

# Construction of Effective Regime-Switching Portfolios Using a Combination of Machine Learning and Traditional Approaches

Piotr Pomorski

A thesis submitted for the degree of  
Doctor of Philosophy

Department of Computer Science  
University College London

13/03/2024



I, Piotr Pomorski, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.



## **Abstract**

This thesis proposes and tests a detection-prediction-optimisation regime-switching framework to harness the regime shifts in financial time series for the advantage of portfolio managers. The work is divided into three research objectives, each explored in a separate chapter. First, however, Chapter 2 provides a thorough review of relevant literature and lays the foundation for the subsequent research chapters. Chapter 3 then focuses on the implementation of a novel regime detection framework that combines a technical indicator with a statistical method of regime detection. The resulting KAMA+MSR model outperforms the benchmarks in terms of stability and accuracy. Chapter 4 moves from detection to prediction, exploring the prediction of financial regimes ex-ante, using the regime labels from Chapter 3 and a Random Forest model as a predictor. Models for three different asset classes show solid out-of-sample classification performance and achieve excellent financial results (as measured by the Sortino ratio) based on a long/short trading strategy. Chapter 5 addresses the contributions and limitations of Chapter 4 by focusing on the third research objective: the construction of realistic, regime-robust portfolios. The constructed portfolio from Chapter 5 outperforms its benchmarks, despite not allowing shorting positions due to potential ethical and institutional constraints. The combined contributions of these three research chapters could serve as key components of a quantitative trading system, emphasising the practical aspect of this research.



## **Impact Statement**

This thesis proposes and tests a novel detection-prediction-optimisation regime-switching framework for asset managers, with the goal of providing a practical solution for real-world financial applications. The research objectives were divided into three chapters, with each chapter building on the preceding work. Chapter 3 implemented a novel regime detection framework, the KAMA+MSR model, which demonstrated superiority over its benchmarks in terms of stability and accuracy. Chapter 4 focused on the prediction of financial regimes ex-ante using the generated labels from Chapter 3, along with feature engineering and a Random Forest model. Models for three different asset classes all showed solid out-of-sample classification results and achieved excellent financial results as measured by the Sortino ratio. Chapter 5 addressed the limitations of the previous chapters, including the reliance on shorting (potentially problematic given the inability of many institutions to short assets), and successfully incorporated real-world factors into portfolio construction using the multi-period optimisation with a model predictive control algorithm. In summary, the combined contributions of Chapters 3, 4, and 5 achieved the overall objective of providing a practical solution for asset managers, potentially serving as key components of a quantitative trading system, as outlined in Section 2.1.3. The successful deployment of such a framework could have significant implications for the financial industry, potentially leading to improved investment strategies and more effective risk management.



## Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr Denise Gorse whose unwavering support and guidance have been invaluable throughout this research. Dr Gorse's academic knowledge and expertise in English grammar has been instrumental in the successful completion of this thesis. In addition, her insightful comments and suggestions have inspired me to strive for excellence and have helped me to develop my research skills.

I am also grateful to my internal viva assessors, Prof. Chris Clack and Dr Silvia Bartolucci, for their valuable critique and feedback, which have enhanced the quality of my work and provided finishing touches. In addition, I would like to acknowledge the academic and technical support provided by the Centre for Doctoral Training in Financial Computing and Data Science, in particular the support of Prof. Philip Treleaven and, more broadly, University College London.

Finally, but by no means least, I would like to thank my wife Zuzia for her understanding and patience, and my former manager James Party for his support and counselling on industrial aspects of this thesis.

# Contents

1. Introduction.....	20
1.1 <i>Motivation for this research</i> .....	22
1.2 <i>Research objectives</i> .....	24
1.3 <i>Thesis structure</i> .....	27
1.4 <i>Related publications</i> .....	28
2. Background and related work.....	29
2.1 <i>Financial background</i> .....	29
2.1.1 A brief description of asset allocation .....	29
2.1.2 MPT and its extensions .....	32
2.1.3 The process of building portfolios .....	35
2.1.4 Introduction to technical analysis.....	37
2.1.5 Regime switches .....	42
2.2 <i>Technical background</i> .....	46
2.2.1 Markov-switching regression model .....	46
2.2.2 Introduction to Kaufman's Adaptive Moving Average.....	49
2.2.3 Limitations of the Markov-switching model.....	51
2.2.4 Machine learning models for regime prediction.....	54
2.3 <i>Related work</i> .....	66
2.3.1 Regime detection methods .....	66
2.3.2 Regime prediction methods .....	77
2.3.3 Portfolio allocation methods related to regime switching.....	95

2.3.4 Does technical analysis add value in building portfolios? .....	101
3. Improving on the Markov-switching regression model by the use of an adaptive moving average .....	104
<i>3.1 Background and related work</i> .....	105
3.1.1 Popular extensions to the two-state Markov-switching regression model .....	105
3.1.2 Related work.....	106
<i>3.2 Data and methodology</i> .....	108
3.2.1 Data .....	108
3.2.2 Methodology .....	111
<i>3.3 Results</i> .....	123
3.3.1 Phase 1 .....	123
3.3.2 Phase 2 .....	126
<i>3.4 Discussion</i> .....	127
3.4.1 Comparison of performance across different asset classes .....	127
3.4.2 Why are three- or four-state models sub-optimal? .....	129
<i>3.5 Conclusions</i> .....	130
4. Predicting financial regimes by the use of the Random Forest-KAMA+MSR framework.....	131
<i>4.1 Background and Related Work</i> .....	132
4.1.1 Fractional differencing.....	132
4.1.2 Feature selection.....	136
4.1.3 Candidate features.....	142
<i>4.2 Data and methodology</i> .....	144
4.2.1 Data .....	144

4.2.2 Methodology.....	157
<i>4.3 Results and discussion.....</i>	<i>171</i>
4.3.1 Optimal hyperparameters.....	171
4.3.2 Feature importance .....	173
4.3.3 Model evaluation .....	178
4.3.4 To short, or not to short, that is the question.....	192
<i>4.4 Conclusions .....</i>	<i>193</i>
5. Combining regime switching predictive framework with model predictive control for multi-period portfolio optimisation.....	195
<i>5.1 Background and related work.....</i>	<i>196</i>
5.1.1 Multi-period optimisation.....	196
5.1.2 Model predictive control .....	197
5.1.3 Kalman filter.....	200
<i>5.2 Data and methodology .....</i>	<i>203</i>
5.2.1 Data .....	203
5.2.2 Methodology.....	205
<i>5.3 Results and discussion.....</i>	<i>213</i>
5.3.1 Optimal MPC parameters .....	213
5.3.2 MPC portfolio performance analysis .....	214
5.3.3 Is the higher risk worth it?.....	221
<i>5.4 Conclusions .....</i>	<i>224</i>
6. Conclusions and future work.....	226
<i>6.1 Summary and discussion of the experiments undertaken .....</i>	<i>226</i>
<i>6.2 Future work.....</i>	<i>228</i>

References .....	230
Appendix.....	258
<i>Appendix A</i> .....	258
<i>Appendix B</i> .....	265

## List of figures

Figure 1: Monthly return variation in major assets since 1985 until June 2021 .....	30
Figure 2: 52-week rolling risk-adjusted performance.....	31
Figure 3: 52-week rolling risk-adjusted performance.....	32
Figure 4: Simplified quantitative trading system. ....	35
Figure 5: Outline of the PGTS process, where blue bars are training and orange bars are validation sets.....	162
Figure 6: Feature importances for the Equity Random Forest model.....	173
Figure 7: Feature importances for the Commodity Random Forest model. ....	174
Figure 8: Feature importances for the FX Random Forest model.....	174
Figure 9: Cumulative geometric returns of the Equity, HMM, and KAMA+MSR models compared to a long-only position in MSCI ACWI (index benchmark).....	182
Figure 10: Cumulative geometric returns of the Commodity, HMM, and KAMA+MSR models compared to a long-only position in the Bloomberg Commodity Index (the index benchmark).....	183
Figure 11: Cumulative geometric returns of the FX, HMM, and KAMA+MSR models compared to a long-only position in the BBDXY Index (the index benchmark)....	184
Figure 12: Performance of the Equity Random Forest model compared to that from holding a long-only position in each equity asset from Table 10.....	185
Figure 13: Performance of the Commodity Random Forest model compared to that from holding a long-only position in each commodity from Table 10.....	185
Figure 14: Performance of the FX Random Forest model compared to that from holding a long-only position in each FX pair from Table 10. ....	186
Figure 15: Equity assets recommended to be bought by the Equity model. Note: "accumulated bearish positions" means assets which received a "buying" signal from the model.....	187

Figure 16: Equity assets recommended to be shorted by the Equity model. Note: "accumulated bullish positions" means assets which received a "shorting" signal from the model.....	188
Figure 17: Commodities recommended to be bought by the Commodity model. Note: "accumulated bearish positions" means assets which received a "buying" signal from the model.....	189
Figure 18: Commodities recommended to be shorted by the Commodity model. Note: "accumulated bullish positions" means assets which received a "shorting" signal from the model.....	189
Figure 19: FX pairs recommended to be bought by the FX model. Note: "accumulated bearish positions" means FX pairs which received a "buying" signal from the model.....	190
Figure 20: FX pairs recommended to be shorted by the FX model. Note: "accumulated bullish positions" means FX pairs which received a "shorting" signal from the model..	191
Figure 21: The Kalman filter in a graphical representation.....	200
Figure 22: KAMA+RF+MPC+Kalman portfolio performance vs. its benchmarks, over the entire holdout set .....	215
Figure 23: Price of nickel since 2018.....	216
Figure 24: The price of wheat since 2018.....	216
Figure 25: KAMA+RF+MPC+Kalman portfolio average monthly asset weights over the entire holdout set.....	217
Figure 26: KAMA+RF+MPC+Kalman portfolio performance vs. its benchmarks, over the holdout set up until 1 <sup>st</sup> of March 2022 .....	219
Figure 27: KAMA+RF+MPC+Kalman portfolio performance vs. its benchmarks, excluding nickel and wheat, over the entire holdout set..	220
Figure 28: KAMA+RF+MPC+Kalman portfolio performance vs. benchmarks up until 1 <sup>st</sup> March 2022 with the $\gamma\sigma$ parameter set to 50.....	223

Figure 29: Annualised risk-return relationship for a range of values of the risk-aversion parameter  $\gamma\sigma$ , calculated for the test data (up until 1<sup>st</sup> March 2022), with the  $\gamma\sigma$  values shown beside each dot ..... 224

## List of tables

Table 1: Pseudocode of a Random Forest algorithm.....	63
Table 2: Hyperparameters of Random Forest.....	64
Table 3: Data used in the work of Chapter 3.....	109
Table 4: Example matrix grouping clusters and regime labels.....	117
Table 5: Models used for comparison.....	119
Table 6: Two-way trading costs per each asset class .....	121
Table 7: Example results for the KAMA+MSR model.....	123
Table 8: Phase 1 (in-sample) results.....	124
Table 9: Phase 2 (out-of-sample) results.....	126
Table 10: List of asset classes, assets, and start dates.....	144
Table 11: Features used in the prediction of each asset class (excluding volume-related features for FX). ....	146
Table 12: Features usually applicable only to a specific asset class (rarely, to two). ....	154
Table 13: Example of a panel-like dataset ready for modelling.....	160
Table 14: PGTS settings.....	162
Table 15: Hyperparameters of Random Forest optimised in every experiment... ..	163
Table 16: Benchmarks used for financial performance assessment in the three different asset classes, where the KAMA+MSR model is the regime detection.....	170
Table 17: Optimal hyperparameters after tuning over PGTS validation splits.....	172
Table 18: Top 20 features with the 10 most positive and 10 most negative impact on the Equity Random Forest model's predictions.....	175

Table 19: Top 20 features with the 10 most positive and 10 most negative impact on the Commodity Random Forest model's predictions.....	176
Table 20: Top 18 features with the 8 most positive and 10 most negative impact on the FX Random Forest model's predictions.....	177
Table 21: Random Forest model financial evaluation metrics.....	179
Table 22: MCC scores for each Random Forest model.....	179
Table 23: Trading performance comparison between tested models.....	181
Table 24: Historical asset data used for MPO.....	204
Table 25: Optimised parameters of MPC .....	211
Table 26: Optimal MPC parameters.....	213
Table 27: KAMA+RF+MPC+Kalman portfolio performance vs. benchmarks, on the entire holdout set.....	214
Table 28: KAMA+RF+MPC+Kalmanp ortfolio performance vs. benchmarks, over the holdout set up until 1 <sup>st</sup> of March 2022.....	218
Table 29: KAMA+RF+MPC+Kalman portfolio performance vs. benchmarks (excluding wheat and nickel) on the entire holdout set .....	221
Table 30: KAMA+RF+MPC+Kalman portfolio performance up until 1 <sup>st</sup> March 2022 with the $\gamma\sigma$ parameter set to 50 .....	222
Table 31: Phase 1 (in-sample): annualised returns winning score ratio for equities .....	258
Table 32: Phase 1 (in-sample): annualised returns winning score ratio for commodities.....	258
Table 33: Phase 1 (in-sample): annualised returns winning score ratio for FX.....	259
Table 34: Phase 1 (in-sample): annualised returns winning score ratio for fixed income ETFs.....	259
Table 35: Phase 1 (in-sample): adjusted Sharpe ratio winning score ratio for equities.....	259

