in this study is live data recorded via a WebSocket through the GDAX exchange WebSocket API. only use as features the top 7 because farther depths contain less information relevant to the price formation, c.f. Table 3.4. More data is collected for The time range of the collected data is from the time of 2018-07-02 17:22:14 to 2018-07-03 23:32:53. BTC-USD data from 2018-08-08 14:31:54 to 2018-08-09 09:01:13, BTC-USD data from 2018-08-11 12:09 to 2018-08-16 23:59, and from 2018-08-24 12:07 to 2018-08-29 23:59 are collected for live back-testing. The order flow data contain 61,909,286 records, the tickers data include 128,593 ticker data points, and the level-2 data contain 40,951,846 records. Table 3.2 lists the available assets on the GDAX exchange and the corresponding number of records.

<table>
<thead>
<tr>
<th>product id \ data type</th>
<th>Ticker</th>
<th>Level-2</th>
<th>Order Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCH-USD</td>
<td>15,213</td>
<td>1,600,474</td>
<td>2,442,323</td>
</tr>
<tr>
<td>BTC-EUR</td>
<td>9,769</td>
<td>4,656,627</td>
<td>7,002,588</td>
</tr>
<tr>
<td>BTC-GBP</td>
<td>3,726</td>
<td>8,849,556</td>
<td>13,280,280</td>
</tr>
<tr>
<td>BTC-USD</td>
<td>25,904</td>
<td>4,110,818</td>
<td>6,282,022</td>
</tr>
<tr>
<td>ETH-BTC</td>
<td>4,016</td>
<td>1,250,202</td>
<td>1,893,851</td>
</tr>
<tr>
<td>ETH-EUR</td>
<td>3,180</td>
<td>4,876,886</td>
<td>7,323,178</td>
</tr>
<tr>
<td>ETH-USD</td>
<td>27,089</td>
<td>6,087,574</td>
<td>9,276,806</td>
</tr>
<tr>
<td>LTC-BTC</td>
<td>2,167</td>
<td>611,682</td>
<td>923,070</td>
</tr>
<tr>
<td>LTC-EUR</td>
<td>4,243</td>
<td>1,260,024</td>
<td>1,897,731</td>
</tr>
<tr>
<td>LTC-USD</td>
<td>32,203</td>
<td>2,391,377</td>
<td>3,700,271</td>
</tr>
<tr>
<td>BCH-EUR</td>
<td>4,243</td>
<td>5,822,103</td>
<td>7,934,653</td>
</tr>
</tbody>
</table>

Following [74], we statistically analyse our datasets in Table 3.3; “Dataset1 - Dataset3” refer to data collected from three time periods for pair BTC-USD. The signal-to-noise
ratio (SNR) of high-frequency data is often low, meaning that there is a lot of noise in the data that can make it difficult to extract meaningful signals. However, despite the low SNR, it can still be appropriate to use raw tick data for analysis. Raw tick data contains more information than lower-frequency data. This can allow for more detailed analysis and potentially more accurate models. Using raw tick data can reduce the latency of analysis since there is no need to wait for aggregation or processing of lower frequency data. And also aggregating data can hide important details about market dynamics, such as the impact of large trades on the market. To use raw tick data with low SNR, such as outliers and bias, we address the challenges in these ways: normalizing the data, using robust statistical methods, and choosing a model that is robust to noise and outliers such as the deep neural network.

### Table 3.3 Data statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Mean</th>
<th>Median</th>
<th>Max</th>
<th>Min</th>
<th>St.Dev</th>
<th>Skew</th>
<th>Kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset1</td>
<td>7,362.338</td>
<td>7,448.725</td>
<td>7,747.219</td>
<td>6,687.066</td>
<td>244.084</td>
<td>-0.578</td>
<td>-1.192</td>
</tr>
<tr>
<td>Dataset2</td>
<td>6,260.872</td>
<td>6,298.001</td>
<td>6,596.083</td>
<td>5,904.602</td>
<td>144.310</td>
<td>-0.459</td>
<td>-0.585</td>
</tr>
<tr>
<td>Dataset3</td>
<td>6,408.662</td>
<td>6,322.626</td>
<td>7,132.121</td>
<td>5,904.602</td>
<td>335.889</td>
<td>1.045</td>
<td>-0.06</td>
</tr>
</tbody>
</table>

In Table 3.4, we have identified the set of features that we use as our input data; these are divided into two categories: basic and order book features. All of these can be directly computed from the aforementioned data. According to [129], price is formed by the supply and demand at a given point in time. In a high-frequency environment, supply and demand are driven by the spread and liquidity, which might affect the rate of mean reversion and price diffusion in a tick-level price change. At the same time, the technical indicators are another ways to know the short-term pressure and support levels for market prices. Limit order book features are important in high-frequency trading because they provide crucial information about the state of the market and can be used to make informed trading decisions. By looking at the limit order flows, many orders at different price levels can indicate a liquid and active market. The features we used in this chapter are mainly related to price, volume, and simple calculations with volume and price information.

### 3.3.3 Methodology

#### Model Architecture

The simple architecture in Figure 3.3 served as the predictive model in this study. This neural network contains two layers of LSTM cells, one layer of fully connected neurons, and one layer of softmax as the output layer which outputs the probability of price movements. The two layers of LSTM cells can be viewed as a filter for capturing non-linear
3.3 Experimental Study

Table 3.4 Feature Set

<table>
<thead>
<tr>
<th>Basic Features</th>
<th>Description (i denotes time step)</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1 = {P_i}</td>
<td>price</td>
</tr>
<tr>
<td>f2 = {V_i}</td>
<td>last size</td>
</tr>
<tr>
<td>f3 = {ln(P_i/P_{i-1})}</td>
<td>log return</td>
</tr>
<tr>
<td>f4 = {P_i - P_{i-1}}</td>
<td>price difference</td>
</tr>
<tr>
<td>f5 = {ema_t = \beta P_t + (1 - \beta)ema_{t-1}, \beta = (2/5)}</td>
<td>EMA 4 periods</td>
</tr>
<tr>
<td>f6 = {ema_t = \beta P_t + (1 - \beta)ema_{t-1}, \beta = (1/5)}</td>
<td>EMA 9 periods</td>
</tr>
<tr>
<td>f7 = {ema_t = \beta P_t + (1 - \beta)ema_{t-1}, \beta = (2/19)}</td>
<td>EMA 18 periods</td>
</tr>
<tr>
<td>f8 = {rsi = 100 - 100/(1 + RS), RS = AvgGain/AvgLoss in 3 periods}</td>
<td>RSI 3 period</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order Book Features</th>
<th>Description (n = 7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>f9 = {[(P^ask_i - P^bid_i)/2]_{i=1}^n}</td>
<td>bid-ask spread</td>
</tr>
<tr>
<td>f10 = {[(P^ask_i + P^bid_i)/2]_{i=1}^n}</td>
<td>mid-price</td>
</tr>
<tr>
<td>f11 = {[(P^ask_i + P^bid_i)/2]<em>{i=1}^n - [(P^ask</em>{i-1} + P^bid_{i-1})/2]_{i=2}^n}</td>
<td>mid-price difference</td>
</tr>
<tr>
<td>f12 = {[(P^ask_i, P^bid_i)]_{i=1}^n}</td>
<td>bids and asks</td>
</tr>
<tr>
<td>f13 = {[(D^ask_i, D^bid_i)]_{i=1}^n}</td>
<td>depths of bids and asks</td>
</tr>
<tr>
<td>f14 = {\sum_{i=1}^n[D^ask_i + D^bid_i]}</td>
<td>cumulative sum of the depths</td>
</tr>
<tr>
<td>f15 = {[P^bid_i/P^ask_i]_{i=1}^n}</td>
<td>slope of bids and asks</td>
</tr>
<tr>
<td>f16 = {cumsum[(P^ask_i - P^bid_i)]_{i=1}^n}</td>
<td>Cumulative sum of bid-ask diff</td>
</tr>
</tbody>
</table>

features from the data, and the fully connected layer can be viewed as the decision layer based on the features provided by the last LSTM layer. This neural network is designed as simple as possible because in the tick data environment, every millisecond matters. Reducing the number of layers and neurons can significantly reduce the computational complexity, thus the time required for the data processing.

Fig. 3.3 LSTM model architecture
3.3 Experimental Study

Model Hyperparameter Setting

**LSTM Architecture.**
1. Sequential()
2. LSTM(40, dropout=0.5, kernel regularizer=regularizers.l1 l2(0.000005, 0.000005), activity regularizer=regularizers.l2(0.0008))
3. LSTM(20, dropout=0.5, kernel regularizer=regularizers.l1 l2(0.000005, 0.000005), activity regularizer=regularizers.l2(0.0008))
4. Dense(15, activation='tanh', kernel regularizer=regularizers.l1 l2(0.0001, 0.0005), activity regularizer=regularizers.l2(0.0008))
5. Dropout(0.4)
6. Dense(2, activation='softmax', kernel regularizer=regularizers.l1 l2(0.000005, 0.000002), activity regularizer=regularizers.l1 l2(0.000001, 0.000002))
7. optimizer = RMSprop(lr=0.010, rho=0.70, epsilon=1e-8, decay=0.005)
8. earlyStop = EarlyStopping(monitor='val loss', min delta=0.0001, patience=200, verbose=0, mode='auto', baseline=None)
9. batch size=64, epochs=1200

**Autoencoder Architecture.**
1. encoded = LSTM(60, return sequences=False, dropout=(0.2))
2. decoded = RepeatVector(timesteps)(encoded)
3. decoded = LSTM(60, return sequences=True, dropout=(0.2))
4. decoded = LSTM(data dim, return sequences=True, kernel regularizer=regularizers.l1 l2(0.00002, 0.00008), activity regularizer=regularizers.l1 l2(0.00002, 0.00008))
5. optimizer = 'adam'
6. earlyStop = EarlyStopping(monitor='loss', min delta=0.0001, patience=20, verbose=0, mode='auto', baseline=None)
7. pca = PCA(n components=data dim)
8. batch size=64, epochs=800

**Multi-label prediction**

Binary classification can be scarcely informative to a trader, as “small” variations are not differentiated from “big” ones. One might want to hold one’s position in the former case and transact only in the latter.

We use 1-min and 5-min data to demonstrate the rate of price change, defined as the ratio between the price change and the transaction (close) price. In both cases, most relative price changes fall in $-0.25\%$ and $0.25\%$. Often these percentages are less than the transaction fees and traders should be able to know when this is the case to develop a successful trading strategy. Therefore, we also investigate multi-label prediction based on
In this multi-label prediction, we replace binary target prediction with four-target prediction. At the structure level, we have four softmax units as output layer instead of two units. By effectively set the boundaries of four units, we can transform the original two-class classifier into a four-class classifier. Using the fees used by Coinbase Pro [352], we use $\pm 0.2\%$ of the transaction price as a reasonable threshold to differentiate large and small changes (see Table 3.5 where we also name the intervals for future references).

![Distribution of historical price changes](image)

**Fig. 3.4 Distribution of historical price changes**

<table>
<thead>
<tr>
<th>Label</th>
<th>Relative Price Change</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significant increase</td>
<td>$(+0.2%, +\infty)$</td>
<td>Sensitive Interval</td>
</tr>
<tr>
<td>Significant decrease</td>
<td>$(-0.2%, -\infty)$</td>
<td></td>
</tr>
<tr>
<td>Insignificant increase</td>
<td>$(0, +0.2%]$</td>
<td>Insensitive Interval</td>
</tr>
<tr>
<td>Insignificant decrease</td>
<td>$[-0.2%, 0)$</td>
<td></td>
</tr>
</tbody>
</table>

**Walkthrough Training**

Prediction model in financial market has timeliness; this is especially true for the high-frequency financial market. For example, should we use historical financial data from 2015 to train a model and test it on 2017 data for predictions, this model might not have a good performance. The old model might not adapt well to the new market environment as it has been trained and tailored on old market conditions. Although a machine learning approach can largely increase prediction accuracy of stock market, such models need to adapt themselves because the stock market is constantly changing.
3.3 Experimental Study

Wan and Banta [416] propose the parameter incremental learning (PIL) method for neural networks; the main idea is that the learning algorithm should not only adapt to the newly presented input-output training pattern by adjusting parameters, but also preserve the prior results. Inspired from this, we propose a method called *Walkthrough Training* in machine learning for our task. This approach is designed to retrain the original machine learning model itself when it “appears” to no longer be valid. We consider two different Walkthrough training methods.