Table 36: Phase 1 (in-sample): adjusted Sharpe ratio winning score ratio for commodities .....	260
Table 37: Phase 1 (in-sample): adjusted Sharpe ratio winning score ratio for FX .....	260
Table 38: Phase 1 (in-sample): adjusted Sharpe ratio winning score ratio for fixed income ETFs .....	261
Table 39: Phase 2 (out-of-sample): annualised returns winning score ratio for equities .....	261
Table 40: Phase 2 (out-of-sample): annualised returns winning score ratio for commodities .....	262
Table 41: Phase 2 (out-of-sample): annualised returns winning score ratio for FX .....	262
Table 42: Phase 2 (out-of-sample): annualised returns winning score ratio for fixed income ETFs .....	262
Table 43: Phase 2 (out-of-sample): adjusted Sharpe ratio winning score ratio for equities .....	263
Table 44: Phase 2 (out-of-sample): adjusted Sharpe ratio winning score ratio for commodities .....	263
Table 45: Phase 2 (out-of-sample): adjusted Sharpe ratio winning score ratio for FX .....	263
Table 46: Phase 2 (out-of-sample): adjusted Sharpe ratio winning score ratio for fixed income ETFs .....	264
Table 47: Feature importances of the Equity Random Forest model. ....	265
Table 48: Feature importances of the Commodity Random Forest model. ....	269
Table 49: Feature importances of the FX Random Forest model. ....	271

## List of abbreviations

- 2S – Two-State
- 3S – Three-State
- ANN – Artificial Neural Networks
- ARCH – Autoregressive Conditional Heteroskedasticity
- ASR – Adjusted Sharpe Ratio
- ATM – At The Money
- BB – Bollinger Bands
- BBG - Bloomberg
- BLM – Black-Litterman Model
- CART – Classification And Regression Trees
- CRBI – Commodity Research Bureau Index
- DCC – Dynamic Conditional Correlation
- DC – Directional Change
- DJIA – Dow Jones Industrial Average
- DT – Decision Tree
- DSGE – Dynamic Stochastic General Equilibrium
- EGARCH – Exponential Generalised Autoregressive Conditional Heteroskedasticity
- EMA – Exponential Moving Average
- EMH – Efficient Market Hypothesis
- ER – Efficiency Ratio
- ETF – Exchange-Traded Funds
- FX – Foreign Exchange
- FIGARCH – Fractionally Integrated Generalised Autoregressive Conditional Heteroskedasticity
- GARCH – Generalised Autoregressive Conditional Heteroskedasticity
- GBT – Gradient Boosting Trees
- GLM – Generalised Linear Model
- GI – Gini Impurity
- GRU – Gated Recurrent Unit

- HLCV – High, Low, Close, Volume
- ICC – Inverse Covariance Clustering
- KAMA – Kaufman’s Adaptive Moving Average
- KF – Kalman Filter
- KNN – k-Nearest Neighbours
- KPSS – Kwiatkowski–Phillips–Schmidt–Shin
- LSTM – Long-Short Term Memory
- MACD – Moving Average Convergence Divergence
- MCC – Matthews Correlation Coefficient
- MPC – Model Predictive Control
- MPO – Multi-Period Optimisation
- MPT – Modern Portfolio Theory
- MSR – Markov-Switching Regression
- PMPT – Post-Modern Portfolio Theory
- QTS – Quantitative Trading System
- RF – Random Forest
- ROC – Rate Of Change
- RSDC – Regime Switching Dynamic Correlation
- RL – Reinforcement Learning
- RNN – Recurrent Neural Network
- SE – Shannon Entropy
- SHAP – Shapley Additive Explanations
- SPO – Single-Period Optimisation
- SVM – Support Vector Machines
- TAR – Threshold Autoregressive
- SETAR – Self-Exciting Threshold Autoregressive
- SKF – Switching Kalman Filter
- TGARCH – Threshold Generalised Autoregressive Conditional Heteroskedasticity
- VAR – Vector Autoregressive
- WQ – WorldQuant
- XGB – Extreme Gradient Boosting

# 1. Introduction

## 1.1 Motivation for this research

Financial markets are characterised by periods of evolving, low-volatility growth separated by disruptive, high-volatility contractions. Bursting stock market bubbles, currency crises, and sharp changes in commodity returns are common examples of panic-driven selloffs after a steady rally. These regimes vary by significant changes in asset returns, variances, and correlations, laying a groundwork for specific research to be carried out and subsequently exploited by portfolio managers.

Classic portfolio allocation methods, such as the legendary Markowitz portfolio theory, do not incorporate varying characteristics of financial time series which are prone to change, sometimes abruptly, whenever an asset price enters a new regime. This effectively may cause portfolios to suffer significant losses in case a sudden market crash occurs, as well as to miss out on large gains if an after-crash defensive portfolio is run on a rallying market. This issue prompted research on methods that can indeed capture these changes and serve as a starting point to build regime-switching portfolios.

Hamilton's seminal work of applying Markovian frameworks to detect regime changes in financial time series (Hamilton, 1989) has initiated a new era of research focused on building robust portfolios capable of dynamically adjusting to new regimes. Indeed, an influential work of (Ang & Bekaert, 2002) on the effect of regime shifts on asset allocation revealed how essential it is to build portfolios using regime-switching detection methods. More specifically, in this work a vanilla portfolio which changed its allocation based only on a detected regime, such as low or high variance, was shown to generate a Sharpe ratio<sup>1</sup> almost twice as high as simply holding the US market. This result was because the vanilla portfolio was capable of dynamically hedging itself in the event of market turbulence and taking

---

<sup>1</sup>  $S_r = \frac{(r_p - r_f)}{\sigma_p}$ , where  $S_r$ ,  $r_p$ ,  $r_f$ ,  $\sigma_p$  stand for Sharpe ratio, mean portfolio return, risk-free return, and standard deviation of portfolio returns, respectively.

on more risk the moment the volatility switched from high to low; however, even though the theory looked ideal, it did not produce good results under real world trading conditions.

While the work of (Ang & Bekaert, 2002) above showed promise, it suffered from the fact that transaction costs were not included. In addition, their use of the two-state Markov-switching model (discussed in detail in Section 2.2 and used also, though in a more sophisticated way, in Chapter 3), to detect regimes is suboptimal in itself, particularly in terms of accurate timing of regime changes and spurious detection of switches. Building a portfolio based on such naïve signals may result in a costly allocation that in the real world may not be prepared for the state shift on time but even the event has occurred. In addition, financial time series cannot be characterised simply by high and low volatility periods; in practice, even as many as four different regimes can be distinguished, which the standard two-state model cannot incorporate. Finally, a vanilla allocation based on a detected regime may also be inefficient—what if the portfolio holds five assets and they are currently all find themselves in one single state? Should the allocation be naïvely even? Or maybe due to specific reasons one asset should receive a higher weight in a portfolio than the others? These questions, alongside the issues of timing, overly-frequent switches, and the existence of multiple regimes within financial time series, significantly complicate the construction of an optimal, robust portfolio ready even for an abrupt change in the regimes of the assets it holds.

The above-described problems have not yet been solved; as will be seen in the related work (Section 2.2) some methods try to tackle some of these problems but overlook the others. For instance, the problem of detecting the regime without overly-frequent switches can be solved even by the academic 'enemy of the state', technical analysis, or by the application of specific machine learning methods; however, both approaches may have issues with accuracy (particularly in comparison with the Markovian framework), fail to distinguish more than two regimes, or be difficult to deploy in the real world. To address the problem of market timing, detection can be replaced by prediction, using various machine learning algorithms which can effectively solve the issue of lagging the reality. This, however, requires accurate labels (regimes) to predict, as well as the selection of the most

efficient multiclass classification algorithm. Finally, the construction and optimisation of portfolios by acting on generated signals (either predicted or detected regimes) has also been an active topic of discussion. The literature focuses either on applying machine learning directly to the results of regime detection mechanisms, to overcome the issue of these detections lagging the reality, or the use of less complicated—but more prone to inefficiencies—rule-based allocation techniques, as a second step after making regime change predictions with machine learning.

The current state of the art, as outlined above, therefore clearly calls for a unified approach, a quantitative trading system that tries to tackle all three major problems—labelling, timing, and regime-based portfolio optimisation—at once, and for a detection-prediction-optimisation regime-switching framework that in addition, and very importantly, would also be efficient in the real-world environment.

## 1.2 Research objectives

Given the ever-changing nature of financial time series, and the issues faced by portfolio managers in adapting to and fully exploiting them, the objective of the thesis will be to develop novel methods that can harness the regime shifts for the manager's own advantage, by the use of various tools that can sequentially help one another in constructing a quantitative trading system which embeds the dynamic regime-switching mechanism. Three research questions will need to be answered en route to this goal:

### 1. Can the standard regime detection method be improved?

As was pointed in the motivation for this thesis, regime detection methods serve as a starting point to build regime-robust portfolios. Even though the most currently popular one, the Markov-switching

regression model (discussed in detail in Section 2.2.2), can be applied to construct portfolios, it is still suboptimal in terms of capturing all relevant financial regimes, as well as being sometimes prone to overly-frequent switches. This phase of the research will overcome these issues by blending the classic two-state Markov-switching model with technical analysis, more specifically, Kaufman's Adaptive Moving Average (KAMA), to discern four major regimes (two longer-lasting and two transitory), while also considering smoothness and accuracy, resulting in what is termed here the *KAMA+MSR* model. The novel method is applied to multiple assets and asset classes, as well as being compared to other techniques, including as the two-state Markov model itself, to show robustness and relative performance, notably also on a cost-adjusted basis. It is, however, critical to point out that the aim of this experiment is not primarily to generate signals which can be immediately acted on in building portfolios (though it is shown that this can, in fact, be profitable, when based on KAMA+MSR detections), as this method, similarly to the other detection methods described in Chapter 2, still suffers from an inherent lag behind reality; it is rather to present a useful framework that can generate accurate labels for the experiment in Chapter 4 which focuses on predicting the regimes ahead of time.

## **2. Can the generated labels from the previous experiment be efficiently predicted?**

The aim of the work of Chapter 4 is to utilise machine learning techniques to predict detected regimes from the previous research phase in Chapter 3. To achieve this goal effectively, this chapter will employ the popular Random Forest machine learning model, whose use is thoroughly motivated in Section 2.2.4. In addition, data science techniques such as feature engineering, feature selection, and hyperparameter optimisation will be applied to significantly enhance the accuracy of the chosen predictive algorithm. Finally, the study will

test the proposed method's robustness by evaluating the model across three asset classes; although a more advanced trading strategy than the traditional asset-cash exchange will be implemented, this experiment will not perform complex asset allocation given the generated signals, which would include factors focusing on risk, turnover, and holding constraints. This is the aim of the work of Chapter 5, which attempts to construct more realistic, regime-robust portfolios by exploiting the Random Forest predictions.

### **3. Can the predictions from the previous experiment be optimised to build regime-robust portfolios?**

The aim of this phase of the research, presented in Chapter 5, is to develop a portfolio optimisation algorithm that addresses the trading performance limitations of the previous experiment. Specifically, the trading strategy based on Random Forest predictions, from the previous phase of the research, could benefit from a dynamic weight allocation and by the abandoning of short trading, which may be impossible for various institutions due to ethical concerns. This study will implement multi-period optimisation (MPO) with model predictive control (MPC), a promising wrapper for a regime-switching predictive framework that can incorporate multiple factors, such as time-varying return estimates, transaction costs constraints, risk constraints, and the trade-off between short-term versus long-term asset holding. MPC can use regime predictions as return estimates to establish realistic long-only asset allocation among different asset classes. This study will apply MPC to the signals generated by the Random Forest models from the previous phase of research in Chapter 4, and thus will effectively complete the proposed regime-switching framework of this thesis.

### **1.3 Thesis structure**

The structure of this report is as follows.

Chapter 2 gives necessary background to the work of the thesis. Section 2.1 gives financial background on the problem of asset allocation and the modern process of building portfolios, as well as brief introductions to the topics of technical analysis and regime switching; Section 2.2 consists of technical background relating to the tools that will be applied in Chapters 3 and 4; and then, finally, Section 2.3 reviews existing literature relating to each of the three research questions of the previous section.

Chapter 3 presents a novel regime detection method that incorporates some elements of technical analysis can detect regime switches more effectively than traditionally-used academic methods based on Markov-switching regression models.

Chapter 4 presents a novel regime prediction framework that uses the regime-detections from the model of Chapter 3 to generate labels that the chosen machine learning algorithm, Random Forest, will aim to predict.

Chapter 5 presents a novel portfolio allocation method, multi-period optimisation with model predictive control, which builds on the work of Chapter 4 and is able to construct regime-robust portfolios under realistic constraints.

## **1.4 Related publications**

The following publications resulted from the work of this thesis:

1. Pomorski, P., & Gorse, D. (2023). "Improving on the Markov-Switching Regression Model by the Use of an Adaptive Moving Average." *New Perspectives and Paradigms in Applied Economics and Business: Select Proceedings of the 2022 6th International Conference on Applied Economics and Business*, pp. 17-30. Cham: Springer International Publishing.
2. Pomorski, P., & Gorse, D. (2023). "Improving Portfolio Performance Using a Novel Method for Predicting Financial Regimes." *International Conference on Machine Learning, Optimization, and Data Science*, pp. 94-108. Cham: Springer Nature Switzerland, 2023.
3. Pomorski, P., & Gorse, D. (2023). "Multi-Period Portfolio Optimisation Using a Regime-Switching Predictive Framework." *arXiv preprint arXiv:2308.09263*

## **2. Background and related work**

### **2.1 Financial background**

This section is focused on the basic idea of asset allocation, particularly its purpose and advantages over naïve keeping of assets. In addition, it will cover a quantitative portfolio construction workflow, and will serve as an introductory point to technical analysis and regime switches.

The purpose of this section is to familiarise the reader with the general terms that will be used throughout this thesis, as well as provide initial reasoning for applying regime-switching framework to build portfolios.

#### **2.1.1 A brief description of asset allocation**

Asset allocation refers to an investment strategy that focuses on dividing investment portfolios between different asset classes to balance risk and reward. These asset classes can be separated into three main categories: public equities, fixed income, and cash and equivalents<sup>2</sup>. Any asset that is outside of these classes, such as commodities, real estate, or private equity, creates a fourth category referred to as alternative. The classes vary from one another with their levels of risk and return; hence each behaves differently over time and provides a diversifying effect for the entire portfolio.