(i). *Walkthrough with stable retrain frequency*. Considering different trading cycles based on the data obtained from the API, we retrain our model at fixed time intervals. The length of the interval depends on our trading strategy and accuracy from data we obtained. This way of retraining helps the model to adjust to the newly acquired features and retain the knowledge gained from the original training.

(ii). *Walkthrough with dynamic retrain frequency*. We use Maximum Accuracy Drawdown (MAD), which is the maximum observed accuracy loss from a peak to a trough before a new peak is attained, as a condition of dynamic retraining. The idea is that stable retraining is not suitable for every condition in retraining model. More specifically, if the old model is aimed for long-term prediction, stable retraining will lead to waste of computing resources and overfitting problem (the model fits the data too well and leads to low prediction accuracy on unseen data). During the process of prediction based on this method, we monitor accuracy of prediction over time. In the following formula, “Min Accuracy Value” and “Max Accuracy Value” identify the highest and lowest prediction accuracy, respectively. All parameters in the formula are in interval between last retraining time and current calculation time. After calculation, “Modified MAD” is considered as hyper-parameter in the whole prediction model to optimize the retraining time.

\[
\text{Modified MAD} = \frac{\text{Max Accuracy Value} - \text{Min Accuracy Value}}{\text{Min accuracy Value}}.
\]

The modified MAD is a measure of accuracy loss that looks for greatest effective period of model. When modified MAD is over 15%, we consider the original machine learning model to be no longer applicable for latest market data. In such a case, we use historical data up to the point when the MAD is measured as training data to retrain original machine learning model. This process will be used throughout the whole time series prediction.
3.3 Experimental Study

Logistic Regression as a benchmark

Another model like default hyperparameters Logistic Regression has been used in the same condition. The same features are to build a logistic regression model. For the July dataset, it can converge. Also, it shows a decent accuracy on the July dataset. But it couldn’t converge on the August dataset. If I force it to converge using other methods like bfgs (Broyden-Flether-Goldfarb Shanno), it will result in one-side prediction or an overfitting model on one side. From the result below, we can see the recall of downticks is much lower than the recall of upticks. Therefore, we assumed that in a short time interval, the LSTM shows some predictive power. However, in the long run, the LSTM model could fall back to the Logistic model. The classification report is as follows:

<table>
<thead>
<tr>
<th>Classification report using Logistic Regression in July</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.70</td>
<td>0.76</td>
<td>0.73</td>
</tr>
<tr>
<td>↓</td>
<td>0.75</td>
<td>0.68</td>
<td>0.71</td>
</tr>
<tr>
<td>avg</td>
<td>0.72</td>
<td>0.72</td>
<td>0.72</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification report using Logistic Regression in August</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.73</td>
<td>0.96</td>
<td>0.83</td>
</tr>
<tr>
<td>↓</td>
<td>0.92</td>
<td>0.56</td>
<td>0.70</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.72</td>
<td>0.78</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Decision Tree as a benchmark

Another model like default hyperparameters Decision Tree has been used in the same condition. The same features are to build a Decision Tree model. The classification report is as follows:

Random Forest as a benchmark

Another model like default hyperparameters Random Forest has been used in the same condition. The same features are to build a Random Forest model. The classification report is as follows:
### 3.3 Experimental Study

Table 3.7 Classification report

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.79</td>
<td>0.77</td>
<td>0.78</td>
</tr>
<tr>
<td>↓</td>
<td>0.37</td>
<td>0.39</td>
<td>0.38</td>
</tr>
<tr>
<td>avg</td>
<td>0.58</td>
<td>0.58</td>
<td>0.58</td>
</tr>
</tbody>
</table>

Classification report using Decision Tree in August

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.83</td>
<td>0.82</td>
<td>0.82</td>
</tr>
<tr>
<td>↓</td>
<td>0.38</td>
<td>0.40</td>
<td>0.39</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.61</td>
<td>0.61</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Table 3.8 Classification report

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.78</td>
<td>0.97</td>
<td>0.86</td>
</tr>
<tr>
<td>↓</td>
<td>0.66</td>
<td>0.17</td>
<td>0.28</td>
</tr>
<tr>
<td>avg</td>
<td>0.72</td>
<td>0.57</td>
<td>0.57</td>
</tr>
</tbody>
</table>

Classification report using Random Forest in August

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.82</td>
<td>0.97</td>
<td>0.89</td>
</tr>
<tr>
<td>↓</td>
<td>0.67</td>
<td>0.23</td>
<td>0.35</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.74</td>
<td>0.60</td>
<td>0.62</td>
</tr>
</tbody>
</table>
3.3 Experimental Study

Support vector machine as a benchmark

Another model like default hyperparameters support vector machine has been used in the same condition. The same features are to build a support vector machine model. The classification report is as follows:

Table 3.9 Classification report

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.75</td>
<td>1.00</td>
<td>0.86</td>
</tr>
<tr>
<td>↓</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>avg</td>
<td>0.37</td>
<td>0.50</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Classification report using support vector machine in August

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.78</td>
<td>1.00</td>
<td>0.88</td>
</tr>
<tr>
<td>↓</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>avg</td>
<td>0.39</td>
<td>0.50</td>
<td>0.44</td>
</tr>
</tbody>
</table>

XGBoost as a benchmark

Another model like default hyperparameters XGBoost has been used in the same condition. The same features are to build a XGBoost model. The classification report is as follows:

Table 3.10 Classification report

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.79</td>
<td>0.94</td>
<td>0.86</td>
</tr>
<tr>
<td>↓</td>
<td>0.59</td>
<td>0.26</td>
<td>0.36</td>
</tr>
<tr>
<td>avg</td>
<td>0.69</td>
<td>0.60</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Classification report using XGBoost in August

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.83</td>
<td>0.94</td>
<td>0.88</td>
</tr>
<tr>
<td>↓</td>
<td>0.61</td>
<td>0.32</td>
<td>0.42</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.72</td>
<td>0.63</td>
<td>0.65</td>
</tr>
</tbody>
</table>
3.3.4 Research Questions

We investigate four specific research questions (RQs, for short) in our general context of interest, price predictions through a machine learning model within the cryptocurrency markets.

RQ1: How well does a universal machine learning model perform?
Sirignano and Cont [382] found that a universal machine learning model would predict well the price formation in relation to stock market. We ask this question to understand if a similar conclusion can be drawn for more emergent, less mature and more volatile cryptocurrency market.

RQ2: How many successive data points should we use to train machine learning models?
The sequential nature of time series naturally puts forward the question of optimizing the number of subsequent data points (i.e., time steps) used to train the deep network. Does it make sense to use more than one data point at a time? If so, how many time steps should be used?

RQ3: How well do machine learning models work on live data?
A good offline prediction based on machine learning may fail to perform well on live data, due to evolving patterns in a highly volatile environment like ours. Is there an accuracy decay on live data? If yes, would Walkthrough training methods help address the issue? Moreover, we want to understand if lean and fast architectures can perform well with tick online data.

RQ4: What is the best Walkthrough method in the context of multi-label prediction?
Making profit on tick data predictions might be too hard for a number of reasons. Firstly, the execution time of the order might make the prediction on the next tick obsolete. Secondly, in the context of multi-label predictions, there might be very few data points in the sensitive intervals which would make transactions potentially more profitable than transaction costs. We therefore wish to determine the best Walkthrough method when we use minute-level data for the task of multi-label classification.

Our research questions are novel for a number of reasons. In RQ1, we analyse the effects of universality in cryptocurrency markets, which is an extension of Sirignano and Cont [382]. Given that the asset classes considered are rather different, it is interesting to study whether a sort of transfer learning translates across different markets. Similarly, whilst RQ2 has been studied by others, few researchers considered the problem for cryptocurrency
prediction models. As for RQ3, we are not aware of any study in which the proposed models are tested on live data; this requires a balance between model complexity and performance. In RQ4, we test the performances of a brand new method in re-training the machine learning model. Ultimately, the findings from the questions above will help a cryptocurrency trader to design a better model and ultimately devise a more profitable trading strategy (i.e., the decision maker in the system of Figure 3.2).

### 3.3.5 Results and Analysis

We organize the discussion of our results according to the research questions of interest. The answer to each question informs the design used to address the challenges of the subsequent questions. In this sense, we use an incremental approach to find our results.

**How well does a universal machine learning model perform?**

We begin by examining RQ1, through training product specific networks of Figure 3.3 in order to establish the baseline for comparison. For each product (i.e., currency pair), five neural networks with the same architecture are initialized. Five training sets are then created by extracting the first 10%, 20%, 50%, 70%, and 85% from the total data of the product. The neural networks are trained and tested with each data split. For example, a product-specific, such as BCH-USD, neural network is trained with the first 10% of the total data using only one time step; the rest of the data are then used to evaluate the performance of the neural network. Subsequently, another neural network is trained and tested with a different amount of data and so on.

The purpose of using this training approach is to evaluate the importance of the amount of data used. The high-frequency markets are often considered extremely noisy and full of unpredictability. If neural networks for the same product showed no performance gain with increasing amount of training data, then it may actually be the case that the majority of the data is noise. In these circumstances, a stochastic model might be a better option than a data-driven model, because a simpler model generally tends to be less overfitting compared to a complex model under noisy environment.

From the result in Table 3.11, the currency pairs with very little samples, such as BCH-EUR, BTC-GBP, ETH-EUR, and BTC-EUR, show a decreasing performance after using training data with a size greater than 50% (shown as Figure 3.5). The decrease in the performance could be a direct result of the lack of testing cases. For other currency pairs, the currency-pair-specified neural network models show a general rise in accuracy when increasing the size of the training data (Figure 3.6), which suggests that there might be some recognizable patterns in the data. The box plots (Figure 3.7) show the comparison of currency pairs with and without improvement. The result above suggests that, at least
3.3 Experimental Study

for our architecture, the neural network is able to learn the hidden pattern from within a dataset when given a sufficient amount of data for most of the currency pairs.

Table 3.11 Out-of-sample accuracy with respect to training sample sizes

<table>
<thead>
<tr>
<th>currency pair</th>
<th>Sample size used in training</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10%</td>
</tr>
<tr>
<td>BCH-USD</td>
<td>0.619</td>
</tr>
<tr>
<td>BTC-EUR</td>
<td>0.554</td>
</tr>
<tr>
<td>BTC-GBP</td>
<td>0.611</td>
</tr>
<tr>
<td>BTC-USD</td>
<td>0.702</td>
</tr>
<tr>
<td>ETH-BTC</td>
<td>0.788</td>
</tr>
<tr>
<td>ETH-EUR</td>
<td>0.633</td>
</tr>
<tr>
<td>ETH-USD</td>
<td>0.599</td>
</tr>
<tr>
<td>LTC-BTC</td>
<td>0.579</td>
</tr>
<tr>
<td>LTC-EUR</td>
<td>0.505</td>
</tr>
<tr>
<td>LTC-USD</td>
<td>0.574</td>
</tr>
<tr>
<td>BCH-EUR</td>
<td>0.540</td>
</tr>
</tbody>
</table>

Table 3.12 Out-of-sample precision with respect to training sample sizes

<table>
<thead>
<tr>
<th>currency pair</th>
<th>Sample size used in training</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10%</td>
</tr>
<tr>
<td>BCH-USD</td>
<td>0.668</td>
</tr>
<tr>
<td>BTC-EUR</td>
<td>0.554</td>
</tr>
<tr>
<td>BTC-GBP</td>
<td>0.620</td>
</tr>
<tr>
<td>BTC-USD</td>
<td>0.752</td>
</tr>
<tr>
<td>ETH-BTC</td>
<td>0.81</td>
</tr>
<tr>
<td>ETH-EUR</td>
<td>0.643</td>
</tr>
<tr>
<td>ETH-USD</td>
<td>0.676</td>
</tr>
<tr>
<td>LTC-BTC</td>
<td>0.740</td>
</tr>
<tr>
<td>LTC-EUR</td>
<td>0.501</td>
</tr>
<tr>
<td>LTC-USD</td>
<td>0.613</td>
</tr>
<tr>
<td>BCH-EUR</td>
<td>0.545</td>
</tr>
</tbody>
</table>