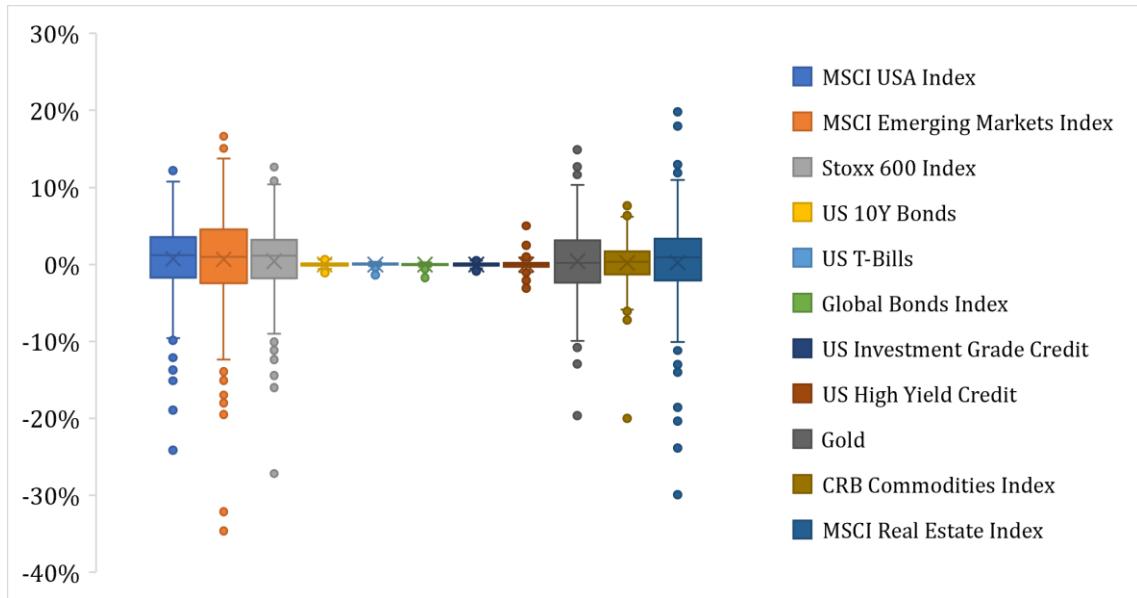
Figure 1 (on the following page) presents return variation within major assets since 1985. Not surprisingly, public equities tend to be more volatile than fixed income, although their underlying risk is rewarded by significantly higher returns. This does not necessarily mean risk-loving or risk-averse asset managers

---

<sup>2</sup> Cash equivalents include bank certificates of deposit, Treasury bills, commercial paper, banker's acceptances, and any other money market instrument with maturity period of 90 days or less.

will opt for one class or another, rather will try to combine both to maximise risk-adjusted returns.

Figure 1: Monthly return variation in major assets since 1985 until June 2021. Source: Bloomberg, Refinitiv Datastream.



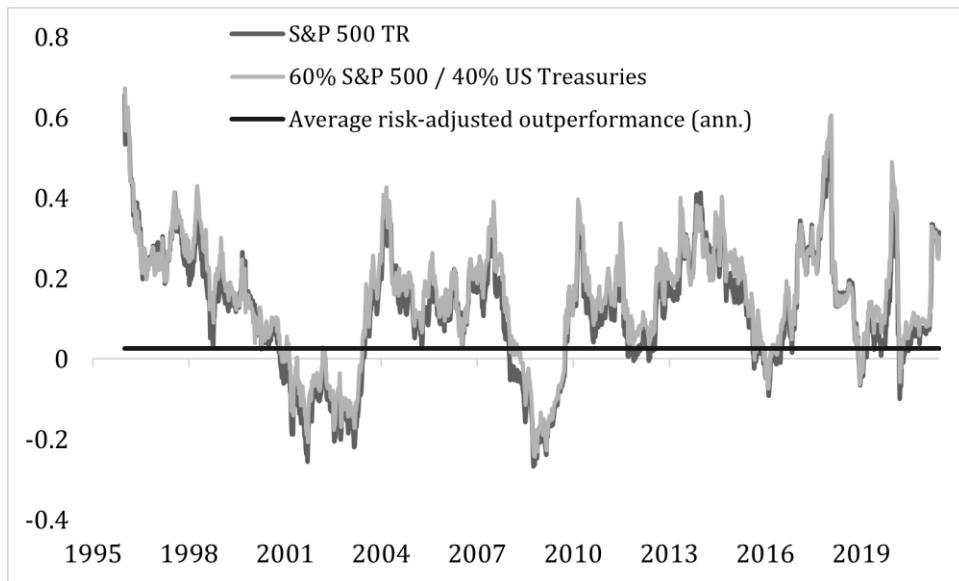
Asset allocation aims to achieve this optimal combination by moving all or partial funds from one class (or asset within the class) to another. There are several strategies an asset manager can use to optimally allocate assets within a portfolio, most notably a naïve  $1/N$  portfolio<sup>3</sup>, buying and holding a specific mix of them, i.e., 60% equities / 40% bonds, or dynamically adjust the class weight depending on the current (and predicted) state of the economy. Some strategies blend static and dynamic approaches by allocating constant weights to classes (equities, bonds, alternatives) over a long term and tactically adjusting the weights to maximize short-term returns. For instance, an asset manager may decide to allocate 50% of funds to equities, 30% to bonds, and 20% to alternatives over next 15 years but tweak these weights over a short period of time; the manager may move 10% from bonds and 10% from alternatives to equities, as they are predicted to significantly outperform other classes within the next three months.

Figure 1 also shows that even within asset classes there are variance discrepancies, such that in equities emerging markets has been a riskier asset than

<sup>3</sup> Where  $N$  stands for the number of assets, hence  $1/N$  translates to an equally weighted portfolio.

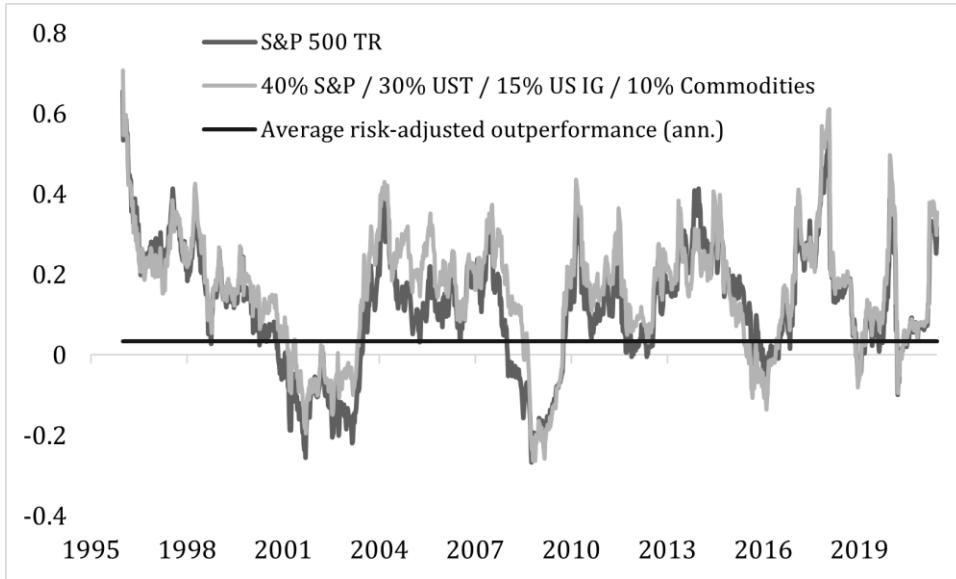
US or Europe (Stoxx 600) indices over last 25 years. Similarly, US high yield corporate bonds tend to generate the highest returns within the fixed income class, relatively not only to US, but also global bonds; however, this feature has also been associated with the highest risk imposed. Both examples indicate that asset managers do not only have to allocate funds between asset classes, but also within them to generate higher risk-adjusted returns.

*Figure 2: 52-week rolling risk-adjusted performance, where risk-adjusted stands for a mean return subtracted by a risk-free rate. Source: Bloomberg.*



Considering Figures 2 (above) and 3 (on the following page), it can be shown how diversification affects risk-adjusted returns. By comparing a simple 60% stocks / 40% bonds (proxied by S&P 500 total return and US Treasuries total return indices) portfolio to 100% stocks, the manager can achieve approximately 2.5% higher risk-adjusted returns on annual basis. If more asset classes are added to diversify the portfolio (namely, US investment grade credit and commodities) while the weight of stocks and bonds are diminished from 60% to 45% and 40% to 30%, respectively, the average risk-adjusted outperformance rises to approximately 3.4% on annual basis.

Figure 3: 52-week rolling risk-adjusted performance, where risk-adjusted stands for a mean return subtracted by a risk-free rate. Source: Bloomberg.



However, to determine what weights exactly should be assigned to each asset or class in order to outperform a benchmark of choice has been a challenge. Even the risk-adjusted performance of the allocations from Figures 2 and 3 may significantly differ from any other appropriations of the same classes, not to mention allocating weights in more complex portfolios consisting of tens, hundreds, or thousands of assets which is not uncommon in the industry. This premise spurred thorough academic research and gained significant attention particularly after the seminal work of Harry Markowitz from (1952) who introduced the Modern Portfolio Theory (*MPT*).

### 2.1.2 MPT and its extensions

#### *Modern Portfolio Theory*

Modern Portfolio Theory tells how risk-averse investors can build portfolios to maximise expected return based on a given level of market risk. The critical part of this theory is the mean-variance model which selects the optimal portfolio by balancing the risk-return ratio using the estimated mean vector and the covariance

matrix of asset returns. The model is inexpensive to use, easy to understand, and has no constraints with regards to type of assets that can be included, i.e., it allows mixing equities with bonds or commodities. However, since it is highly sensitive to expected mean-variance parameters (often calculated using historical data), it may provide misleading results, particularly in case of events that never happened in the past. In addition, it assumes a static mean vector and the covariance matrix of asset returns which do not adapt to changing conditions of the market. Also, it assumes normal distribution of asset returns that in reality often follow highly skewed distributions (log-normal) resulting in constructing portfolios with understated volatility or overstated returns (Bae, Kim, & Mulvey, 2014), (Low, Faff, & Aas, 2016). Finally, the mean-variance model has struggled to outperform the naïve  $1/N$  portfolio out-of-sample in terms of a higher Sharpe ratio, indicating that the errors in estimating means and covariances wear down all the gains from optimal diversification (DeMiguel, Garlappi, & Raman, 2009).

### *Classic extensions to MPT*

The disadvantages of the Modern Portfolio Theory, which somewhat hinder its vanilla application in real world, effectively have given rise to new theories and models, such as popular in the industry the *Post-Modern Portfolio Theory* (PMPT) and the *Black-Litterman Model* (BLM).

PMPT tackles the issue of the assumption of normality in MPT by focusing on the downside risk only. To specify, if there are two portfolios, one with small frequent losses and the other with rare large declines, the mean-variance model may treat them equally. However, an asset manager may in fact prefer the portfolio with small frequent losses, since they are easier to sustain. By including semi-variance instead of variance, PMPT allows the manager to select the portfolio which minimises the downside risk instead of both negative and upside variance, thereby allowing non-normal distribution of asset returns (Rom & Ferguson, 1994).

The Black-Litterman Model tends to extend MPT by moving from constant to changing mean-variance parameters, effectively incorporating new assumptions about the future. The mean-variance settings of the model are initially like MPT but

instead of keeping them static going forward, BLM allows their modification by considering the opinion of the asset manager regarding future asset performance. Such solution remains effective until the manager's outlook is correct and unbiased; in case these assumptions are poor, the asset allocation based on the model's results is no longer optimal. In addition, frequent changes in mean-variance assumptions may lead to excessive transaction costs rendering BLM less practical (Black & Litterman, 1992), (Cheung W., 2010).

Both classic extensions try to help build portfolios which should be robust to dynamic market conditions that the simple mean-variance model may over- or underestimate. In case of the PMPT framework, the objective is to eliminate portfolios with large downside risks which are present during market corrections or recessions, whereas the BLM can incorporate ever-changing assumptions about the future which the manager may predict to be more volatile than usual. This still does not fully solve the problem of adapting portfolios to shifting market conditions, as PMPT does not anticipate anything rather adjusts MPT for a non-normal distribution of returns, whereas the qualitative input of the asset manager to the BLM model may simply be inaccurate. The described inefficiencies have called for alternative asset allocation methods, a selected number of which (listed below) will be discussed in Section 2.3:

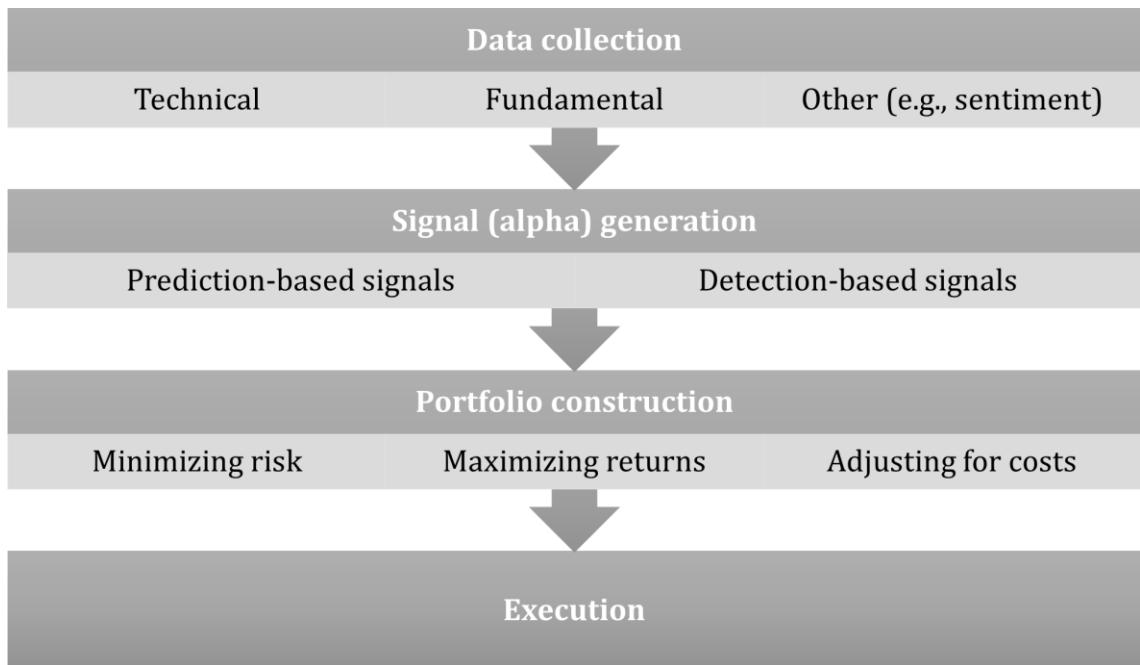
- 1) Traditional techniques which exploit regime switching to adapt portfolios to dynamic market conditions.
- 2) Methods that utilise *machine learning* (ML) and optimal control solutions.
- 3) Methods associated with technical analysis.