We are now ready to test the findings of Sirignano and Cont [382] about the existence of a universal predictive model in the context of cryptocurrencies. We are interested to see whether a universal predictive model for all available currency pairs can outperform the product-specific ones introduced above. Table 3.14 displays the performance of different models using F1-score as performance measure. We selected F1 score as an indicator of
### Table 3.13 Out-of-sample f1 with respect to training sample sizes

<table>
<thead>
<tr>
<th>Currency pair</th>
<th>10%</th>
<th>20%</th>
<th>50%</th>
<th>70%</th>
<th>85%</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCH-USD</td>
<td>0.592</td>
<td>0.656</td>
<td>0.669</td>
<td>0.696</td>
<td>0.662</td>
</tr>
<tr>
<td>BTC-EUR</td>
<td>0.554</td>
<td>0.526</td>
<td>0.540</td>
<td>0.596</td>
<td>0.320</td>
</tr>
<tr>
<td>BTC-GBP</td>
<td>0.603</td>
<td>0.473</td>
<td>0.634</td>
<td>0.480</td>
<td>0.416</td>
</tr>
<tr>
<td>BTC-USD</td>
<td>0.688</td>
<td>0.786</td>
<td>0.793</td>
<td>0.825</td>
<td>0.810</td>
</tr>
<tr>
<td>ETH-BTC</td>
<td>0.785</td>
<td>0.824</td>
<td>0.839</td>
<td>0.774</td>
<td>0.728</td>
</tr>
<tr>
<td>ETH-EUR</td>
<td>0.627</td>
<td>0.630</td>
<td>0.714</td>
<td>0.658</td>
<td>0.592</td>
</tr>
<tr>
<td>ETH-USD</td>
<td>0.555</td>
<td>0.573</td>
<td>0.481</td>
<td>0.703</td>
<td>0.734</td>
</tr>
<tr>
<td>LTC-BTC</td>
<td>0.496</td>
<td>0.666</td>
<td>0.735</td>
<td>0.751</td>
<td>0.720</td>
</tr>
<tr>
<td>LTC-EUR</td>
<td>0.484</td>
<td>0.428</td>
<td>0.575</td>
<td>0.595</td>
<td>0.671</td>
</tr>
<tr>
<td>LTC-USD</td>
<td>0.531</td>
<td>0.591</td>
<td>0.765</td>
<td>0.785</td>
<td>0.814</td>
</tr>
<tr>
<td>BCH-EUR</td>
<td>0.529</td>
<td>0.511</td>
<td>0.511</td>
<td>0.365</td>
<td>0.344</td>
</tr>
</tbody>
</table>

#### Fig. 3.5 Currency pairs without improvement
3.3 Experimental Study

Fig. 3.6 Currency pairs with improvement

Fig. 3.7 Box plots of currency pair with and without improvement
measuring accuracy. F1 score is defined as

\[
F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

where Precision is the fraction of relevant instances among the retrieved instances (i.e., the ratio between True Positives and the sum of true and false positives) and Recall is fraction of the total amount of relevant instances that were actually retrieved (that is, the ratio between true Positives and the sum of true positives and false negatives). F1 score is an important evaluation measure when we are not familiar with the target class distribution. The label “AVG” represents the mean performance of all the individual currency models. The label “Universal” represents using joint data (merging all models as a new model).

We know from the analysis above (Figure 3.5 and 3.6) that for some currency pairs, the current neural network architecture is not performing very well. Therefore, for more precise and targeted analysis, those currency pairs are excluded from the original dataset, and a new dataset is generated without them. The label of “Selected” represents the mean performance of all models excluding those pairs, namely, BCH-EUR, BTC-GBP, ETH-EUR and BTC-EUR. The “Universal selected” neural network is trained with the “selected” approach but with joined data across all available products.

Table 3.14 Models’ performance with different sample sizes used in training

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>AVG</th>
<th>Selected</th>
<th>Universal</th>
<th>Universal Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0.64789</td>
<td>0.60456</td>
<td>0.66399</td>
<td>0.67382</td>
</tr>
<tr>
<td>20%</td>
<td>0.61996</td>
<td>0.61417</td>
<td>0.64996</td>
<td>0.67045</td>
</tr>
<tr>
<td>50%</td>
<td>0.58626</td>
<td>0.61604</td>
<td>0.64966</td>
<td>0.66489</td>
</tr>
<tr>
<td>70%</td>
<td>0.56106</td>
<td>0.60056</td>
<td>0.63986</td>
<td>0.66286</td>
</tr>
<tr>
<td>85%</td>
<td>0.53878</td>
<td>0.58640</td>
<td>0.62184</td>
<td>0.65183</td>
</tr>
</tbody>
</table>

We can see that the universal model slightly outperforms the mean of product-specific models, for each size of the training set, by an average of 5.88% in terms of F1-score. Similarly, the universal with selected currency pairs outperforms the selected product-specific model by an average of 7.50%. In general, both of the universal models achieved higher F1-score than the product-specific ones. Therefore, we can conclude that the universal model has better performance than the currency-pair specific model. The performance gain in the universal model and the universal model with selected currency pairs may be explained with the following rationale. Firstly, there are some universal features on the limit order book which could be observed by the LSTM neural network for most of the currency pairs on the exchange. Secondly, the increased amount of the training data helps the network to generalize better, since 10% of joined data is much larger than 10% of one currency pair data. It also means that the LSTM model can learn the pattern from the data of multiple currency pairs having the same time horizon, then apply the pattern to another currency pair. To test whether this difference is statistically significant, we ran the t-test and Wilcoxon test between the performance of product-specific and universal models.
3.3 Experimental Study

models (using accuracy as criteria). The \(t\)-test has the result that statistic is -53.885 and \(p\)-value is 3.0302e-22 (\(\ll 0.0000\)). The Wilcoxon test shows a \(p\)-value of 8.5745e-05 (\(\ll 0.0000\)). From both tests, we can conclude that the product-specific and universal models are statistically different.

We reach the following conclusion from this section. The answer to RQ1 is that the universal model has better performance than the currency-pair specific model for all the available currency pairs in (the chosen) cryptocurrency market.

**How many successive data points should we use to train machine learning models?**

In this subsection we examine RQ2. Informed by our findings in relation to RQ1, we next fix the training set size to 70% of the total sample size and focus our attention to the universal and universal selected neural networks. To investigate RQ2, we train both networks with 70% of the total data using increasing time steps of 1, 3, 5, 7, 10, 20, 40. For example, the 3-time-steps input contains the feature vector of the current tick \(F_t\), feature vector of the previous tick \(F_{t-1}\), and feature vector of two ticks prior \(F_{t-2}\). This approach aims to discover whether there are any observable patterns related to the sequence of data and how persistent it is.

The results are shown in Table 3.15. To make sense of them, we fit the data points in a linear regression model, with ordinary least squares, and obtain the coefficients in Table 3.16, where Const represents the intercept and X the slope.

<table>
<thead>
<tr>
<th>timesteps</th>
<th>Universal Precision</th>
<th>Universal Accuracy</th>
<th>Universal F1</th>
<th>Universal Selected Precision</th>
<th>Universal Selected Accuracy</th>
<th>Universal Selected F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7179</td>
<td>0.7111</td>
<td>0.7097</td>
<td>0.7312</td>
<td>0.7294</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.7263</td>
<td>0.7263</td>
<td>0.7260</td>
<td>0.7445</td>
<td>0.7442</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.7209</td>
<td>0.7200</td>
<td>0.7200</td>
<td>0.7419</td>
<td>0.7419</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.7086</td>
<td>0.7086</td>
<td>0.7086</td>
<td>0.7389</td>
<td>0.7389</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.7146</td>
<td>0.7146</td>
<td>0.7146</td>
<td>0.7389</td>
<td>0.7389</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.7136</td>
<td>0.7116</td>
<td>0.7114</td>
<td>0.7421</td>
<td>0.7421</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>0.7138</td>
<td>0.7131</td>
<td>0.7131</td>
<td>0.7275</td>
<td>0.7273</td>
<td></td>
</tr>
</tbody>
</table>

As depicted in Figure 3.8, the slopes of the linear equations are very close to zero, and are negative. This result suggests that increasing the time steps of the training data does not have a significant effect on the performance of the model. On the contrary, increasing the time steps too much may also have a negative impact to the model’s performance. We
3.3 Experimental Study

<table>
<thead>
<tr>
<th>Universal</th>
<th>Universal Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>Accuracy</td>
</tr>
<tr>
<td>Const</td>
<td>X</td>
</tr>
<tr>
<td>0.7186</td>
<td>-0.0002</td>
</tr>
</tbody>
</table>

here stress that Tran et al. [403] studied this problem by applying a temporal attention-augmented bilinear network and testing using three types of movement (decrease, stationary and increase). The prediction horizon considered is 10, 20 and 50. Their method gives a comparison among different machine learning methods, whilst we here focus on our particular universal model.

Fig. 3.8 Relationship between accuracy and number of time steps used in training

The answer to RQ2 is that one time step/data point is the best choice in our context. Choosing one time step carries some further advantages; one step, in fact, opens the possibility of using different machine learning algorithms since most of them are not designed to handle sequential input.

How well do machine learning models work on live data?

We here examine RQ3. In this section, the challenges and performances of using our predictive models on live data are discussed.

The red line in Figure 3.9 is the baseline of the performance, and the blue line is the performance of the predictive model on live data. This figure shows that the performance of the universal model slowly decays to almost random guessing over the period of interest. This behavior could be caused by some non-stationary features of the limit order book,
which means that the hidden pattern captured by the universal model is no longer applicable to the new data.

![Performance decay on the live data](image)

Fig. 3.9 Performance decay on the live data

To resolve this problem, an autoencoder is used (Figure 3.10). In general, autoencoders are often used for data compression and feature extraction tasks, which can help to reduce the complexity of the data and make it more manageable for downstream tasks. This can also help to reduce the non-stationarity of the data by extracting meaningful features that are more robust to changes in the data distribution over time. Autoencoders can also be used to retrain models more efficiently by providing a pre-processing step that extracts useful features from the data. By training the autoencoder on a large dataset, the model can learn to extract useful features that can be used to train downstream models with much smaller datasets. The characteristic of the autoencoder is that the input layer and the output layer usually have the same number of neurons, and the hidden layers of the autoencoder must have a lower number of neurons compared to the input and output layers. The reason for using such an architecture is that the reduced number of neurons in the hidden layers can form a bottleneck in the neural network. Thus, the autoencoder cannot learn by simply remembering the input only. This architecture, in fact, forces the autoencoder to compress the input data and then decompress the data before outputting it. Therefore, the autoencoder can learn from the input structure. The trained autoencoder performs two tasks. The first one is to remove noise; the trained autoencoder can suppress abnormal features by reconstructing the input data. This process usually removes abnormal spikes in a feature. The second one is to map the new data into a more familiar space for the LSTM model.

Figure 3.11 shows the prediction of the LSTM with an autoencoder by using live data of BTC-USD from 2018-08-01 15:10:43 to 2018-08-02 08:33:50. Bitcoin has a great dominance and the BTC-USD is also the most traded product on the market. The
3.3 Experimental Study

Fig. 3.10 Architecture of the autoencoder

Performance decays slower with the autoencoder than the original LSTM model. Figure 3.12 is the distribution of the predictions made by the universal model with autoencoder and the aggregated real-time target; each point of the aggregated real-time target is equal to the mean of upticks and downticks for every 20 samples. The red line depicts the ratio of downticks given by the predictive model, and the grey line is the ratio of downticks given by real-time target. From the distribution of prediction and real-time target we can observe that the autoencoder is slightly biased to the downtrend market. This explains the gradual decrease in the accuracy under the uptrend market after the 3,000 predictions mark (cf. Figure 3.11) because small errors accumulate over time and eventually affect the overall accuracy. In other words, the biased training data could cause a biased model. For example, the training data used to train the model could be experiencing a bearish market so that the model is more sensitive to the downtrends.