However, in terms of the entire asset allocation process, *de facto* portfolio optimisation is one of the last steps. The section on the next page briefly outlines the granular process of asset allocation, also known as a quantitative trading system.

### 2.1.3 The process of building portfolios

Many investment managers nowadays implement a procedure for asset allocation, known as a quantitative trading system (QTS), which involves collecting data, generating trading signals, constructing and optimising portfolios, and finally executing the trade (Ta, Liu, & Addis, 2018) (see Figure 4 below for a simplified QTS pipeline).

Figure 4: Simplified quantitative trading system.



The data collection step begins with obtaining relevant datapoints that will be subsequently fed into the signal generation stage. These readings can be as simple as asset returns data; however, they may also include fundamental, technical, macroeconomic, or even text-based entries for the detection of sentiment, depending on the trader's modelling needs. Common financial data providers available on the market are Bloomberg, Refinitiv, and FactSet.

The following signal generation stage is a critical step to build successful portfolios. It involves methods which focus on detecting patterns (such as regime shifts described in Section 2.1.5 later) or predicting them ahead. Both techniques

require algorithms (either traditional or machine learning to generate necessary detections or predictions that ultimately become signals to allocate weights or pick assets within a portfolio. This thesis will focus, in Chapters 4, on generating signals by predicting events rather than detecting them, a justification for which will be outlined later in Sections 2.2.3 and 2.2.4, which describe the limitations of merely detecting patterns in financial time series (in this case, regime switches); however, a detection stage is a critical component of the entire framework, and will be addressed in Chapter 3. In addition, some limited application of detection-only methods to build portfolios will be discussed in Section 2.3.3, which describes portfolio allocation methods.

Although more details are outlined in Section 2.3.2, detecting a pattern in a financial time series is useful in answering the question of what exactly should be predicted, that is, what labels should the model utilise for training, as such predicted labels will become an alpha signal to act on in the portfolio optimisation stage. Section 2.3.1 describes multiple methods based on regime-switching that can be helpful in generating such labels, particularly the *Markov-switching regression model* (MSR). Section 2.3.2 also discusses helpful techniques to predict these labels ahead of time to serve as final signals, ready to be optimised.

The generated signals are subsequently fed to a portfolio optimisation algorithm of the trader's choice. The idea behind this stage is to utilise the alpha signals in order to maximise risk-adjusted returns for the portfolio while taking into consideration potential costs of transaction. Both the common and less popular techniques, whether traditional or machine learning-based, of portfolio construction will be outlined in Section 2.3.3. This topic is also the main focus of Chapter 5.

The final stage, execution, incorporates algorithms which send signals to, for instance, market brokers, that the trader is willing to buy or sell assets of interest. This thesis will not address this topic; however, the execution step is very important from the portfolio performance perspective, as making a badly untimed or too large transaction may significantly increase the implicit costs, such as bid-ask spread or market impact cost (Ta, Liu, & Addis, 2018).

This thesis will focus on portfolio building from the regime-switching point of view, the reasoning for which is outlined in Section 2.1.5 and further in Section 2.3. However, before getting into details of techniques that could be applied to signal generation or portfolio construction, it is critical to introduce terms that are a significant part of this work: technical analysis and regime switched.

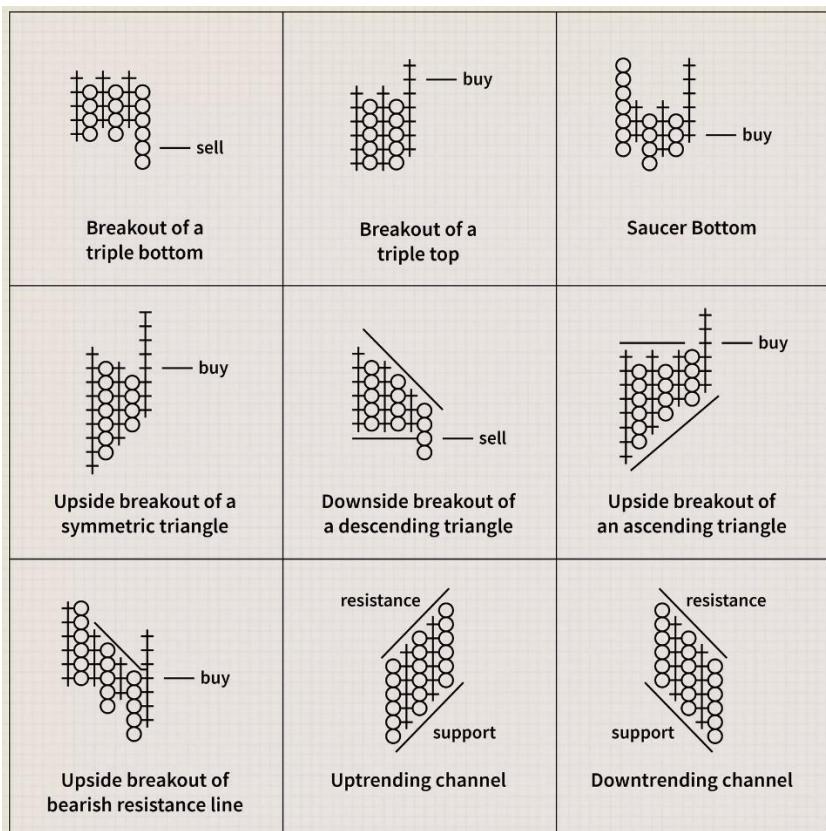
#### **2.1.4 Introduction to technical analysis**

##### *A short history of technical analysis*

*Technical analysis* (TA) can be understood as a technique that attempts to predict future price movements based on the study of the asset historical price and volume. It is tightly linked to analysing charts and detecting buying and selling patterns which can be ultimately helpful in forecasting short- or long-term price directions. Many proponents of TA consider it to be the original form of investment analysis, dating back even to Dutch financial markets of the 17<sup>th</sup> century. In the 18<sup>th</sup> century, a merchant from Japan, Munehisa Honma, began studying charts of rice prices, subsequently creating candlestick techniques that have remained an important trading tool until today. In the US, the use of technical analysis dates to the late 1800's when Charles Dow, the inventor of Dow Jones Industrial Average, began publishing his TA theories in *The Wall Street Journal*. However, the first major academic study appeared much later, in 1933, when economist Alfred Cowles III published his "Can Stock Market Forecasters Forecast?" paper in the *Econometrica* journal. Also, during the 1930's, several prominent market experts issued their own theories and findings on TA, most notably Richard W. Schabacker in "Technical Analysis and Stock Market Profits", and Richard Wyckoff in "The Richard D. Wyckoff Method of Trading in Stocks"; the latter introduced the four price cycles theory which will be discussed in Chapter 3.

Pre-World War II positions in technical analysis principally focused on charting (due to a lack of essential computing power), such as the famous "point and figure" chart introduced by already mentioned Charles Dow (see Figure 5, on the following page).

Figure 5: Pre-World War II point and figure charts and associated strategies. Source: Investopedia.com.



After the War, more publications began focusing on the mathematical side of TA, beginning with a seminal work of Robert D. Edwards and John Magee from 1948, who, apart from chart patterns, concentrated on trend analysis. Soon, new developments emerged, additionally encompassing price momentum (e.g., Relative Strength Index), volatility (e.g., Bollinger Bands), and volume (e.g., On-Balance Volume). Wall Street quickly caught up with these nuances and has begun implementing TA in trading and research on a larger scale. Figure 6 (on the following page), showing NASDAQ 100 with various technical indicators overlaid, can serve as an example of more recent use of TA.

Figure 6: NASDAQ 100 (2018 - 2021) with technical indicators, such as Relative Strength Index, moving averages, Bollinger Bands, and Moving Average Convergence-Divergence (MACD).



However, together with a growing interest in technical analysis within the industry, TA spurred controversy in academia, which started referring to it as pseudoscience (a topic that will be revisited in more detail in Section 2.2.1). Burton Malkiel, a leading proponent of the Efficient Market Hypothesis (EMH), author of the classic “A Random Walk Down Wall Street” from 1973, and critic of technical analysis, said:

“Obviously, I am biased against the chartist. This is not only a personal predilection, but a professional one as well. Technical analysis is anathema to the academic world. We love to pick on it. Our bullying tactics are prompted by two considerations: (1) the method is patently false; and (2) it's easy to pick on. And while it may seem a bit unfair to pick on such a sorry target, just remember: it is your money we are trying to save.” (Brock, Lakonishok, & LeBaron, 1992)

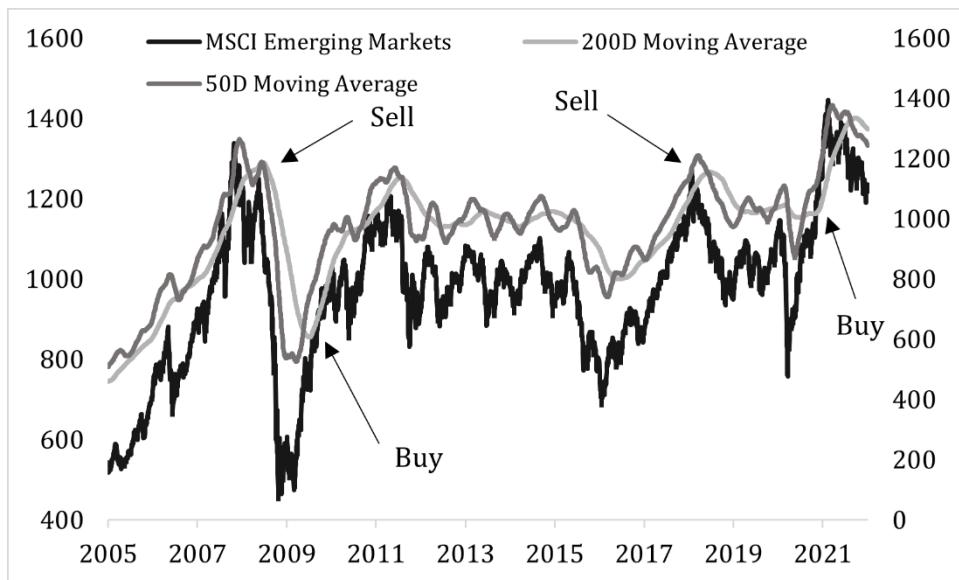
This adverse attitude stems from the fact that TA contradicts EMH which states it is impossible to “beat the market” using past data, since the stock price already incorporates historical and current information. Thus, according to this theory, stocks are always traded at their fair value which effectively renders technical analysis obsolete. However, as EMH has itself been challenged by the emerging field of behavioural finance and its theories, such as the Adaptive Market Hypothesis and Behavioural Technical Analysis, the industry has continued to welcome TA. Over the following decades, the statistical toolset of TA grew, encompassing both simple and more elaborate techniques. One of these tools, based on using moving averages in trading, has attracted particular attention by the industry professionals. Moving averages will also play a role in the work of this thesis, and therefore will be briefly described below.

### *Moving averages*

A simple moving average is an arithmetic mean of an underlying asset price over a period of  $n$  timesteps. Traditionally, traders seek times when the underlying price crosses a moving average or when two moving averages, with different  $n$  values, cross each other. Such events essentially serve as trading signals. If the price intersects a length  $n$  moving average in an upward direction, or a short-term moving average crosses the long-term moving average in an upward direction, it is considered a bullish signal and the trader buys an asset. If the opposite occurs, then the trader initiates a sale, since it indicates a bearish signal.

Figure 7, below, shows selected signals generated from two crossing moving averages, specifically 50 and 200-day moving averages.

*Figure 7: MSCI Emerging Markets trading price movements after moving average crosses. Source: Bloomberg.*



As can be seen in the above figure, in 2008 the 200-day moving average crossed downwards the 50-day moving average, signalling a “sell” signal which turned into “buy” between 2009 and 2010. Similarly, the “sell” signal was generated in 2018, pointing at the end of a significant rally since 2016. However, Figure 6 also clearly shows that bluntly relying on the crossing points is not optimal. For instance, the first “buy” signal ideally should be flagged earlier, as it missed significant gains, similarly to the one in early 2020. Also, between 2011 and 2017 the moving averages cross each other too frequently which may adversely affect cost-adjusted returns. These issues are the reason why traders use other combinations of moving averages (e.g., 20-day with 100-day) or simply rely on one moving average and monitor how the price oscillates around it.