Fig. 3.11 Performance of the universal model with autoencoder
An intuitive way to adjust the bias of the model is walkthrough training, i.e. retraining the model with recent data. This way, the model can learn from the most recent data, and integrate it with the knowledge learnt from the original data. We implement a walkthrough with stable retrain time as follows. First, a queue buffer is set up to collect features from the live data. After every 196 predictions made by the model, the model retrains by the newly collected features in the buffer. To test the effectiveness of this modification, we use live data of BTC-USD from 2018-08-08 14:31:54 to 2018-08-09 09:01:13. The results are plotted in Figure 3.13. We observe that before the first retraining, the model lacks the predictive power on live data. It starts with an accuracy of less than 50%, which is worse than random guessing. After the first few instances of retraining, however, the model improves accuracy from 58% to 78%, to finally stabilize around 76%. Moreover, the distribution of the predictions of the model shows a similar shape to the real-time target distribution, and no apparent bias can be observed, cf. Figure 3.14. The biased training data could cause the biased model, then an intuitive to adjust the bias of the model is to retrain the model with recent data. Therefore, the model can learn from the most recent data, and integrate it with the original data. A queue buffer is set up to collect features from the live data. After every 196 predictions made by the model, the model retrains by the newly collected features in the buffer. This way of retraining helps the LSTM model adjust to the newly acquired features and retains the knowledge gained from the original training.

We have run an augmented Dickey–Fuller test (ADF) testing the null hypothesis that a unit root is present in the time series comprised of the live data samples. The results showed that the $p$-value of BTC-USD data from 2018-08-11 12:09 to 2018-08-16 23:59 is 0.781425 and the $t$-statistics value is -0.919673. We have a value of -3.431 when the
3.3 Experimental Study

Fig. 3.13 Performance of the universal model with autoencoder

Fig. 3.14 Predictions distribution and real-time target distribution
confidence level is 1%, value of -2.862 when the confidence level is 5%, value of -2.567 when the confidence level is 10%. We can see that the value is larger than the critical values in 1%, meaning that we can accept the null hypothesis and in turn conclude that the time series is non-stationary.

A further improvement of the model to work on live data is needed to improve the execution speed and reduce the chance of overfitting. This is achieved by reducing the dimension of the input data. The intermediate output of the autoencoder, which is the output of the encoder part, is used instead of using the original data. Because of the architecture of the autoencoder, the hidden layer contains fewer neurons than the output layer. Although the hidden layer contains fewer neurons, it preserves all the essential information of the input data. By using this approach, the universal model can use fewer neurons to capture the information that is needed to make predictions. Therefore, the neural network has less freedom to be overfitted, and the reduction of the size of the neural network also improves the execution speed. Our architecture uses the intermediate encoder output as the input for the LSTM model, cf. Figure 3.15. The autoencoder used in this study is also constructed by LSTM cells. The characteristic of the LSTM is that the input layer and the output layer usually have a same number of neurons, and the hidden layers of the autoencoder must have less number of neurons compared to the input and output layers. The reason for this architecture is that, the fewer neurons in the hidden layers can form a bottleneck of the neural network, so the autoencoder cannot learn by remembering the input only. This architecture forces the autoencoder to compress input data and then decompress the data before outputting it. PCA and autoencoders are both methods for dimensionality reduction and feature extraction. However, while autoencoders are generally used for non-linear data, PCA is often used for linear data. In our case, PCA can be applied to the output of an autoencoder to further reduce the dimensionality of the latent variables. It can help to identify the most important latent variables that capture the majority of the information in the data. This can be useful for tasks such as clustering or visualization, where a lower-dimensional representation of the data is desired. By combining the strengths of both techniques, it is possible to achieve better performance in some applications. So after the dimension has been reduced by the autoencoder, the PCA is put in the middle between autoencoder and the LSTM model. This architecture can further reduce the number of features.

The answer to RQ3 is summarized in Table 3.17, where we display the performance metrics of the predictive model with a reduced architecture on the same live data of BTC-USD from 2018-08-08 14:31:54 to 2018-08-09 09:01:13. Up arrow means the prediction of prices’ going up, down arrow means the prediction of prices’ going down. In the table, “autoencoder as denoiser” refers to architecture in Figure 3.10 (including encoder, decoder, PCA, universal model and output) whilst “autoencoder as reducer” refers to the architecture
3.3 Experimental Study

![Diagram of LSTM model with autoencoder](image)

Fig. 3.15 LSTM model with autoencoder

in Figure 3.15 (including encoder, PCA, universal model and output). The difference of the precision score and the accuracy score is much lower than the original model, which suggests that there is less overfitting. Interestingly, the reduced model has a 2.43% increase in the accuracy score. Compared to research of Easley et al. [149], our method gives an optimisation in LSTM structure; in particular, the machine learning model we designed is more suitable for live data prediction. Furthermore, we optimise the process of retraining, which can also be applied in multi-class prediction.

Table 3.17 Classification report

<table>
<thead>
<tr>
<th>Classification report of autoencoder as denoiser</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>↑</td>
<td>0.72</td>
<td>0.86</td>
<td>0.78</td>
</tr>
<tr>
<td>↓</td>
<td>0.81</td>
<td>0.65</td>
<td>0.72</td>
</tr>
<tr>
<td>avg</td>
<td>0.77</td>
<td>0.76</td>
<td>0.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification report of autoencoder as reducer</th>
<th>Precision</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>0.79</td>
<td>0.78</td>
<td>0.79</td>
</tr>
<tr>
<td>↓</td>
<td>0.77</td>
<td>0.78</td>
<td>0.77</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.78</td>
<td>0.78</td>
<td>0.78</td>
</tr>
</tbody>
</table>

We also collected and displayed the performance metrics in the predictive model with the latest model on the live data of BTC-USD in OkEX. We separate the live dataset into
3.3 Experimental Study

to pieces: from 2018-08-11 12:09 to 2018-08-16 23:59 and from 2018-08-24 12:07 to 2018-08-29 23:59 (cf. Table 3.18). The results are slightly worse than the ones in Table 3.17 considering Precision and F1-score, especially in down classification. This might be due to a very unstable spread in those periods; this affects the hyper-parameters of our predictive model. A better and more stable live model against the cryptocurrency fluctuations is one question left open by our work.

<table>
<thead>
<tr>
<th>Classification report of 2018/8/11-2018/8/16 (P1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
</tr>
<tr>
<td>↑</td>
</tr>
<tr>
<td>↓</td>
</tr>
<tr>
<td>avg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification report of 2018/8/24-2018/8/29 (P2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
</tr>
<tr>
<td>↑</td>
</tr>
<tr>
<td>↓</td>
</tr>
<tr>
<td>avg / total</td>
</tr>
</tbody>
</table>

What is the best Walkthrough method in the context of multi-label prediction?

Lastly, we discuss RQ4. In this section, we concentrate on multi-label prediction using the four classes identified in Table 3.5. We use the last model in RQ3 but we use multi-label as target classification and walkthrough method as a research variable. We use 1-min live data in 2018 for 1 month-window and the time interval of data is randomly selected.

Figure 3.16 gives an overview of the comparison between the three different walkthrough methods we tested. For the first method, we train the machine learning model statically without walkthrough, which means we will not retrain or update the model when the accuracy decays. We can observe how the accuracy drops significantly after roughly 1,500 predictions and reaches a value of less than 40%, almost as low as random-guessing (25%), in the end. When we use stable walkthrough to train our model, we will retrain at regular time periods. Our tests are based on trading period or a financial trading cycle of five days. On our data, this leads to four retrains (identified by rectangular points on the accuracy line). The first retrain point has obvious effects in improving accuracy and it starts to go up slightly before retraining. After four instances of retrain, the accuracy stabilizes around 80%. When we use MAD-Dynamic Walkthrough method, we retrain the original model when the accuracy drops by more than 15%. In our test, there is only one
such instance (circle point at around 2,000 mark). The accuracy has apparent growth after this model adjustment.

![Snapshot of comparison of different Walkthrough methods](image)

**Fig. 3.16** Snapshot of comparison between different walkthrough methods

We also perform the experiment for 20 times to compare the different walkthrough methods in order to have a stronger statistical guarantee; results are shown in Figure 3.18. (Considering that repeated experiments cost significant computation power and time, we repeat the experiment 20 times to gather the results.) The results are in line with those discussed above, i.e., stable walkthrough is better than the other two methods. The results also show that, for dynamic (MAD) walkthrough method, only 2/3 retraining points occur in most experiments (90%) while 2 experiments require 5 retraining points. As a comparison, for stable walkthrough method, all the experiments need 4 retraining points. Therefore, when retrain time is a factor to consider, the dynamic (MAD) walkthrough method is better because it needs less retraining in most cases. Extreme market conditions like jumps, crashes, and rebounds are other problems that could affect the performance of the dynamic walkthrough method. The experiments have been made to compare Dynamic Walkthrough methods with or without extreme conditions. The results show that the dynamic walkthrough method with extreme conditions would reduce accuracy and reveal very high fluctuations in predictive accuracy.

Dynamic or constant frequency to retrain machine learning has been widely used in academic and industrial areas. But in high-frequency modeling, the model is easy to lapse because the demand and supply conditions are quickly changed. The finer the time scale,
3.4 Validity of findings

The answer to RQ4 is multifold. Firstly, in multi-label prediction with machine learning model, walkthrough training significantly improves the prediction accuracy. The reason is that, as discussed above, machine learning prediction models need to update itself. When the model is not fit for the new market conditions, then it must be updated to achieve accurate results. Secondly, stable walkthrough method is better than MAD-dynamic walkthrough method, unless retrain time is important.

3.4 Validity of findings

The model is based on trading system; we select historical data and collect live data for a long time span. The selection of data has no bias because historical trading contains all available transaction data and available currency pairs in cryptocurrency market. Moreover, the experiments are not affected by bull or bear market, policy impact and other factors.
3.5 Conclusions

We indicate that the selected data slice had not met short-term extreme volatility (like flash crash), and the selected currency pairs in cryptocurrency exchanges had not received allegations or legal proceedings. And the selection of the datasets’ timeframe is completely random.

Quality of data is another important factor to discuss. As the data is collected live from Coinbase Pro, poor connection might affect the data (e.g., missing values). To mitigate this risk, we have compared the data collected from Coinbase Pro with other third party service providers to make sure the experiment have not been affected by inappropriate financial data.

3.5 Conclusions

This paper analyzes a data-driven approach to predict mid-price movements in cryptocurrency markets, and covered a number of research questions en route regarding parameter settings, design of neural networks and universality of the models. The main finding of our work is the successful combination of an autoencoder and a walkthrough retraining method to overcome the decay in predictive power on live data due to non-stationary features on the order book. Our results show that our model has achieved good performance, quantified in a consistent F1-score of around 78%. By comparing different retraining methods (we call that Walkthrough), we found some tradeoffs between fixed and dynamic retraining. Prediction in high-frequency cryptocurrency markets is a challenging task because the
environment contains noisy information and is highly unpredictable. We believe that our results can inform the design of higher level trading strategies and our networks architecture can be used as a feature to another estimator. One interesting direction for future research might be a more extensive treatment of how time persistent the performances of the model are, similarly to [382].

However, we must also realise that machine learning has obvious limitations, which must be overcome to reach artificial general intelligence [309]. Marcus pointed out that machine learning models are data-hungry and the knowledge gathered by deep learning systems is primarily concerned with correlations between features, rather than abstractions like quantified statements. These characteristics have negative impacts on machine learning in financial prediction. Moreover, we know that when applying out-of-sample tests in non-stationary data, the prediction made are not entirely “honest” [220]. The corresponding forecast error may underestimate the magnitude of the error that will arise when the model is used to forecast the future, as the data may overfit the squared error and the model and inadvertently fit some “noise” during the estimation. To deal with these aspects, our model uses retraining and is tested on different live time series (and perform consistently well).
Chapter 4

Dispersion of feature importance in tree models

4.1 Introduction

Variable importance (a.k.a., feature importance) represents the statistical significance of the impact of each variable in the data on the generated machine learning models [386]. Variable importance can be used to measure the increment of model prediction error after replacing the object features and, in turns, breaking the relationship between the features and the real results [77]. In our previous research, variables like price, volumes, and order flows are very important in constructing a robust deep-learning model. Easlet et al. [151] found that some microstructural features with high explanatory power exhibited low predictive power, while other microstructural features with lower explanatory power had high predictive power. In fact, models with complex structures like deep learning are hard to explain and to be controlled when people are trying to find ontogeny issues in predictive modeling. Reviewing the network structure in model building and optimising it with the aim of improving prediction accuracy makes it difficult to find the underlying factors that caused the model to fail when it does so again. So in important application scenarios (e.g. important financial credit or medical diagnostics), prediction tasks using simple or highly interpretable machine learning models like tree models are more popular. Tree models are divided into black tree models (hard to explain to humans) like RandomForestTree and white box model tree models (easy to explain) like DecisionTree [364]. Compared to other complex black box models, such as neural networks, it is relatively easy to understand (and explain) the contributions that each variable makes to the decision of tree models [323]. Furthermore, “features” are the core hyper-parameters in all tree models, which means it is important to combine “features” with model selection and/or model explanation in
4.1 Introduction

As from above, the importance of the features is measured by the increase of the prediction error of the calculated model after arranging the features. If shuffling the values of a feature increases model error, the feature is considered as “important” [170]. This method gives feature importance an interpretation that it is the increase in model error when the feature’s information is destroyed [324]. Related research in model selection usually involves appropriate criteria, usually based on an estimate of the generalization error, such as k-fold cross validation [313]. Other complex model selection methods, like model selection using combinatorial optimisation and genetic algorithms have also been proposed [48]. There are various types of famous explanation-based technology for machine learning like Lime [298] and Shap [390]. Lime explains the prediction results of machine learning models by generating interpretable local models, and Shap is a game-theoretic-based interpretation technique used to measure the contribution of each feature to the prediction results. They can help us understand how the model makes predictions and help us identify potential problems in the model. But it is not enough to rely on these two techniques for interpretable analysis of tree models. Firstly, both models rely on the quality of the input data, and unbalanced classification or data noise can lead to a failure in the interpretation of the model. They are more of a contribution or explanatory analysis for the case, and do not allow us to filter the best model for the current set of models more quickly through this explanatory analysis. This subsection goes some way towards optimizing the speed of model selection with some explanatory power while ensuring the quality of selection.

The performance of machine learning (ML) models is limited by the quality of the data. While researchers and practitioners have focused on improving the quality of models (e.g. neural architecture search and automatic feature selection), limited efforts have been made to improve the quality of data [222]. The important perspective is that we should have a balance between dataset and models, which means improving the degree of generalizability of the model. Model selection is the task of selecting a better model from a set of candidate models for given data. Cross validation is arguably the most used technique for estimating the risk of a machine learning model or performing model selection [21]. But cross validation through data splitting provides little additional information during the evaluation of the model and, importantly, costs a long time in retraining the model [263]. Although cross-validation achieves a near optimal variance reduction factor of $(1 + o(1))k$ [236], it does not take full advantage of the characteristics of the data considering features. We implied the definition **Utilisation rate of feature** to measure the efficiency of models’ training and optimisation of hyperparameters tuning using FEATURES data. The efficiency can be embodied in the dispersion of feature importance. When FEATURES data is feed

---

tree models. We call these dataset received prepossessing and with segmentation set of FEATURES characteristics as **FEATURES data**.
into the model, the model is difficult to deal with extreme situations due to features specificity when the feature importance has low dispersion, which reflected that the FEATURES data have similar optimisation directions for the model (and vice versa when feature importance has high dispersion). We will prove the point above with experiments in this subsection. And this characteristics will be extended to model selection. The brand new model selection method better uses the property of dispersion in feature importance, which demonstrates good model generalisation and time superiority. We here initiate this research by looking at tree models.

This research aims at revealing the characteristics of utilisation rate of features using feature importance dispersion when we perform hyperparameter tuning in tree models. The findings are then utilized in model selection to reinforce our common model selection framework. Two forms of calculation of feature importance dispersion are proposed including Coefficient of Variation[79] and Entropy values. Detailed, in method “Coefficient of Variation”, we design an estimation of tree-structure models (including DecisionTree, RandomForest, ExtraTree, GradientBoostingTree and XgboostTree) considering weighted contribution of features, the performance of contribution and feature quantity (i.e., the number of the features) to measure the utilisation of features. And in method “Entropy Values”, we use the definition of entropy in information theory to calculate the value of information gain in new data information based on features, which is an estimation of utilisation of features in our tuning models.

In summary, this subsection makes the following contributions:

- This novel conceptual contribution connects common feature weighting techniques with the dispersion degree matrix. The subtle effect of features on the generalisation of the model shows that there is a strong relationship between the distribution of feature importance and the classification prediction performance of the model.

- This paper uses Feature Contribution Dispersion notion to define a new pipeline (two measurements to calculate the dispersion extent) for model selection as opposed to cross validation method.

- The highlight of Feature Contribution Dispersion method is that the optimised pipeline can generally maintain the performance of cross validation (in terms of test accuracy as generalisation) whilst reducing computation time by at least a third.

### 4.2 Related Work

A large body of recent research has been devoted to the estimation of the performance of a machine learning model, including (i) estimating the generalization performance of models
on future data; and, (ii) selecting the best performing model from a given hypothesis space [356]. Basic selection/evaluation methods like Resubstitution validation, Stratified resampling and Holdout validation were proposed to be effective in selecting “good” models. With the development of machine learning, a large numbers of settings (hyperparameters) need to be specified. Hyperparameter tuning allows to find the balance between bias and variance when optimizing the performance of these models. Cross validation is a great improvement based on holdout method in evaluating hyperparameters [255]. Cross validation was proposed to help to select models with a better (average) generalization than just relying on the training score [368]. In particular, in $k$-fold cross validation (aka, repeated hold-out method) the data is split in $k$ chunks and each chunk is used for testing the model trained on the remaining $k - 1$ folds. Models selected by this method generally get results that are less biased and less optimistic than other methods.

As the core definition in machine learning research, “features” refer to an multi-dimensional vector representing the (numerical) characteristics of an object. They are core to many fields of machine learning research including dimension reduction, relevance research, automation and model explanations [434]. In Explainable AI (XAI), features are a medium for humans to understand the machine learning models that are hard to explain (commonly known as “Black box models”) [354]. Feature interaction is a method to explain models by understanding whether features affect each other and to what extent they interact. Variable Interaction Networks are a tool proposed to decompose the prediction function into main effects and feature interactions and then visualize those as a network [212]. Partial dependence based feature interaction is applied in measuring the feature importance by calculating the variance of the partial dependence function [194], which illustrates and explains interaction among features in machine learning models. But feature interaction is computationally expensive, and if we do not use all of the data points, the estimate has a non-negligible variance.

Permutation feature importance (PFI) is a concept to parse models by calculating the increase of model prediction error after the feature values are permuted. The PFI method was introduced by Breiman [77] for random forest first. Based on idea of PFI, a model-agnostic version of the feature importance was proposed (called model reliance) [170]. The PFI method considers both the influence of the main feature effect and the interaction effects on the performance of the model. But this approach has obvious drawbacks. Notably, if the features are related, the ranking of the importance of the features may be biased by unrealistic data instances. Moreover, the importance of the associated feature might be decreased by adding a correlated feature.

Recently, a new interpretability of tree-based models features is proposed. Lundberg et al. [297] proposed to improve the explanation of tree along three dimensions: polynomial time algorithm for optimal interpretation of time based on game theory, direct
4.3 Methodology

4.3.1 Feature importance in Tree Models

Variable (Feature) importance describes the contribution of covariates to the prediction and model accuracy [170]. While this works well in linear models, tree models have different variable importance calculation. Linear models’ variable importance and explanation use loss functions which map the value of one or more event/variables to a real number that intuitively represents the “cost” associated with the event to optimize the original linear models. Tree-structure models have specific realization, which separates “nodes” and “edges”. Moreover, the splitting process will continue until no further “revenue” can be obtained or the preset rules are met [180].

The variable importance of tree-structure models is calculated by Mean Decrease in Impurity (MDI) [294] of the node, and the probability of the impurity reaching the node is obtained. The node probability can be calculated by dividing the number of samples arriving at the node by the total number of samples. The higher the value, the more important the node. But experience tells us that through this method explanation is hard to understand and it is difficult to link decisions with insights into actual data. An alternative method is to iterate through all the splits that use this element, and measure the degree to which variance or Gini coefficient is reduced compared to the parent node. The sum of all importance is scaled to 100. This means that each importance can be explained as part of the overall model importance [323].

4.3.2 Prediction Path based feature importance

Researchers consider using decision paths to explain tree-structure models, which consist of paths (decisions) from the root of the tree to the leaf (outcome). Every decision path contributes to the final prediction [199]. This decision function returns a value at the correct leaf of the tree, but ignores the operational aspect of the decision tree, namely the path through the decision node and the information available there. Since each decision path is determined by features, and the decision will be added or subtracted from the value given in the parent node, the prediction can be defined as the sum of feature contributions.
4.3 Methodology

plus "bias" covering the entire training set. This feature-based concept is the core method in selecting tree-structure models.

As defined in Tree-Interpreter library in Python, the prediction function can be defined as

\[ f(x) = c_{full} + \sum_{k=1}^{K} contrib(x, k) \]  

(4.1)

where \( K \) is the number of features, \( c_{full} \) is the value at the root node calculated according to information gain and \( contrib(x, k) \) is the contribution from the \( k \)-th feature in the whole dataset \( x \). In tree-structure models, contribution of each feature depends on the rest of the dataset, which determines the decision path of traversing the decision tree, thereby determining the protection/contribution passed along.

We use an example to explain how the path-based prediction could measure the feature importance through the shape of trees. When making a specific prediction, a decision tree or random forest follows a specific path to make a prediction. Each node in the decision tree represents a certain feature and makes a decision based on the value of the feature in the sample. The space of tree interpreter prediction regions is divided into regions with the same number of leaves present in that tree. At each internal node in the tree, the prediction value will be the average of all possible predictions in the data from the path through that node. We will also get the average value at the root node, which will be the average of all predictions. This way we can get some prediction value at each node in the tree. This tree interpreter uses these values to find out the contribution of each feature in the prediction by finding out the difference between the prediction of a particular node and the node in its previous path.

In Figure 4.1, the “RM”, “LSTAT”, and “DIS” are specific features that may affect the output. We start with a base value of 22.60 and subtract 2.64 (because the value of the feature RM is less than 6.94) to arrive at a predicted value of 19.96. We then add 3.51 to 19.96 to arrive at a predicted value of 23.47 because the value of the LSTAT feature in the sample is less than 14.40. We then add 22.12 to the previously predicted value of 23.47 to arrive at a final predicted value of 45.59 because the value of the feature DIS in the sample is less than 1.38. This allows us to start from a base value of 22.60 and then add values based on feature contribution. 22.12 to the previous prediction value of 23.47, resulting in a final prediction value of 45.59 because the value of the feature DIS in the sample is less than 1.38. This allows us to start with a base value of 22.60 and then increase the value based on the contribution of the feature [199]. By then we could get the contribution of features through the prediction path.
4.3 Methodology

We build upon (4.1) as follows. The contribution \( f(x,k) \) from the specific feature \( k \) on input \( x \) can be defined as

\[
\begin{align*}
  f(x,k) &= c_{full} + \text{contrib}(x,k).
\end{align*}
\] (4.2)

Tree-structure models use some hyper-parameters like “depth” and “max leaf” to control the complexity of tree models, given data set with features. To some extent, these parameters help tree models realize dimension reduction of decision rules to improve accuracy of the classifier, which is feature selection [250]. Meanwhile, the quantity of features affects the tree-structure models formation, which promoted the formation of the distribution of feature importance matrix when the model optimisation completes.

4.3.3 Two methods to measure utilisation rate of features

When we feed the FEATURES data into tree models, features affect the direction of optimisation of the model in different directions and to different degrees. The values of feature importance have different distribution by this effect. Some explainable machine learning techniques takes advantage of this characteristics to evaluate feature correlation for highly complex or non-parametric models and provide interpretability for machine learning models [17]. If the dataset is mature and trusted when the features are classified reasonably, these models must have the ability to use dataset more efficiently in terms of FEATURES, which we call it **Utilisation Rate of Features**. As can be seen in the upper
part of Figure 4.2, horizontal slicing of dataset contains the information of known data. In machine learning, we use several validating techniques to prevent over-interpretation and lack-interpretation of dataset (we call them overfitting and underfitting), normally with CrossValidation tool [80]. In the longitudinal perspective of data slicing, these features Optimise model updates driven by the features perspective of the data. Especially in tree models, the features play important parts of building logical judgements in the tree(s). Utilisation rate of features help us get the in-depth knowledge of models’ generalisation of features.