The popularity of utilizing moving averages in trading strategies prompted the creation of different forms of these trend-following indicators, such as exponential moving averages, weighted moving averages, triangular moving averages, and adaptive moving averages, to name a few<sup>4</sup>. The difference between

---

<sup>4</sup> See <http://www.tadoc.org/index.htm> for an extended list. Accessed on 04/03/2022.

these variations lies in their differing capabilities to smooth the underlying price without increasing the lag, to more quickly capture the change in trend (Aistis, Vaidotas, & Edmundas, 2013). Some are better at smoothing the underlying price, such as simple moving averages, and some, such as the zero-lag moving average, are better at minimising the lag, though these cannot be achieved simultaneously. An adaptive moving average is, however, a specific class of a moving average that even so attempts to do so, accounting for both trend and volatility factors to balance the smooth-lag issue (Kaufman, 1995). Because this particular form of moving average analysis plays a key role in the work of this thesis, it will be described in detail in Section 2.1.7 and in the work dependent on it, presented in Chapter 3.

The trend shifts in underlying asset prices that different moving average strategies try to detect are tightly linked to the notion of changing market conditions discussed in Section 2.1.2. These are often referred to as “regimes” or “states”, and are a significant component of financial time series, and the topic of regime switches will hence now be addressed.

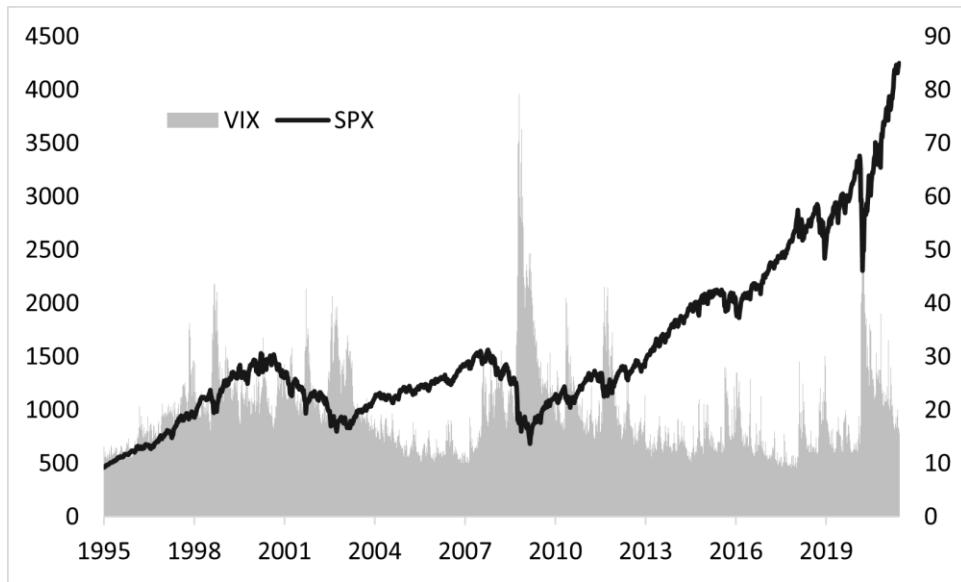
### **2.1.5 Regime switches**

Financial markets often change their behaviour in a recurring (long-term recessions vs. expansions) or unique (short-term price corrections or abnormal rallies) ways which significantly affects the mean, volatility, and correlation patterns of asset returns (Ang & Bekaert, 2004). These regimes are often associated with changing variance, specifically low and high volatility states which affect how asset returns are distributed. Such distressed and calm periods in a financial market is shown in Figure 8 (on the following page) which plots S&P 500 and VIX<sup>5</sup> that stands for implied volatility of S&P 500 returns.

---

<sup>5</sup> Refer to a white paper of (Cboe Exchange, Inc., 2021) for more details on VIX.

Figure 8: S&P 500 and its implied volatility captured by VIX. Source: Bloomberg, CBOE.

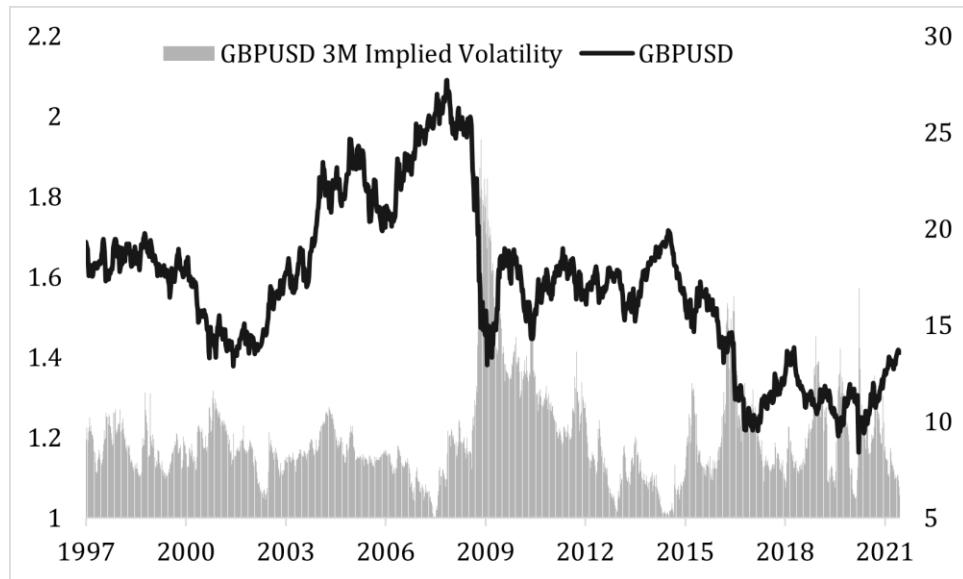


Clearly, as can be seen in the above figure, some periods stand out with regards to high VIX readings (hence rallying volatility), such as the dot-com bubble (2000-2002), The Great Financial Crash (2008-2009), Euro crisis and taper tantrum between 2011 and 2013, and finally coronavirus breakdown in 2020. On the other hand, when S&P 500 steadily increased, such as between 2004-2007 or 2013-2020 (with small breaks in 2015/2016 and 2018), the VIX fell, indicating a low volatility regime. Apart from variance, both regimes also differ by asset returns: low volatility states are usually featured by higher asset returns, whereas high volatility periods are often associated with relatively lower asset returns. Finally, the high variance regime is usually linked to higher equity return correlations relatively to the low variance state (Ang & Chen, 2002), (Ang & Bekaert, How Regimes Affect Asset Allocation, 2004), (Ang & Timmermann, 2012).

Multiple studies have affirmed the existence of regimes in other than equity financial time series, most notably in business cycles, interest rates, commodities, and foreign exchange (Hamilton, 1989), (Garcia & Perron, 1996), (Jeanne & Masson, 2000), (Alizadeh, Nomikos, & Pouliasis, 2008), (Bae, Kim, & Mulvey, 2014). In business cycles, expansions and recessions evolving around a long-term trend are examples of two different states. In interest rates, the regime switch may stem from a central bank changing its policies to regulate monetary aggregates, such as

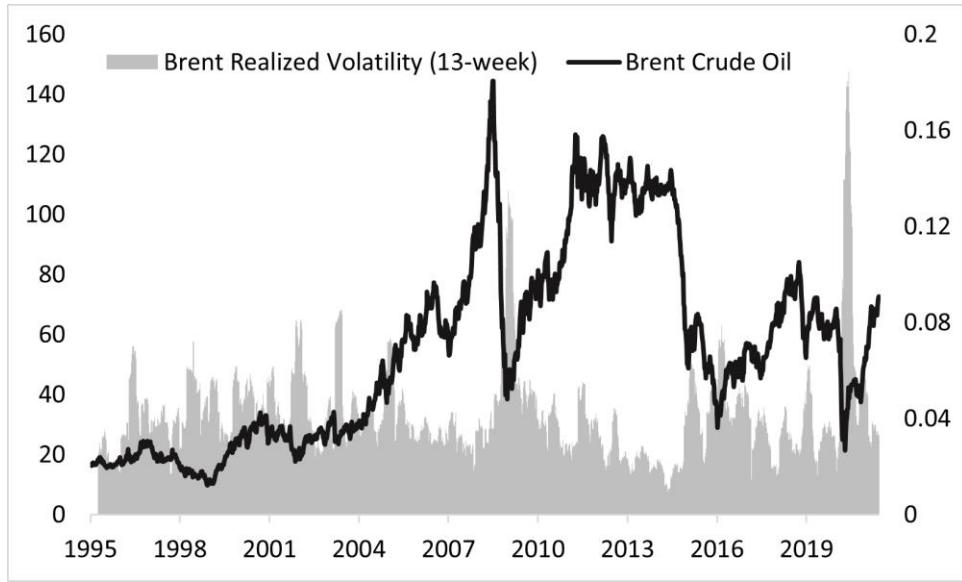
between 1979-1982 or 2007-2009 in the US, and thus controlling inflation. The regime shifts in foreign exchange often correspond to major adverse economic events and central bank interventions, such as during The Great Financial Crash (2008-2009) and Brexit (2016-2020) in the UK, which can be associated with spikes of GBPUSD 3M implied volatility shown in Figure 9, below.

Figure 9: GBPUSD and the 3M implied volatility on ATM GBPUSD options. Source: Bloomberg.



Finally, in commodities different regimes are usually linked to business cycles, since prices rally during expansions and fall concurrently with recessions due to a change in demand. Additionally, they are associated with major military conflicts. Figure 10 (on the following page) shows the price of Brent Crude Oil and 13-week (3-month) realised volatility. Clearly, from the figure, it can be seen that significant economic events, such as The Great Financial Crash and Covid Recession (2020) correspond to surging variance; however, smaller spikes can also be seen during the beginning periods of Afghanistan and Iraq conflicts between 2001 and 2004.

Figure 10: Brent Crude Oil prices and realised volatility. Source: Bloomberg.



The volatility-based measures shown in Figures 8, 9, and 10 that aim to detect when a regime switch occurred are, however, suboptimal in forming trading strategies. Mere eyeballing the chart is far from useful and applying certain thresholds, such as 30 in VIX<sup>6</sup>, may miss important market corrections, e.g., in 2015/2016 or 2018 (see Figure 8). Also, the volatility spikes appear *ex post* and are not indicative of any persistent behaviour, that is, it is difficult to assess whether a similar spike will take place in the next timestep. These limitations call for more sophisticated regime-switching detection techniques, such as Markov-switching regression model which has been considered as one of the most popular and effective tools to date, and is another key component in the work of the thesis, whose detailed workings will be described next, in Section 2.2.1.

---

<sup>6</sup> In the industry, the values of VIX higher than 30 are generally considered as more “fearful”, thus indicating a possible regime shift. However, as Figure 6 suggests, this is not ideal, since some market corrections are overlooked. Refer to (Whaley, 2009) for more details on how to interpret VIX.

## 2.2 Technical background

This section will describe a number of techniques and models. At this stage of the thesis, only one research chapter (Chapter 3) has been completed, and most of what will be described below will relate to that chapter. However, as outlined in the Introduction, it is planned to move from the nowcasting of Chapter 3 to forecasting with machine learning in Chapter 4, and, it is already known that Random Forest will be used as a main predictive algorithms, whereas the *Hidden Markov Model* (HMM) will serve as a benchmark method; thus, both Random Forest and HMM will also be described, in Section 2.2.4 on machine learning models for prediction that concludes this section.

### 2.2.1 Markov-switching regression model

The key to the Markov-switching regression model's widespread interest to researchers is its ability to incorporate characteristics of financial time series including fat tails, persistently occurring periods of turbulence followed by periods of low volatility, skewness, kurtosis, and time-varying correlations (Timmermann, 2000), (Ang & Timmermann, 2012). This gives a way to separate the underlying process  $y_t$  into  $S_t$  states ("regimes") at times  $t \in \{1, 2, \dots, T\}$  where  $S_t \in \{1, 2, \dots, k\}$ . Assuming the process  $y_t$  refers to daily log returns  $r_t$ , the model can be specified as

$$r_t = \mu_{S_t} + r_{t-1}\beta_{S_t} + \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1), \quad (1)$$

where  $\mu_{S_t}$  is a state-dependent intercept,  $\beta_{S_t}$  is a state-dependent coefficient of lagged log returns, and  $\sigma_{S_t}$  is a state-dependent volatility (Krolzig, 1997). With  $S_t \in \{1, 2\}$ , the governing dynamics of the underlying regime  $S_t$  are considered to follow a time-homogeneous Markov chain with fixed transition probabilities  $p, q \in [0, 1]$ ,

$$\begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix},$$

where

$$p = \Pr(S_t = 1 | S_{t-1} = 1), \quad (2)$$

$$q = \Pr(S_t = 2 | S_{t-1} = 2). \quad (3)$$

If  $(\delta, 1 - \delta)$  with  $\delta = \Pr(S_1 = 1) \in [0,1]$  is the initial distribution of the Markov chain, then the model is completely specified by the vector of parameters,  $\boldsymbol{\theta}$ :

$$\boldsymbol{\theta} = (p, q, \mu_0, \mu_1, \beta_0, \beta_1, \sigma_0, \sigma_1, \delta), \quad (4)$$

(Krolzig, 1997), (Hauptmann, Hoppenkamps, Min, Ramsauer, & Zagst, 2014), and the estimation of  $\boldsymbol{\theta}$  is performed by Gibbs sampling, a type of Markov chain Monte Carlo algorithm described in (Kim & Nelson, 1999).

If  $\widehat{\boldsymbol{\theta}}_t$  denotes the estimate of  $\boldsymbol{\theta}_t$ , then for the two volatility states  $i \in \{1, 2\}$  the transition probabilities at time  $t \in \{1, 2, \dots, T\}$  are given by

$$p_t^i = \Pr(S_t = i | \ln r_t; \widehat{\boldsymbol{\theta}}_t), \quad (5)$$

in which  $S_t = 1$  and  $S_t = 2$  are the low and high volatility periods, respectively.