![Diagram of Feature Slicing and Cross Validation](image)

Cross Validation: (10-fold)  
Get datasets in slice dimension

$$E = \frac{1}{10} \sum_{i=1}^{10} E_i$$  
* refers to slice pieces (10 means to split dataset into 10 pieces)

Feature Explanation:  
Get datasets in slice and feature dimension

$$Dispersion(f) = \sum_{k=1}^{K} \left( f(x,k) - \frac{1}{K} \right)^2$$  
* X in Feature explanation refers to the whole dataset

Fig. 4.2 Comparison between cross validation and feature contribution dispersion

Our hypothesis is that the **degree of utilisation rate of features, reflected in the dispersion of features’ contribution or the characteristics of features may affect the performance of tree models’ selection when we tune the hyperparameters to optimise models’ parameters**. In other words, we must combine the performance of features (feature importance/contribution) with complexity (quantity) of features when we select tree-structure models. We design two methods in measuring this: Coefficient of Variation and Entropy Values of feature importance. And these methods are applied in more stable and faster tree models selection when we use hyperparameter tuning to select models.
4.3 Methodology

4.3.4 Coefficient of Variation of Feature Importance

Firstly, we use the contribution of features, which we called the weight of feature $k$ for input $x$ in the model to define the feature importance

$$\text{weight}(x,k) = f(x,k).$$

Where the $f(x,k)$ is the feature importance from the specific feature $k$ on input $x$ which is defined in formula 4.2. We then consider using Coefficient of Variation [79] as dispersion of model $f$ with feature $k$ contribution, and obtain

$$\text{Dispersion}(f) = \frac{\sum_{k=1}^{K}(\text{weight}(x,k) - \overline{\text{weight}})^2}{K \cdot \overline{\text{weight}}},$$

where $\overline{\text{weight}}$ is the mean of weights from 1 to $K$, i.e., $\overline{\text{weight}} = \frac{1}{K} \sum_{k=1}^{K} \text{weight}(x,k)$, and $K$ is the number of features. From the definition of $\text{weight}(x,k)$, the sum of features’ importance is equal to one, so we know that

$$\sum_{k=1}^{K} \text{weight}(x,k) = 1.$$

We assume that the importance of these $K$ features is uniformly distributed. Firstly, we lack a priori knowledge. In some cases, we may not know the true distribution of the features, or we may not have sufficient a priori knowledge to determine the distribution of the features. In such cases, assuming that the features obey a uniform distribution can be a reasonable choice, as it is a more neutral assumption and does not introduce too much subjectivity. Secondly, the assumption could reduce overfitting. Some features may overfit the model, and assuming that all features obey a uniform distribution reduces this risk. This is because uniformly distributed features have a relatively balanced effect on the model and do not rely too heavily on particular features. Overall, We then get

$$\text{Dispersion}(f) = \sum_{k=1}^{K} \left( f(x,k) - \frac{1}{K} \right)^2.$$

(4.3)

Eq 4.3 is defined as the degree of dispersion in features’ contribution of tree-structure model $f$. In this equation, $f(x,k)$ is the weight of specific feature $k$ in the construction of tree-structure models; $K$ is the number of features used in the model and $x$ refers to the whole input dataset while $(x,k)$ is the dataset with selected feature $k$. Eq 4.3 is derived from coefficient of variation of feature weights to the entire tree model. From the theoretical analysis, Coefficient of Variation measures the dispersion of data point around the mean [79]. When we apply the Coefficient of Variation to the weight of the specific
4.3 Methodology

feature, we weigh the contribution of the feature, the performance of the contribution and quantity. And the weight of features is estimated according to path-based feature contribution which is introduced in Section 4.3.2. What we consider is whether the original features combined with the coefficient of variation has better performance in selecting tree-structure models.

Considering the characteristics of the degree of dispersion in features’ contribution we defined, we design the workflow in model selection using cross validation and feature contribution dispersion method (depicted in Figure 4.3). In particular, when compared to the state of the art, we compare cross validation with feature contribution dispersion in model selection using Pareto optimality (see below for a definition). The model selection of Cross validation method combines the Training performance and cross validation performance to select models while the feature contribution dispersion method combines “Dispersion” values and Training accuracy to select models with good generalisation. Cross validation costs a lot of time in model selection because it needs to train data set in a loop, which is computationally expensive. More importantly, feature contribution dispersion has the function of choosing best parameters considering the training accuracy, features’ contribution and feature complexity (quantity).

Figure 4.2 describes the comparison between cross validation and feature contribution dispersion. We use 10-fold cross validation as an example. The data sampling considers slices in (row) dimension. Considering the cross validation has 10 fold iterations and each iteration has an evaluation accuracy $E_i$, the final cross validation result is $E = \frac{1}{10} \sum_{i=1}^{10} E_i$. Each iteration needs a new training for original model. When applying feature contribution dispersion method, We use feature contribution of tree models combined with complexity.

![Fig. 4.3 Workflow in Model selection using Cross validation and Feature Contribution Dispersion](image-url)
of features (which is the numbers of features we used \( K \)) to represent the characteristics of the dataset. This method does not need a retrain of model we have trained before.

### 4.3.5 Entropy Values of Feature Importance

In information theory, the **entropy values** of a random variable is the average level of "information", "surprise", or "uncertainty" inherent to the variable’s possible outcomes. The evaluation of entropy values is behind quantifying information, which believes that those events with low probability are more surprising and therefore have more information than those with high probability [41]. This is in line with our vision of assumptions of the utilisation rate of features, considering models with different hyperparameters and same FEATURES data.

Firstly, we define the weight of feature \( k \) for input \( x \) in the model (which is calculated by path-based feature contribution in Section 4.3.2) as

\[
w_k = \text{weight}(x, k) = f(x, k), \quad k = 1, \cdots, N,
\]

where \( N \) is the number of features in input \( x \), \( f(x, k) \) is defined as (4.2) which is calculated by "Tree Interpreter", and every feature weight \( \{w_1, \cdots, w_k, \cdots, w_N\} \in W \) (\( W \) refers to collections of all feature sets). We then consider the entropy values of all features’ importance as the dispersion evaluation of the feature importance of model \( f \).

K-fold is a good way to validate the robustness of the model with dataset. Repeating k-fold cross-validation can increase the precision of the estimates while still maintaining a small bias [356]. We apply similar K-fold split of dataset in calculating \( w_k \). After optimising the process, we obtain uniformly distributed feature importance matrix for training/test fold, see Figure 4.4. The average entropy value is calculated by averaging the entropy values in the K-fold data split. After all, we could get the fair entropy value of the features’ importance when training the tree model.

We assume that the feature importance entropy gives insight into selecting tree models, which could significantly reduce the time we consume in hyperparameter traversal. If we could find there exits relationship between the model’s generalisation (accuracy et al.) and feature importance entropy, we could use this relationship to simplify the process of our hyperparameter tuning. Thus, we design the workflow in finding the relationship between model generalisation criteria (test accuracy) and feature importance entropy (depicted in Figure 4.5).
4.3 Methodology

Fig. 4.4 Process of calculating k-fold feature importance

\[ \text{Average Entropy Values} = \frac{1}{k} \sum_{i=0}^{k} \text{Entropy Values}_k \]

Fig. 4.5 Workflow in finding the relationship between Model generalisation and Feature importance Entropy values
4.4 Experimental Study

We evaluate the performance of our feature contribution dispersion method using tree-structure models with standard data sets. In the experiment study process, we adopt the workflow in Figure 4.3 and Figure 4.5. Our results are evaluated by test accuracy and time efficiency according to the models we selected. We considered $k$-fold cross validation as a benchmark method to optimise/apply our feature utilisation rate (feature contribution dispersion) method.

4.4.1 Experimental Setup

The tree-structure models in this research include Decision Trees, Random Forests, Extra Trees, Gradient Boosting and XGB Classifiers. Hyper-parameters used in these models are through parameter tuning [273] with available parameters in models. For tree-structure models, important features as “max depth”, “min sample leaf” and “criterion” are included; “max features” are limited by quantity of features. All hyper-parameters used in certain model are selected from hyper-parameter tuning and features in hyper-parameter tuning are obtained in an interval of values (opportunely defined as from [308]).

4.4.2 Datasets

We list details of data sets we have used in the experiment (cf. Table 4.1) for binary classification. As the scientific characteristics of features composition and division might effect the results of feature importance values, we select relatively formal and standardised data sets. We choose some classification datasets encompassing different areas, complexities and data size. The datasets come from an open science online platform for machine learning “OpenML (Open Machine Learning)”. The standard data sets contains EEG state Data set, Chess Data set [372], QSAR biodegradation Data Set [307], Steel Plates Faults Data Set [84], Phoneme Data Set. The dataset will be split into training dataset and test dataset (0.5/0.5) when using feature contribution dispersion as a method. When using cross validation, we will further split 20% of training dataset for validation.

4.4.3 Evaluation Criteria

In this research, we focus on using feature contribution dispersion in model selection. When we compare performance of cross validation based selection and model selection by feature contribution dispersion, time efficiency and accuracy (ACC) performance are two factors we consider. As usual, after we trained the tree models using parameter tuning...
Table 4.1 This table shows the property of data sets used in experimental study. The #Classes, #Instances, #Features and #Attribute present the number of classes, instances, features and attribute characteristics, respectively.

<table>
<thead>
<tr>
<th></th>
<th>#Classes</th>
<th>#Instances</th>
<th>#Features</th>
<th>#Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>openML1471</td>
<td>2</td>
<td>14980</td>
<td>14</td>
<td>Integer, Real</td>
</tr>
<tr>
<td>openML3</td>
<td>2</td>
<td>3196</td>
<td>36</td>
<td>Categorical</td>
</tr>
<tr>
<td>openML1494</td>
<td>2</td>
<td>1055</td>
<td>41</td>
<td>Integer, Real</td>
</tr>
<tr>
<td>openML1504</td>
<td>2</td>
<td>1941</td>
<td>33</td>
<td>Integer, Real</td>
</tr>
<tr>
<td>openML1489</td>
<td>2</td>
<td>5404</td>
<td>5</td>
<td>Real / Integer / Nominal</td>
</tr>
</tbody>
</table>

with training data sets, we will use test data sets to evaluate the accuracy performance of selecting models according to accuracy matrix.

We noticed that several tree-based methods allow for out-of-bag estimates of error especially for Random Forest model, which is a well-established and fast method for evaluating generalisation performance. We use out-of-bag (OOB) estimates of error as an evaluation criteria in some cases as a comparison. We found that in general OOB performs similar with cross validation in model generalisation and OOB is limited in other tree-structure models except Random Forests. Thus we use cross validation and feature contribution dispersion as base model generalisation methods.

4.4.4 Research Questions

To evaluate our method and compare it to cross validation in model selection, we explore the following research questions (RQs, for short):

**RQ1.** What is the relationship between feature contribution dispersion calculated by Coefficient of Variation and accuracy matrix in model selection? How can we apply this method to select tree models?

We know that Coefficient of Variation based feature contribution dispersion method evaluates tree models from training accuracy, features contribution and model complexity. We need to experimentally evaluate what is the relationship between feature contribution dispersion and (test) accuracy. Depending on the empirical findings about the relationship, we could apply feature contribution dispersion method in tree models selection.

**RQ2.** What is the effect to apply Coefficient of Variation in feature contribution dispersion method to select tree models compared to $k$-fold cross validation?

Applying the results from RQ1 about selecting tree models using feature contribution dispersion, we need to experimentally compare the performance of our method with
4.5 Experimental Results

$k$-fold cross validation. The evaluation will look at accuracy performance and computational cost (time).

**RQ3.** How the entropy values of feature importance would change when we optimised the tree models? What the relationship between tree models’ generalisation and entropy values of feature importance? Can we narrow the hyper-parameter tuning space when we applying this relationship? Can we combine entropy values in feature contribution dispersion method with Crossvalidation method in selecting tree models more efficient?

We know that entropy value is a measurement to evaluate distribution (dispersion) of feature importance when tuning these tree models. We need to evaluate the difference between entropy of feature importance when simulating tuning the tree models. When comparing accuracy performance and computational cost, it is possible to use or partial use this criteria with Crossvalidation in selecting models during hyperparameter tuning.

### 4.5 Experimental Results

In this section, we present the results of the experimental study, and interpret the research questions sequentially and separately to explain how these two proposed approaches in dispersion of feature importance can be applied in traditional $k$-fold cross validation to optimise the model selection.