$p_t^i = \Pr(S_t = i | \mathbf{r}_t; \widehat{\boldsymbol{\theta}}_t)$  estimates the probability of a low ( $i = 1$ ) or high ( $i = 2$ ) volatility regime at time  $t$ ; the higher the  $p_t^i$  the more likely the process is persistent (Krolzig, 1997), (Hauptmann, Hoppenkamps, Min, Ramsauer, & Zagst, 2014).

The model described is also known as the *two-state Markov-switching dynamic regression* (MSR) (Krolzig, 1997), and will be used throughout this thesis as a fundamental part of the novel regime-switching framework, as well as one of the models used for performance comparison. Figure 11 on the following page presents an example of a MSR application to MSCI USA Index.

Figure 11: Detecting high variance (red) and low variance (green) regimes in MSCI USA Index using the two-state MSR model.



Markov-switching regressions still base their regime-detecting abilities merely on changing variance. However, a period of low or high volatility may not necessarily indicate a rising or falling asset price for the entire duration of the same regime, particularly during transition periods when the variance is gradually increasing or decreasing (Guidolin & Timmermann, 2007), (Srivastava & Bhattacharyya, 2018). In addition, as can be seen in Figure 10, even though the general trend is bullish, such as between 2012 and 2016, it still contains periods of high variance which are just temporary. To identify the market's overall trend, outside of academia technical indicators have been broadly used, specifically different kinds of moving averages, as was described in Section 2.1.4. While there has been controversy in the academic community about the value of technical analysis, it will be shown in this thesis that one specific tool of technical analysis, *Kaufman's adaptive moving average* (KAMA), can substantially improve regime-switching detection.

### 2.2.2 Introduction to Kaufman's Adaptive Moving Average

As mentioned in Section 2.1.3, Kaufman's Adaptive Moving Average is a specific class of a moving average that attempts to balance the smooth-lag issue of moving averages by incorporating trend and volatility factors. KAMA has been constructed to closely track the low-swinging price during a calm period and to stay away from the price index if volatility increases, so that it avoids too-frequent crosses. Thus, only when there is a large change in volatility, KAMA will cross the price series. To achieve this, the calculation of KAMA begins with determining an *efficiency ratio* (ER). The efficiency ratio helps KAMA identify and adapt to ever-changing asset conditions by measuring the relative speed of price movement from one period to another. Assuming a daily frequency over time  $t \in \{1, 2, \dots, T\}$ , ER can be calculated as

$$ER_t = \frac{M_t}{V_t}, \quad (6)$$

where

$$M_t = |P_t - P_{t-n}|, \quad (7)$$

$$V_t = \sum_{i=1}^n |P_t - P_{t-i}|, \quad (8)$$

and where momentum,  $M_t$ , is the change in closing price  $P$  over a period of length  $n$  ( $n$ -period), and volatility,  $V_t$ , is the sum of the absolute value of daily closing price changes during an  $n$ -period. The efficiency ratio is constrained to  $0 \leq ER \leq 1$ , such that a significant price momentum as a proportion of low volatility will bring the ER closer to 1, whereas a minor price change as a proportion of high volatility will yield an ER closer to 0. In other words, values close to 1 indicate a clearly defined trend in the price, whereas values nearing 0 indicate a consolidating and directionless market. The ER is embedded into KAMA's full formula as a part of its scaled smoothing coefficient  $C$ . Considering again the daily frequency over time  $t \in \{1, 2, \dots, T\}$ , the adaptive moving average is computed as

$$KAMA_t = KAMA_{t-1} + C_t(P_t - KAMA_{t-1}), \quad (9)$$

where the scaled smoothing coefficient  $C$  over time  $t$  is expressed as

$$C_t = [ER_t(k_s - k_l) + k_l]^2. \quad (10)$$

Terms  $k_s$  and  $k_l$  in the above Equation 10 are smoothing constants relevant to a pair of different (short-term and long-term) simple moving averages over  $n$  period, and are calculated as

$$k_s = \frac{2}{n_s + 1}, \quad (11)$$

$$k_l = \frac{2}{n_l + 1}, \quad (12)$$

where  $n_s$  and  $n_l$  refer to shorter and longer time windows, respectively (Kaufman, 1995), (Ellis & Parbery, 2005); the selection of  $n_s$  and  $n_l$  is described later in the methodology section.

Figure 12: Example of KAMA (blue line) overlaid on S&P 500 Index.



By combining the trending aspect of moving averages and the volatility measurement provided by the efficiency ratio, KAMA is able to identify an overall price trend and accurately locate larger turning points (see Figure 12 above for an example application). Thus, the potential trader would not have to frequently switch positions by buying and selling over local price swings (Kaufman, 1995). This feature of KAMA is one it shares with Markov-switching regression models which

also seek to determine the current state the process is in with higher probability, to essentially avoid switching states too often. However, KAMA is primarily used to capture the overall trend, bullish or bearish, whereas Markov-switching regression model detects disruptions in variance of the underlying series. Since both trend and volatility are important in practical asset allocation, the combination of KAMA with the Markov-switching regression model holds promise and will indeed be demonstrated in Chapter 3 to lead to improved portfolio performance.

### 2.2.3 Limitations of the Markov-switching model

So far, a widely popular regime-detection technique has been discussed (in Section 2.2.1). As given in Equation 1, the model is able to predict one-step-ahead returns of a chosen asset by estimating the vector of parameters  $\theta$ . Subsequently, the forecasted returns are used to calculate the state occupation probabilities at given time  $t$  (see Equation 5). Effectively, this makes MSR an unsupervised learning technique for distinguishing financial regimes, as before the model's application, it remains unknown what state the time series has been in. This indicates that MSR does not focus on predicting regimes but rather on detecting them which may be problematic from the asset manager perspective, as detection usually lags the reality. Ultimately, we seek to be able to predict regime switches. It is plain that as MSR is sub-optimal in nowcasting regimes, it will not be able to cope well in forecasting regime switches. The simple NASDAQ example below explains the nature of MSR's nowcasting problem.

Before the so-called "Covid recession", NASDAQ had hit the peak on February 19<sup>th</sup> 2020 with an all-time high price of 9718.73. Around a month later, on March 20<sup>th</sup>, NASDAQ hit the trough with the price of 6994.29. This means there was a sharp decrease of -28.03% within a mere month. Running the two-state MSR up to February 19<sup>th</sup> confirms a low volatility regime, since the market simply kept rallying, as can be seen in Figure 13 on the next page, where the rightmost point corresponds to February 19<sup>th</sup>.

Figure 13: Application of the two-state MSR on NASDAQ log price up till February 19th, 2020.



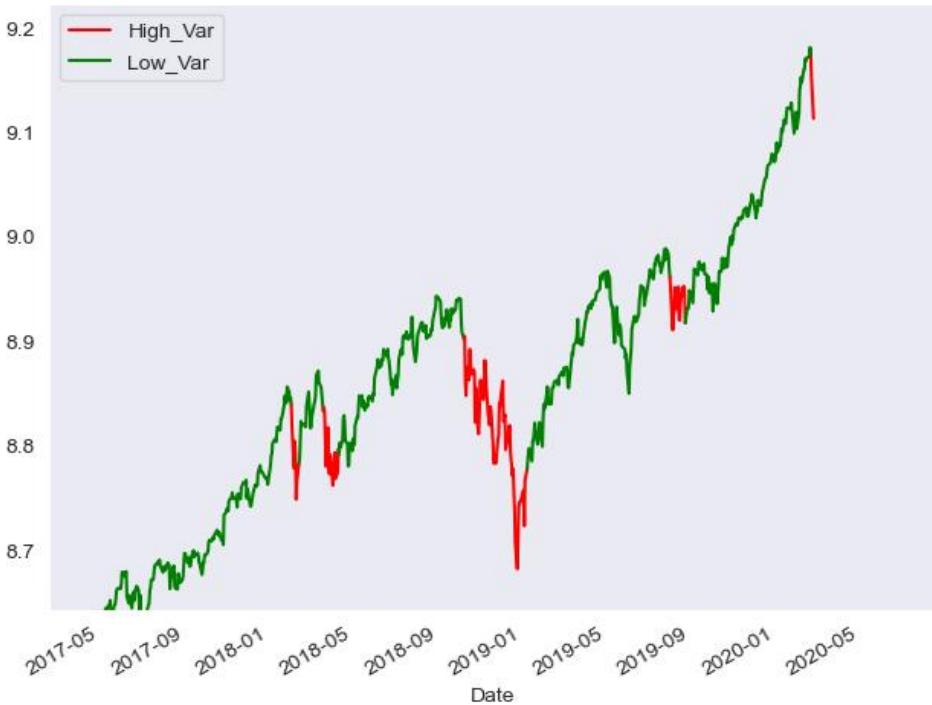
Four days later, on February 23<sup>rd</sup>, the MSR model still detected the low variance regime, even though the market had begun its sharp fall (Figure 14 on the following page, in which the rightmost point now corresponds to February 23<sup>rd</sup>).

Figure 14: Application of the two-state MSR on NASDAQ log price up till February 23rd, 2020.



Only on the 5<sup>th</sup> day after the peak (February 24<sup>th</sup>) did the two-state MSR confirm a regime switch from low to high variance, while correcting for couple of days between the peak and February 24<sup>th</sup> (Figure 15 on the following page, in which a single additional point has been added to Figure 14).

Figure 15: Application of the two-state MSR on NASDAQ log price up till February 24th, 2020.



The Markov-switching model inevitably suffers from this lag in its regime detection ability, even though it is able to detect regimes effectively, as will be shown in Chapter 3. A trader could potentially lag the data to wait for a signal confirmation, as can be seen on Figures 14 and 15; however, from the asset manager's perspective that could be inefficient, as the markets may have already collapsed. Even though, the significant amount of financial literature finds MSR to be of better use in lower frequency series (such as monthly) (Guidolin, 2011), the inherent lag still makes it more useful for the asset manager to predict these states ahead of time. The problem of forward prediction is solved by machine learning algorithms which are briefly outlined below.

#### 2.2.4 Machine learning models for regime prediction

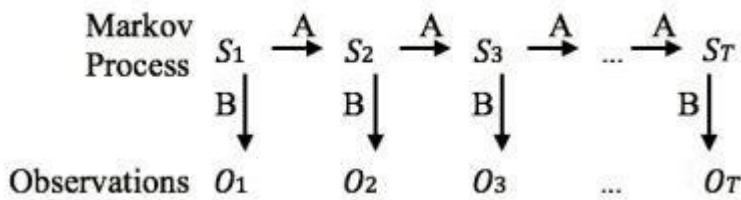
This section discusses technical backgrounds of two main machine learning algorithms that will be further explored in Chapter 4: Hidden Markov Model and

*Random Forest* (RF). HMM, due to its popularity in both industry and academia, will serve as a benchmark to Random Forest in regime prediction tasks; the reasoning for the choice of both methods is given in Section 2.3.2 of related work.

### *Hidden Markov Model: definition and comparison with the Markov-switching regression*

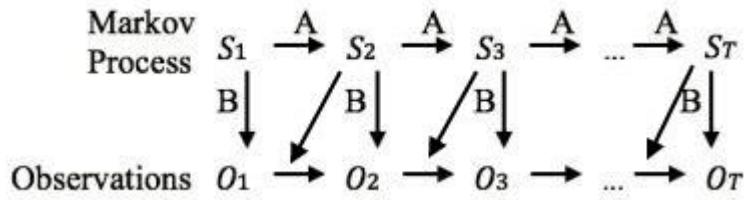
The *Hidden Markov Model* (HMM) is often confused with and interchangeably called the Markov-switching regression model (MSR) due to its application in forecasting regime shifts, although this thesis acknowledges main differences between both models. First, MSR is a generalisation of HMM which allows temporal dynamics within the regimes (hence the inclusion of a lagged variable in Equation 1) (Hamilton, 1989), (Krolzig, 1997). This leads to a notable difference between general structures of HMM and MSR (described in Section 2.2.1). Let  $A$  be the transition probability matrix where  $A = a_{ij} \dots a_{NN}$ , and each  $a_{ij}$  symbolises the probability of moving from state  $i$  to state  $j$ . Next, let  $B$  stand for the emission probabilities where  $B = b_i(O_t)$  which represents the probability of an observation  $O_t$  being generated from state  $i$ . Figure 16 shows the general structure of a Hidden Markov Model, where hidden state variables  $S_T$  and observed variables  $O_T$  serve as inputs to a learning algorithm which attempts to learn the parameters  $A$  and  $B$ .

Figure 16: A general structure of the HMM. Source: (Chen & Tsang, 2020).



As Figure 16 shows, the observed variable  $O_T$  only depends on the hidden state variable  $S_T$  at the same time  $t$ . This is in contrast to the MSR model which, due to inclusion of the lagged term (see Equation 1) inherently causes the observed variable  $O_T$  to be dependent on both its historical values and the hidden state variable  $S_T$ . This difference can be seen on Figure 17 on the following page.

Figure 17: A general structure of the MSR model. Source: (Chen & Tsang, 2020).



Finally, the estimation of the probability matrix and emission probabilities is done using a specific kind of an expectation-maximisation algorithm, namely a Baum-Welch algorithm<sup>7</sup>, which is an alternative to Gibbs sampling mentioned in Section 2.1.5. Beginning with computed initial estimates for the parameters  $A$  and  $B$ , the algorithm iteratively improves the probabilities during training, until it finds the transition probabilities ( $A$ ) and the emission probabilities ( $B$ ) that best fit the input data (Chen & Tsang, 2020). This learning procedure is very similar to training processes of machine learning algorithms thus it is safe to treat HMM as such, whereas MSR is closer to the science of econometrics (Nguyen, 2018). Indeed, some academics recognise this difference, such as in a recent book of (Chen & Tsang, 2020) which placed HMM in a machine learning section, or in (Rao & Hong, 2010) where HMM is labelled as an AI technique, or in (Krolzig, 1997) that linked MSR to econometrics and pointed at discrepancies in parameter optimisation algorithms between MSR and HMM, or finally in (Mikaeil, Bin, Xuemei, & Zhijun, 2014) where HMM and its characteristics were precisely separated from MSR. Thus, this thesis reaffirms this partition and will link HMM with machine learning algorithms for regime shifts, whereas MSR will only be associated with traditional regime-switching detection methods.

### *Hidden Markov Model: limitations*

Similarly to MSR's inability to immediately detect a regime change, HMM suffers from the inability to predict regime switches without experiencing the onset of the new regime, which is plainly not ideal: the HMM can in effect only predict

---

<sup>7</sup> For derivation, refer to (Baum & Petrie, 1966).

*continuations* of already-changed regimes. The application of the two-state HMM on NASDAQ shows almost the same results with a small difference in the number of days needed to detect the adequate regime. Figure 18 below shows low (in green), and high (in red) variance regimes as detected by the Hidden Markov Model in NASDAQ up until the February 19<sup>th</sup> peak.

Figure 18: Application of the two-state HMM on NASDAQ until February 19th.



Interestingly, once predicting the states after February 19<sup>th</sup> until February 24<sup>th</sup>, HMM still detects a bullish regime and changes the assumptions only after the next day, as can be seen in Figures 19 and 20 on the following page.

Figure 19: Forecasting NASDAQ regimes by the use of two-state HMM between February 19th and February 24th.

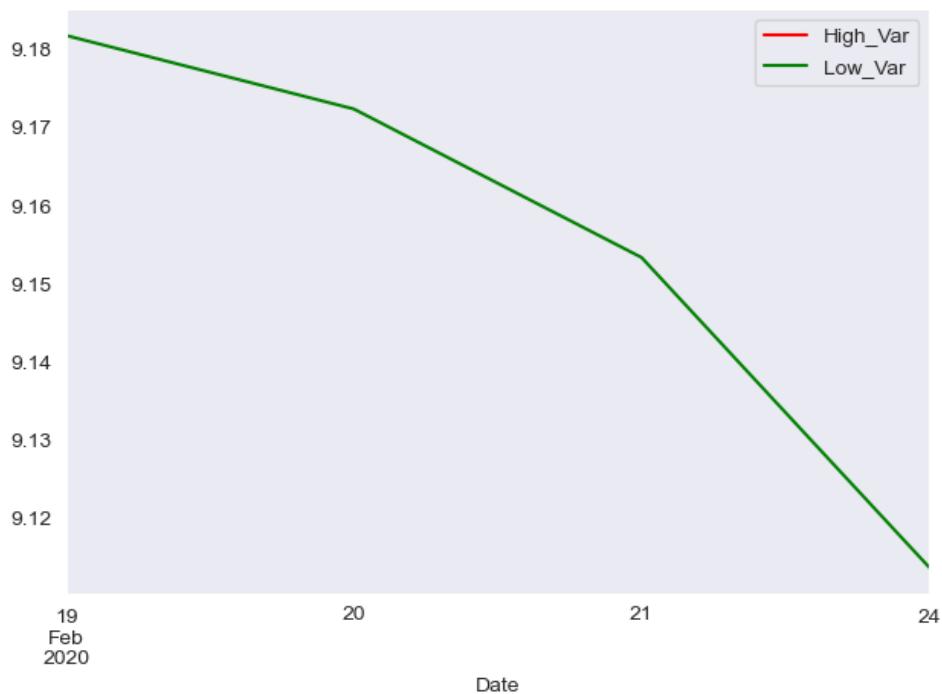
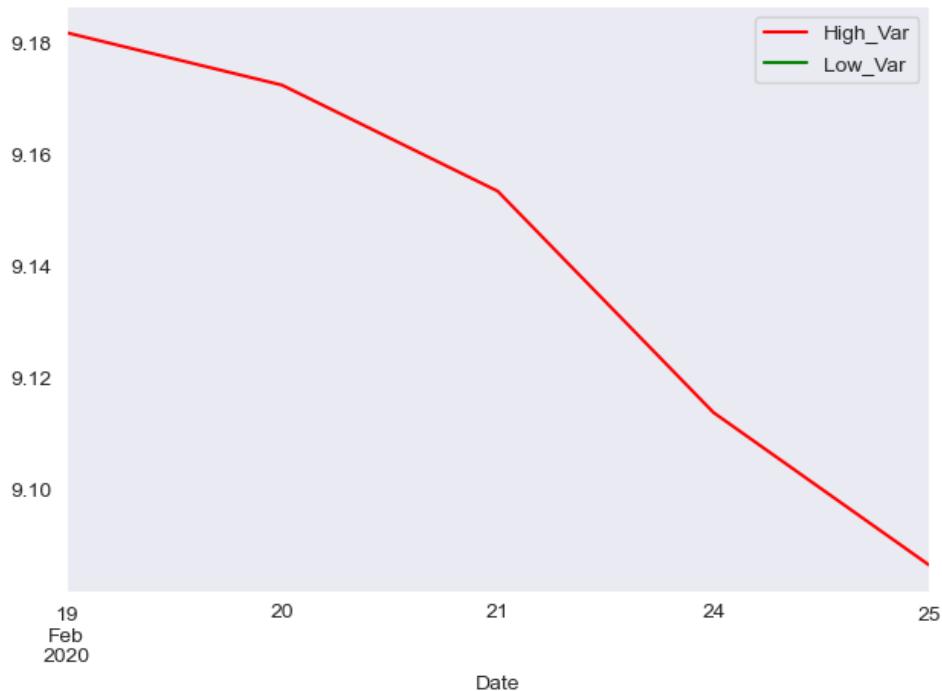


Figure 20: Forecasting NASDAQ regimes by the use of two-state HMM between February 19th and February 25th.



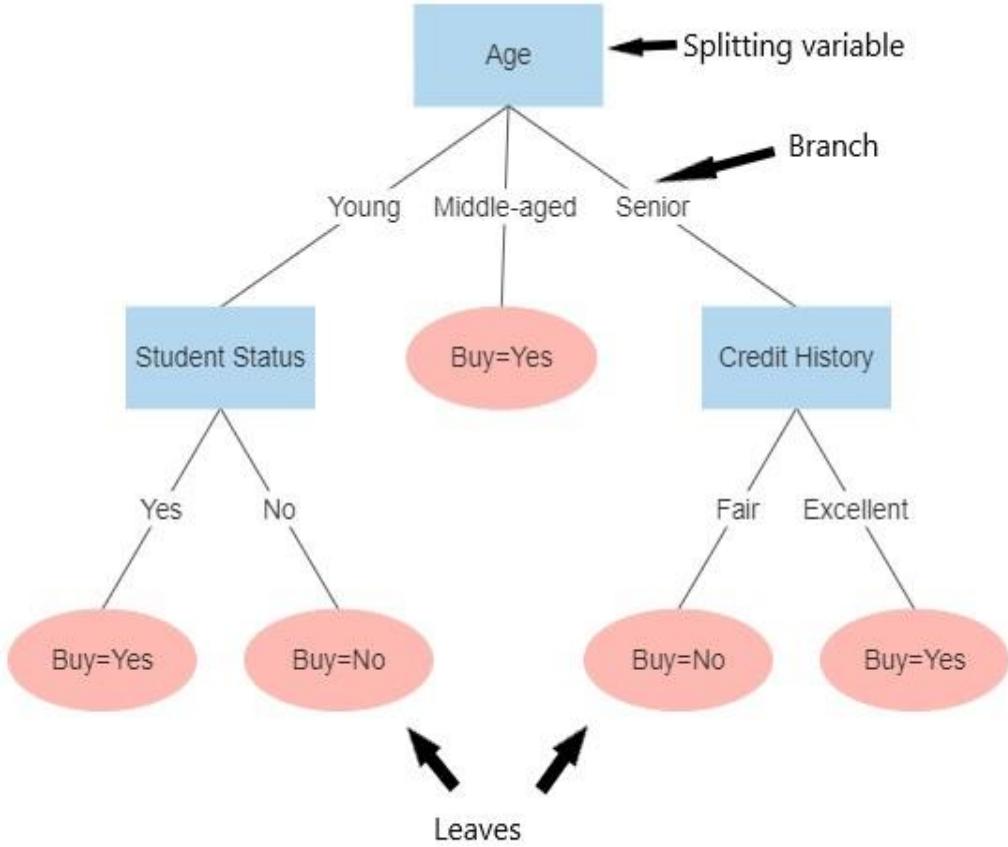
This shows that applying HMM in real world problems would also require lagging the data to avoid the look-ahead bias, or still be just used as a detecting, rather than regime predicting tool.

The problem of poor predictive capabilities of HMM can be solved by an alternative machine learning algorithm. This thesis has selected Random Forest for this task due to its superiority over competing methods in predicting financial regimes; the technical details of RF are discussed below, whereas the relevant applications are outlined in the Section 2.3.2 which describes machine learning approaches to predict regimes.

### *Random Forest*

**Theoretical background of Random Forest.** The fundamental component of the Random Forest algorithm is a single *decision tree* (DT). DT learning is a non-parametric supervised learning method used in data science for classification and regression tasks. One popular training algorithm which uses decision trees is the *Classification and Regression Tree* (CART). The aim of CART is to predict the value of a target variable by learning simple decision rules inferred from the input features. CART achieves this by recursively partitioning the training data through a series of binary splits of these predictors. Figure 21 on the following page shows a simplified CART containing decision nodes (made of splitting features), branches, and leaves. Since this thesis focuses in Chapter 4 on classifying states which describe regime prediction, the discussion here will henceforth assume any algorithms based on decision trees to be classifiers rather than regressors.

Figure 21: A simple decision tree which forms a CART. The target variable is the probability of a person buying a computer which depends on the given splitting variables: person's age, student status, and credit history.



The variable split that occurs in a decision node is done by asking a question about this feature (by generating subsequent branches, depending on the answer); in Figure 21 above this can be whether the person is young or middle-aged, as well as whether they are a student or not. The choice of how the features of a dataset should split decision nodes to form the tree is based on impurity measures, most commonly *Gini Impurity* (GI) and *Shannon Entropy* (SE).

GI is a number between 0-0.5 that indicates the likelihood of new, random data being misclassified if it were given a random class label according to the class distribution in the dataset. The GI of a node  $D$  is defined as

$$GI(D) = 1 - \sum_{i=1}^n p_i^2, \quad (13)$$

where  $n$  is the number of classes and  $p_i$  is the probability of samples belonging to class  $i$  at a given decision node. The lower the Gini Impurity (where ideally  $GI(D) = 0$ ), the higher the likelihood that all samples belong to the same class in a decision node  $D$ .

As mentioned above, the Shannon Entropy is another function that can be used to assess the quality of a split. It effectively measures the unpredictability in the information contained in a specific decision node of a tree; it is defined as

$$SE(D) = - \sum_{i=1}^n p_i \log_2 p_i, \quad (14)$$

where  $p_i$  is the probability of samples belonging to class  $i$  at a given decision node. Similarly to GI, SE is highest ( $SE(D) = 1$ ) when all the classes are contained in equal proportion in the node  $D$ , whereas it is lowest ( $SE(D) = 0$ ) when the decision node  $D$  is pure, that is, it contains only one class.

The decision to choose the optimal feature to split the node is made on maximizing the *information gain* (IG) which is calculated as the impurity of a parent node subtracted by a sum of impurities of child nodes. For instance, if a given node  $D$  is split on a random feature into two further nodes  $D_1$  and  $D_2$ , with sample sizes  $s_1$  and  $s_2$ , respectively, the IG of  $D$  can be defined as

$$\Delta IG(D) = IM(D) - \left( \frac{s_1}{s} IM(D_1) + \frac{s_2}{s} IM(D_2) \right), \quad (15)$$

where  $IM$  is a selected impurity measure (e.g., GI or SE) (Khaidem, Saha, & Dey, 2016).

Even though decision trees are simple to understand and interpret, as well as requiring little data preparation, the CART algorithm is prone to overfitting due its low bias and high variance<sup>8</sup>. In addition, a CART learner is very sensitive to even slight noise in the data which can cause it to grow in a completely different manner

---

<sup>8</sup>The bias-variance trade-off is a common issue in machine learning algorithms: methods with a good in-sample fit (low bias) tend to poorly generalize on the holdout sample (high variance). As the bias is increased, the variance decreases; however, a high bias can cause an algorithm to miss the relevant relations between features and target outputs, which leads to underfitting. Thus, an optimal machine learning method should balance the bias and variance to avoid both underfitting and overfitting.

every time it is created. Both issues can be tackled by forming an ensemble algorithm which incorporates more than one decision tree. One of such ensemble method is bootstrap aggregating, also known as bagging, which trains a number  $C$  of decision trees and then averages the class predictions of each tree to arrive at the final prediction. This significantly reduces the variance, thus, at least partially, solves the issue of overfitting in decision trees learning (Breiman, 1996).

However, the standard bagging approach may still suffer from potentially high correlations across the individual decision trees, since they are trained on overlapping bootstrap training samples. This in turn decreases the variance improvements, thus reducing the benefit of bagging. As a result, a significant extension to bootstrap aggregating has been introduced to tackle the high correlation among the bagged trees without enlarging the bias: the Random Forest (RF) algorithm.

RF effectively de-correlates individual classification trees by randomly selecting the subset of features that create each tree. While the standard bagging approach searches over the entire set of predictors  $k = 1, \dots, N$  for the best splitting feature at each decision node  $D$ , RF instead randomly selects  $Q \ll N$  predictor features at such node. Subsequently, the algorithm searches only over these randomly chosen features for the splitting variable. Such a randomised CART is created  $C$  times to generate the equivalent number of predictions that are then averaged for one final prediction, similarly to the training procedure of the standard bagging algorithm (Breiman, 2001). On the following page is the pseudocode describing the Random Forest algorithm in detail:

Table 1: Pseudocode of a Random Forest algorithm. Source: (Khaidem, Saha, & Dey, 2016).