#### 4.5.1 RQ1: What is the relationship between feature contribution dispersion and accuracy matrix in model selection?

As discussed above, feature contribution dispersion method parses tree models from training accuracy, features contribution and model complexity. When we apply feature contribution dispersion in model selection, we experiment the relationship between dispersion values and accuracy matrix using binary classification datasets. We have listed three of the results of relationship between feature contribution dispersion values and accuracy matrix in Figure 4.6 from experiments of Decision tree model. Each point in the relationship figure refers to a model evaluation based on hyperparameter tuning. In each experiment we present two figures, plotting the relationship between train accuracy and cross validation accuracy (CV accuracy in figure) or Dispersion value (we defined in Eq 4.3). We use “test accuracy” as the evaluation criteria, which is showed using the colour bar on the right. The left part of the figure shows that combining Training performance and cross validation performance, we could choose “better” models by using Test Accuracy as
a criteria matrix. At the same time, we find performance of training set is a core/startup method in selecting models with good generalisation. By optimising cross validation and Training performance at the same time, we could select models with better generalisation; we will discuss this bi-objective optimisation strategy later. The right part of figure shows that when the value of feature contribution dispersion is smaller, the model is likely to achieve a higher test accuracy. But this regularity shows a large margin of error when the Training set performance is not good. In Figure 4.6, we can notice some models (points) with bad training accuracy, low feature contribution dispersion values and bad Test accuracy. But if we could optimise feature contribution dispersion values and Training accuracy together, it might be a good method in selecting models with good generalisation because features parsing process saves much time than cross validation or retraining these models.

Figure 4.15, 4.16, 4.17 and 4.18 are provided for completeness; the plots show more results for the relationship between train accuracy and cross validation accuracy in performance of model generalisation. In most cases, we find that low values of feature contribution dispersion lead to higher test accuracy (which means “better” models have been found) on the performance level of all hyper-parameter tuning model. From the point distribution level, cross validation and feature contribution dispersion appear to aggregate when selecting models with high test accuracy, which might be because cross validation and feature contribution dispersion have similar learning effects of features’ characteristics.

The results show that compared to cross validation, in the majority of cases, the value of feature contribution dispersion in tree models show an inverse relationship with test accuracy in sparse / dense distribution. When we have a high test accuracy, feature contribution dispersion method and normal cross validation method have similar dense distribution. It is interesting to combine Train accuracy with cross validation / feature contribution dispersion method and apply them in model selection.

4.5.2 RQ2: What is the effect to apply our CV based feature contribution dispersion method in model selection compared to $k$-fold cross validation?

To select models with good generalisation performance, we should not focus on cross validation only. Normally we should focus on training set performance and validation set performance together, meanwhile, training set performance is the dominant concern because the training set is larger and influences the choice of the model. Here we start a “two-step” model selection. First part we select those “Pareto optimal” models. By using Pareto optimal method, we are trying to find undominated solutions when we set Cross validation accuracy and Training set Accuracy as optimisation objectives. In the left part
4.5 Experimental Results

Fig. 4.6 Comparison between feature contribution dispersion and cross validation evaluating in accuracy matrix for the following tasks/models: (a) OpenML 1471 dataset and DecisionTree, (b) OpenML 1494 and DecisionTree, (c) OpenML 3 dataset and DecisionTree, (d) OpenML 1504 and DecisionTree, (e) OpenML 1489 and DecisionTree
of Figure 4.6, we call the bold red dots “Pareto optimal” points, which means that those models are Pareto optimal (i.e., it is not possible to improve both Cross validation accuracy and Training accuracy). Similarly, we select “Pareto optimal” models by training accuracy and feature contribution dispersion values. It is worth noting that we need to minimise the feature contribution dispersion values according to our findings in RQ1. The right part of Figure 4.6 selects “Pareto optimal” models with training accuracy and feature contribution dispersion values, which are marked bold red dots. We notice that some red points in the figure have high CV accuracy (low feature contribution dispersion value) with low training accuracy and, furthermore, low test accuracy. The condition is caused by our Pareto optimisation. When we use this method to select models, we focus on optimising training set performance and cross validation set performance (or training set performance and feature contribution dispersion values) together. But in fact, training accuracy performance has a more significant share in model selection (dominated by Training set performance). So we further sort these red points (models) by Training set performance again. These models with higher training accuracy after Pareto optimisation process are models with better generalisation and are selected at the end of this stage.

To evaluate the performance of Pareto optimisation in model selection, we simulate the basic model selection on five openML binary classification datasets (cf. Table 4.2). In general, the majority of the experiments show that FD method could select models with better generalisation than CV method. The results also show that FD method saves at least 300% of execution time when selecting models in tree-structure models in binary classification (average around 400% to 500%, and could achieve more time efficiency in XGB models). Time efficiency of feature contribution dispersion is because the comparison of mechanism between CV and FD (cf. Figure 4.2). In model selection, the program needs to retrain and evaluate the model up to ten times (in 10-fold cross validation), which costs too much time. In feature contribution dispersion method instead, models that have been trained will be analysed by features contribution according to models’ formation and features’ contribution.

The results from the experiment of CV based feature importance dispersion method show that feature contribution dispersion performs at least as well as general cross validation method (k-fold) in tree-structure model selection while feature contribution dispersion method has a notable advantages in time efficiency when applying binary classification task. The selected model under feature contribution dispersion and cross validation also appear similarity in generalisation. It means that we can safely replace general cross validation with feature contribution dispersion method in tree-structure model selection in the vast majority of cases when applying Pareto Optimisation.
Table 4.2 This table compares performance of model selection from 10-fold cross validation method (referred to as CV) and feature contribution dispersion (FD) method, using test accuracy as evaluation criteria. “Test Acc” means the average test accuracy from the best model selected from cross validation method (CV) and feature contribution dispersion (FD) method based on Pareto optimisation. “Hyper space” means hyperparameters combination types in hyperparameter tuning for model selection. We try to keep the hyper space as consistent as possible but we do adjust it according to the number of features. “Exec time” refers to the execution time when applying CV and FD method in model selection respectively (including the whole process of Pareto optimisation). “Best test Acc” means the average test accuracy from the best model selected from cross validation method (CV) and feature contribution dispersion (FD) method, using test accuracy as evaluation criteria. Two numbers in the brackets represent the rank of CV/FD of all models in hyper space according to the test accuracy.

<table>
<thead>
<tr>
<th>OpenML</th>
<th>Test Acc (CV/FD)</th>
<th>DecisionTree</th>
<th>RandomForest</th>
<th>ExtraTree</th>
<th>GradientBoostingTree</th>
<th>XgboostTree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1471</td>
<td>71.72% / 71.72%</td>
<td>75.95% / 75.99%</td>
<td>71.74% / 71.74%</td>
<td>75.95% / 75.95%</td>
<td>88.26% / 87.60%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>exec time (s)</td>
<td>61.19 / 22.93</td>
<td>15537 / 2996</td>
<td>2262 / 318</td>
<td>3689 / 745</td>
<td>7983 / 110</td>
</tr>
<tr>
<td></td>
<td>hyper space</td>
<td>1260 / 720</td>
<td>720 / 720</td>
<td>720 / 720</td>
<td>720 / 720</td>
<td>1450 / 450</td>
</tr>
<tr>
<td>3</td>
<td>72.08% (2/2)</td>
<td>76.06% (11/11)</td>
<td>71.74% (1/1)</td>
<td>76.13% (2/2)</td>
<td>88.26% (1/34)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>test Acc (CV/FD)</td>
<td>92.49% / 92.49%</td>
<td>94.43% / 94.43%</td>
<td>68.56% / 68.56%</td>
<td>95.49% / 95.49%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>exec time (s)</td>
<td>6.14 / 5.91</td>
<td>7888 / 1269</td>
<td>935 / 121</td>
<td>325 / 70</td>
<td>2407 / 36</td>
</tr>
<tr>
<td></td>
<td>hyper space</td>
<td>1125 / 1125</td>
<td>1125 / 1125</td>
<td>1125 / 1125</td>
<td>1125 / 1125</td>
<td>1125 / 1125</td>
</tr>
<tr>
<td>1494</td>
<td>75.76% / 75.76%</td>
<td>85.04% / 85.04%</td>
<td>82.95% / 82.95%</td>
<td>84.28% / 84.28%</td>
<td>83.33% / 83.33%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>exec time (s)</td>
<td>5.32 / 5.06</td>
<td>8725 / 1555</td>
<td>605 / 102</td>
<td>292 / 61</td>
<td>1559 / 26</td>
</tr>
<tr>
<td></td>
<td>hyper space</td>
<td>1260 / 720</td>
<td>720 / 720</td>
<td>720 / 720</td>
<td>720 / 720</td>
<td>540 / 540</td>
</tr>
<tr>
<td>1504</td>
<td>83.71% (283/283)</td>
<td>85.79% (29/29)</td>
<td>84.28% (10/10)</td>
<td>85.42% (31/31)</td>
<td>86.36% (475/475)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>test Acc (CV/FD)</td>
<td>97.12% / 97.12%</td>
<td>95.57% / 95.57%</td>
<td>99.89% / 99.89%</td>
<td>89.08% / 89.08%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>exec time (s)</td>
<td>6.96 / 5.08</td>
<td>9762 / 1484</td>
<td>658 / 107</td>
<td>389 / 64</td>
<td>1069 / 26</td>
</tr>
<tr>
<td></td>
<td>hyper space</td>
<td>370 / 2160</td>
<td>2160 / 2160</td>
<td>2160 / 2160</td>
<td>2160 / 2160</td>
<td>1350 / 1350</td>
</tr>
<tr>
<td>1489</td>
<td>97.12% (1/1)</td>
<td>96.08% (12/12)</td>
<td>100% (1/18)</td>
<td>89.70% (7/7)</td>
<td>100% (1/1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>test Acc (CV/FD)</td>
<td>91.86% / 79.42%</td>
<td>84.20% / 84.20%</td>
<td>81.61% / 81.61%</td>
<td>88.12% / 88.12%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>exec time (s)</td>
<td>9.57 / 4.98</td>
<td>11046 / 1932</td>
<td>462 / 76</td>
<td>427 / 93</td>
<td>2291 / 44</td>
</tr>
<tr>
<td></td>
<td>hyper space</td>
<td>540 / 540</td>
<td>540 / 540</td>
<td>540 / 540</td>
<td>540 / 540</td>
<td>540 / 540</td>
</tr>
<tr>
<td>1489</td>
<td>82.12% (2/42)</td>
<td>94.31% (11/11)</td>
<td>81.97% (2/2)</td>
<td>85.86% (1/1)</td>
<td>88.19% (4/4)</td>
<td></td>
</tr>
</tbody>
</table>
4.5 Experimental Results

4.5.3 RQ3: How the entropy values of feature importance can be applied in narrowing parameter tuning space and selecting tree models efficiently?

As discussed in former chapter, maximum feature importance entropy is the principle when we consider the specific-parameter-model construct exact statement with prior information (which is FEATURES data). In the experiment, we evaluate our models by hyperparameter tuning by Training accuracy, Test Accuracy, CrossValidation Accuracy and entropy values of feature importance. According to the principle of maximum feature importance entropy, we sort the results according the entropy values of feature importance (cf. Figure 4.7). And we check the performance in experimenting other datasets, the results show that when we sorting the tuning models according to entropy values of feature importance, the model generalisation face rapid decline. The phenomenon might be caused by the explosively increase of entropy values, which can be captured at the tail of entropy value line in the Figure 4.7. The results can also prove that the principle of maximum feature importance entropy does not apply throughout the hyperparameter tuning process, especially when the entropy values are extremely high. As this phenomenon is not very clear in Figure 4.7, we scaled this process by refinement the hyperparameter tuning process (Detailed, increasing the space for taking values of hyperparameter “min sample split” and “max depth” to slow down the learning rate). By this adjustments, the tuning experiment is implemented in large hyperparameter space.

Fig. 4.7 Relationship between entropy values of feature importance and model generalisation (sorted results of tuning by the entropy values of feature importance) [Dataset: OpenML 1471 & OpenML 1504 and DT model]

Figure 4.8 and Figure 4.9 perform the density plot of entropy values of feature importance, train accuracy and test accuracy with sorted by the entropy values of feature importance. The experiment owns over 70 million hyperparameters in DT model. The two figures show that the train accuracy and test accuracy appear abrupt decrease when the entropy values of feature importance explosively increase.
4.5 Experimental Results

Fig. 4.8 Density Plot of entropy values of feature importance (sorted results of tuning by the entropy values of feature importance) [Dataset: OpenML 1471 and DT model]

Fig. 4.9 Density Plot of Train accuracy and Test accuracy (sorted results of tuning by the entropy values of feature importance) [Dataset: OpenML 1471 and DT model]
4.5 Experimental Results

Figure 4.10 scatters entropy values and train/test accuracy. From the left part of the figure we can clearly observe the explosively increase of entropy values while the models have a precipitous fall in model generalisation. We can also observe that when the entropy value is close to zero, the models (especially according to the training accuracy) are likely to have local optimum generalisation, but these models will not include a globally optimal model. But according to more hyper-parameter tuning experiments by different dataset, the model have better generalisation when entropy values are higher, except for explosive increase of entropy values at the tail (cf. Figure 4.11).

Notice the relationship between entropy values of feature importance and model generalisation in above experiment, we can use this property to optimise the process of selecting models by hyperparameter tuning. It is known that CrossValidation is a normal method to select models with good generalisation, but it is computational expensive. We can use entropy values of feature importance to partial substitution the function of CrossValidation in model selection. Another way to phrase our method is that applying entropy values of feature importance could narrow the hyper-parameter space for selecting models, and this process is risk-free. We call the process of removing these decision tree models with bad generalisation before the overall CrossValidation “Model Pruning”.

As can be seen in Figure 4.12, we split three zones in pruning DT models according to empirical results above. Zone1 is the models’ group without efficient learning by the FEATURES dataset feed, but local optimum models appear in this zone. However, the vast majority of models in Zone1 have bad generalisation and local optimum models will not be selected as best model in all our empirical experiments. Although Zone1 owns a large
amount of models, the entropy values of feature importance are around zero, which means these models have not learn about features in a diverse way. Zone2 is the models’ group with explosively entropy of feature importance increasing. Features are learned by models more dispersed, which leads to dispersion of feature importance distribution. But models in this area have bad generalization, which does not conform the principle of maximum feature importance entropy. We cannot explain what caused this situation in theory but this phenomenon leads to the fact that models have bad generalisation when these models utilise features extremely disperse. Zone3 is the models’ group with increasing feature importance entropy values and better model generalization. And it is obvious that models in Zone3 are our purpose ones after pruning process. Lastly, models in Zone3 can be selected by CrossValidation accuracy to validate the models’ generalisation. The whole process is computational cheaper than operating CrossValidation in the whole hyperparameter tuning process.

The entropy value of feature importance depends on number of features and characteristics of dataset. Thus it is not reasonable to prune only according to the values’ number. Here we give a solution to rationally prune models according to confidence interval(cf. Figure 4.13). First, models in Zone1 with entropy values around zero can be pruned by a control parameter $\delta$. All models with feature importance’s entropy values under $\delta$ will be directly pruned. Entropy values in Zone2 and Zone3 have multi-segment increase, so we cannot distinguish them by values or slope tools. In this stage, we fit the rest models in Zone2 and Zone3 in probability distribution according to feature importance entropy values. According to our empirical findings, the distributions might be right-skewed
4.5 Experimental Results

Fig. 4.12 Regional divisions according to entropy values of feature importance

according to the emerging with explosive increase of entropy values. Here we transfer
original distribution to log-normal distribution and use Modified Cox method to calculate
the confidence interval. The left and tight tail of confidence interval could be optimised by
feedback of experiments.

Fig. 4.13 Process of Pruning models by confidence interval

Figure 4.14 shows the performance in selecting models after our pruning in Figure 4.13.
After pruning, we select these models with relatively good generalisability (the models
in the dashed box) from a global perspective. In other words, our pruning proposal helps
tree models to get better optimization to the optimal models’ interval in terms of model
generalisation. Since models in the optimal model interval exhibit a certain degree of
agglomeration in terms of dispersion, crossvalidation is a method for selecting the best
model within this optimal interval. The good news is that by combining the pruning method with crossvalidation, the whole process becomes more time efficient and effective. In the experiment, original crossvalidation in selecting best 10 models will cost 27:52:15.704 hours in training and validating. Combining pruning with crossvalidation, the process will cost 10:14:16.68 hours in training and validating. It means our proposal could save almost one-third of time in selecting tree models. The selecting models after pruning are accurate due to the accurate characteristics of optimal models’ interval. Pruning added crossvalidation can always get similar best models in terms of model generalisation compared to pure crossvalidation (evaluated by test accuracy).

![Comparison of performance between before/after pruning](image)

Fig. 4.14 Comparison of performance between before/after pruning

### 4.6 Threats to Validity

The feature contribution dispersion method in this research is based on a novel notion in tree models (containing training accuracy, features contribution and model complexity). So this method is very sensitive to input features. Consider the noise of features (we consider the “noise" as repeated or inappropriate features in datasets); selecting bias of features in datasets might impact the practical efficacy of our method. For example, in cases in which there is no benchmark feature collections (like financial market price prediction), the same model works better on datasets including more reasonable features than datasets with unreasonable ones. It means, the application of our method might take more time on feature selection in the data processing phase.

Another threat we must point out is about the settings of hyperparameters in models. In our experimental study, hyperparameters are selected from a reasonable search space.
But we cannot cover all possible hyperparemeters in the experiments. In this research, we select most hyperparemeters that work for classification purpose.

The stochastic characteristics of model evaluation is also a threat. We are trying to mitigate this threat by running 20 times each of the experiments and choosing 5 different tree models to increase the diversity domain of all aspects.

4.7 Conclusions

In this paper, we propose a new feature contribution dispersion approach for tree-models in model selection. According to our experiments, the results show that, in the vast majority of the cases, feature contribution dispersion allows to select models with high test accuracy when the model’s feature contribution dispersion value is low. Compared to general cross validation method (k-fold), our method performs better or similar to model selection performance while our method has a large efficiency improvement in model selection execution time. At the same time, we noticed that feature contribution dispersion based model selection would be impacted by features selection (eliminate noise generated by useless features).

Possible future research directions might be the optimisation of this explanation method and its application to other machine learning models.

4.8 Supplement figures
Fig. 4.15 Comparison between feature contribution dispersion and cross validation evaluating in accuracy matrix for the following tasks/models: (a) OpenML 1471 dataset and RandomForest, (b) OpenML 1494 and RandomForest, (c) OpenML 3 dataset and RandomForest, (d) OpenML 1504 and RandomForest, (e) OpenML 1489 and RandomForest
Fig. 4.16 Comparison between feature contribution dispersion and cross validation evaluating in accuracy matrix for the following tasks/models: (a) OpenML 1471 dataset and ExtraTree, (b) OpenML 1494 and ExtraTree, (c) OpenML 3 dataset and ExtraTree, (d) OpenML 1504 and ExtraTree, (e) OpenML 1489 and ExtraTree
Fig. 4.17 Comparison between feature contribution dispersion and cross validation evaluating in accuracy matrix for the following tasks/models: (a) OpenML 1471 dataset and GradientBoostingTree, (b) OpenML 1494 and GradientBoostingTree, (c) OpenML 3 dataset and GradientBoostingTree, (d) OpenML 1504 and GradientBoostingTree, (e) OpenML 1489 and GradientBoostingTree
Fig. 4.18 Comparison between feature contribution dispersion and cross validation evaluating in accuracy matrix for the following tasks/models: (a) OpenML 1471 dataset and XGboostTree, (b) OpenML 1494 and XGboostTree, (c) OpenML 3 dataset and XGboostTree, (d) OpenML 1504 and XGboostTree, (e) OpenML 1489 and XGboostTree
Chapter 5

Conclusion

This chapter concludes the thesis by restating our contributions and presenting potential directions for future work. A discussion is also included to reflect our thoughts on the usage of robust machine learning applications in quantitative finance.

5.1 Summary of Contributions

We restate the contributions of our thesis in three parts. In the first part, we provided a comprehensive overview and analysis of the research work on cryptocurrency trading. This survey presented a nomenclature of the definitions and current state of the art. The paper provides a comprehensive survey of 146 cryptocurrency trading papers and analyses the research distribution that characterise the cryptocurrency trading literature. Research distribution among properties and categories/technologies are analysed in this survey respectively. We further summarised the datasets used for experiments and analysed the research trends and opportunities in cryptocurrency trading. Future research directions and opportunities in cryptocurrency trading area are discussed in my reserach. We expect this survey to be beneficial to academics (e.g., finance researchers) and quantitative traders alike. The survey represents a quick way to get familiar with the literature on cryptocurrency trading and can motivate more researchers to contribute to the pressing problems in the area, for example along the lines we have identified.

In the second part, we analyzed a data-driven approach to predict mid-price movements in cryptocurrency markets, and covered a number of research questions en route regarding parameter settings, design of neural networks and universality of the models. The main finding of our work is the successful combination of an autoencoder and a walkthrough retraining method to overcome the decay in predictive power on live data due to non-stationary features on the order book. Our results show that our model has achieved good performance, quantified in a consistent F1-score of around 78%. By comparing different
5.2 Discussion

retraining methods (we call that Walkthrough), we found some tradeoffs between fixed and dynamic retraining. Prediction in high-frequency cryptocurrency markets is a challenging task because the environment contains noisy information and is highly unpredictable. We believe that our results can inform the design of higher level trading strategies and our networks architecture can be used as a feature to another estimator.

In the third part, we propose a new feature contribution dispersion approach for tree-models in model selection. According to our experiments, the results show that, in the vast majority of the cases, feature contribution dispersion allows to select models with high test accuracy when the model’s feature contribution dispersion value is low. Compared to general cross validation method (k-fold), our method performs better or similar to model selection performance while our method has a large efficiency improvement in model selection execution time. At the same time, we noticed that feature contribution dispersion based model selection would be impacted by features selection (eliminate noise generated by useless features).

5.2 Discussion

In this section, we include a discussion to reflect our thoughts on the usage of novel machine learning applications in financial markets and cryptocurrency trading.

In this thesis, we comprehensively talked about cryptocurrency trading. As we have analysed, we believe there exist multiple possible directions in cryptocurrency trading including Sentiment-based research, Long-and-short term trading research, Correlation between cryptocurrency and other assets, Bubbles and crash research, Game theory and agent-based analysis, Public nature of Blockchain technology and Balance between the opening of trading research literature and the fading of alphas. As an emerging market, cryptocurrency markets are more and more favored by alternative investment managers, no matter from the perspective of asset allocation or pure long short trade research. From an academic point of view, there are still many opportunities in the research of cryptocurrency market mechanism. At the same time, from the perspective of quantitative research or trading analysis, cryptocurrency still has a large gap compared with the traditional financial market analysis. Cryptocurrency is still a long way from being a stable asset allocation mode, which is also the motivation and starting point for us to study cryptocurrency transactions.

We have discussed the machine learning in applying in financial prediction in this thesis. Machine learning is a popular tool in quantitative analysing price formation or risk control in financial market. The core direction in applying machine learning in finance is to maintain the performances of machine learning models, or to transparency these black-box
5.2 Discussion

machine learning models to at least realise semi-automatic risk control while trading. If relatively stable or transparent machine models can be researched, the trading system framework based entirely on machine learning is likely to be widely used in the trading system. As far as the current situation is concerned, due to the bottleneck of machine learning and Explainable AI related technologies, machine learning can still be used as an auxiliary control or invisible factor extractor for auxiliary trading in the trading link, and machine learning still has great potential in the financial market trading environment.

In recent years researchers in academic and industry background focused more on AI in finance applications. Many famous conferences, like ICAIF, have attracted relatively broad interests in academic circles. We expect to see more of these discussions in future.
References


References


References


market for studying a cryptocurrency market. Journal of Economic Interaction and
Coordination, 12(2):345–365.

[118] Cocco, L. and Marchesi, M. (2016). Modeling and simulation of the economics of
mining in the bitcoin market. PloS one, 11(10).


[120] Coinbase (2020b). Our path to listing sec-regulated crypto securi-

[121] CoinMaketCap (2019). Top 100 cryptocurrencies by market capitalization.


tocurrency based on twitter sentiment analysis. CS229 Project, pages 1–5.


theorems and diffusion approximations. Available at SSRN 1757861.


References


167
References


learning with model-based acceleration. In *International Conference on Machine
Learning*, pages 2829–2838.

(CSUR)*, 51(5):1–42.

with a glimpse into buy and sell orders. In *2018 IEEE International Conference on
Data Mining (ICDM)*, pages 989–994. IEEE.


and other cryptocurrency (volume 2).


Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on
Artificial Intelligence*.


workbench. In *Proceedings of ANZIIS’94-Australian New Zealand Intelligent Information
Systems Conference*, pages 357–361. IEEE.


References


References


References