**Algorithm: Random Forest Classifier**

```
1: procedure RandomForestClassifier( $\mathcal{B}$ ) ->  $\mathcal{B}$  is the labelled dataset
2:    $forest = newArray()$ 
3:   for  $i = 0$  to  $C$  -> Bootstrap aggregation
4:      $\mathcal{B}_i = \text{Bagging}(\mathcal{B})$ 
5:      $T_i = \text{new CART}()$ 
6:      $features_i = \text{RandomFeatureSelection}(\mathcal{B}_i)$ 
7:      $T_i.\text{fit}(\mathcal{B}_i, features_i)$ 
8:      $forest.append(T_i)$ 
9:   end for
10:  return  $forest$ 
11: end procedure
```

Random Forest learns from the data with the help of multiple hyperparameters. The paragraph below outlines these more and less relevant.

**Hyperparameters of Random Forest.** Random Forest has 14 hyperparameters whose values are selected to train the algorithm. However, not every hyperparameter is equally important; a common approach, supported by recent work (e.g., (López de Prado, 2018), (Probst, Wright, & Boulesteix, 2019)), is to focus on those with the largest impact on the accuracy of the trained model, which is about a half of the total number. Table 2 on the following page lists all the hyperparameters available in the scikit-learn<sup>9</sup> package used for the training of machine learning models in this thesis, and highlights those that will be tuned in Chapter 4; the non-tuned hyperparameters will assume their default values which will also be given in Chapter 4.

---

<sup>9</sup> Scikit-learn is one of the most popular machine learning packages available for Python. Refer to <https://scikit-learn.org/stable/> for further reading. Accessed on 06/09/2022.

Table 2: Hyperparameters of Random Forest. Source: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>. Accessed on 06/09/2022.

Hyperparameter	Description	Tuned?
<b><i>n_estimators</i></b>	Number of trees used to build one forest.	Yes
<b><i>max_depth</i></b>	The maximum depth of a single tree.	Yes
<b><i>min_samples_split</i></b>	The minimum number of samples required to split an internal node.	Yes
<b><i>min_samples_leaf</i></b>	The minimum number of samples required at a leaf node (see Figure 21 for leaves).	Yes
<b><i>min_weight_fraction_leaf</i></b>	The fraction of the input samples required to be at a leaf node where weights are determined by sample weights.	Yes
<b><i>max_features</i></b>	The number of features to consider when looking for the best split.	Yes
<b><i>max_samples</i></b>	The number of samples to draw from the training set to train each base estimator.	Yes
<b><i>criterion</i></b>	The function to measure the quality of a split. This could be Gini Impurity (Equation 13) or Shannon Entropy (Equation 14).	No – the default choice, Gini Impurity, will be used in Chapter 4.
<b><i>class_weight</i></b>	Weights associated with each class to tackle imbalanced datasets.	No – balanced class weights will be used in Chapter 4.
<b><i>max_leaf_nodes</i></b>	The number of leaf nodes in a tree. This hyperparameter sets a condition on the splitting of the nodes in the tree and hence restricts the growth of the tree.	No – the default value, an unlimited number, will be chosen in Chapter 4. (Note that the complexity of the tree is already restrained by <i>max_depth</i> .)

<b><i>min_impurity_decrease</i></b>	An internal node can only be split if this split induces a decrease of the impurity greater than or equal to this value.	No – the default value, 0.0, will be chosen in Chapter 4.
<b><i>bootstrap</i></b>	The use of bootstrap samples to build a tree. If False, then the entire training set is used to construct a tree.	No – the default value, True, will be chosen in Chapter 4.
<b><i>oob_score</i></b>	The use of out-of-bag samples to estimate the validation score. Only available for bootstrapped samples.	No – the default value, False, will be chosen in Chapter 4.
<b><i>ccp_alpha</i></b>	Complexity parameter used for Minimal Cost-Complexity Pruning. The subtree with the largest cost complexity that is smaller than <i>ccp_alpha</i> will be chosen.	No – the default value, 0.0, will be chosen in Chapter 4.

There have been many successful applications of Random Forest to financial time series over time. The next section will show evidence of RF being an optimal method for regime prediction tasks. In addition, it will discuss other machine learning methods that can be useful in predicting regimes ahead of time (though less well than RF), as well as techniques related to regime detection for label generation. However, before delving into related work, a concise overview of concept drift, an important factor affecting the performance of machine learning models, will be discussed.

## 2.3 Related work

As described in Section 2.1.3, which discussed the QTS portfolio building pipeline, a decision as to a manner of generating alpha, as well as constructing the portfolio, must be made. In the financial literature there are multiple ways by which both stages can be successfully executed. Since the work of this thesis is focused on regime-switching, this section will outline both common and less popular methods of detecting and predicting regimes that can be helpful in generating alpha, the former for label generation, the latter for signal prediction. The first two sections focus on the regime detection applications of the Markov-switching model (2.3.1), and the regime prediction capabilities of Random Forest (2.3.2), since this thesis will pursue these topics further in Chapters 3 (detection) and 4 (prediction). The following Section 2.3.3 will briefly describe techniques used for portfolio allocation, a critical component of QTS, as well as elaborate on an ongoing dispute of adding value to constructing portfolios through technical analysis.

### 2.3.1 Regime detection methods

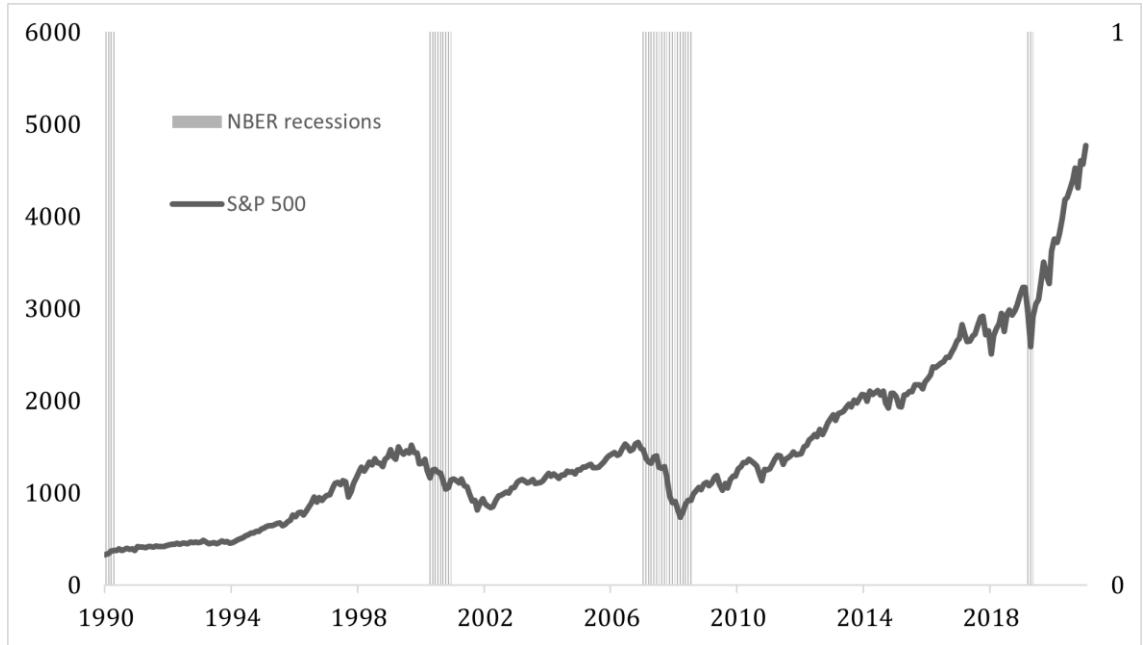
Regime prediction, the ultimate goal of the work of this thesis, whether using classic or machine learning techniques, is specified as a supervised learning problem. (Usually binary because binary models are more stable than continuous models with regards to predicting regimes (Estrella, Rodrigues, & Schich, 2003).) But prediction first requires adequate labels for different states, such as recessions, i.e., a process of regime detection. One possible approach considers taking traditional NBER<sup>10</sup> business cycle dates to mark US economic recessions over time. However, NBER recessions strictly follow macroeconomic activity which does not fully mirror market movements, as can be seen in Figure 22 on the following page. In addition, they are limited to the US economy, leaving, for

---

<sup>10</sup> See <https://www.nber.org/research/business-cycle-dating> for details on NBER business cycle dating. Accessed on 04/03/2022.

instance, European states or emerging markets behind, though for these countries a more trivial method can be applied, such as identifying periods with a consecutive two-quarter decline in GDP (Filardo, 1999).

*Figure 22: NBER recessions vs. S&P 500 performance. Source: Bloomberg.*



Another method, which has become an active research topic due to its ability to correct for NBER limitations, utilises available regime detection methods. Such techniques generate labels for regime switches that can be subsequently predicted ahead using various traditional and machine learning techniques (Mullainathan & Spiess, 2017), (Uysal & Mulvey, 2021). This section will outline both traditional and machine learning approaches to detect regimes; the former are dominated by the Markov-switching model (the chosen method for this thesis) and the (G)ARCH framework, whereas the latter usually revolve around clustering or  $\ell_1$ -filtering. It's worth noting that while some models described next, such as (G)ARCH, may not strictly be considered regime-switching models, they can still be treated as such due to their ability to analyse and detect changes in volatility, which is often an important aspect of regime detection.

### *Traditional regime detection methods*

**Markov-switching regression model (MSR).** (See Section 2.2.1 for technical details on MSR.) The Markov-switching regression model (see Section 2.1.5), introduced by (Goldfeld & Quandt, 1973), was initially applied (in its basic form) to the US housing market; however, its popularity spiked years later after extending the original formula with an autoregressive component and showing its state detection capabilities on US and international GDP, which allowed the model to capture business expansions and recessions (Hamilton, 1989), (Krolzig, 1997). Subsequently, MSR was applied to other macroeconomic factors, such as inflation (Kim C.-J. , 1993) and US, German, and UK interest rates (Garcia & Perron, 1996), (Ang & Bekaert, 2002), as well as providing evidence of regime changes in the US monetary policy (Sims & Zha, 2006).

While the original work of (Hamilton, 1989) focused on the traditional assumption of exogenous switching<sup>11</sup>, (Kim, Piger, & Startz, 2003) introduced a subclass of an MSR model that allows for the endogeneity of the latent state variable controlling regime change. The Endogenous Regime Switching model relaxes the assumption of exogenous switching and presents a parsimonious model of endogenous Markov regime-switching. Inference through maximum likelihood estimation is feasible with minor adjustments to existing recursive filters, with the model encompassing the exogenous switching model and enabling straightforward tests for endogeneity. Monte Carlo experiments demonstrate the accuracy of maximum likelihood estimates in the presence of certain model misspecifications, highlighting the effectiveness of this model in capturing endogenous regime changes, for example, in US equities (Kim, Piger, & Startz, 2003).

In financial markets, the class of the Markov-switching regression models has been widely applied to equity returns, both in the US (Schaller & Van Norden, 1997), and in Western and Central Europe (Bialkowski, 2004), (Moore & Wang, 2007), as well as in emerging markets (Assoe, 1998). Abundant evidence confirms MSR to be

---

<sup>11</sup> The assumption of exogenous switching in Markov-switching regression models is based on the belief that the latent state variable controlling regime shifts is independent of all realizations of the regression disturbance. This assumption implies that the state variable is not influenced by any observations of the explanatory variables, maintaining a level of exogeneity throughout the model (Kim, Piger, & Startz, 2003).

a robust regime detection method. Not only does it efficiently discover state shifts in various equity markets, but it also paves the way for further analysis; for instance, MSR allows the discovery of changing correlations during different regimes between equities and other asset classes, such as bonds (Bansal, Connolly, & Stivers, 2010), currencies, or oil prices (Roubaud & Mohamed, 2018).

MSR was also successfully applied to foreign exchange, whether US dollar (Engel & Hamilton, 1990), European currencies and Japanese yen (Jeanne & Masson, 2000), (Dueker & Neely, 2007), or emerging market currencies (e.g. BRIC<sup>12</sup>) (Das & Roy, 2021). All works stressed the model's ability to identify structural breaks and indicate currency crises, which can potentially be used to create trading strategies. Similar conclusions were drawn from commodity markets, such as oil (both WTI and Brent crudes), gold, silver, and copper (Choi & Hammoudeh, 2010). Recently, MSR has also been tested as a useful tool to detect state shifts in cryptocurrencies, namely Bitcoin (Chappell, 2019). In line with other asset classes, the Markov-switching model has also been able to detect in Bitcoin price data variance clustering, jumps, and asymmetric transitions, as well as infer the persistence of shocks. Finally, MSR has been described in the literature as a helpful method to construct regime-based portfolios, a topic that will be more thoroughly discussed in Section 2.3.3 on portfolio allocation methods.

The above-described examples point specifically at the MSR model's robustness and potential universal use, regardless of the asset class. Due to its efficiency and long history of application to various financial markets, this thesis will work with MSR, in a way which is outlined in more detail in Chapter 3; however, it is worth mentioning below other traditional regime switching methods which have been applied to financial time series with limited success.

**(Generalised) AutoRegressive Conditional Heteroskedasticity model (GARCH).** The *AutoRegressive Conditional Heteroskedasticity* (ARCH) model has been widely used in finance to estimate the volatility of an underlying instrument. Even though, as was mentioned previously, (G)ARCH framework is not a regime-

---

<sup>12</sup> BRIC stands for Brazil, Russia, India, and China.

switching model explicitly, this method is able to model the change in variance over time, thus able to capture different regimes in time series, such as market crashes, during which the volatility tends to cluster (see Section 2.1.4). Originally introduced by (Engle, Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation, 1982), the ARCH model uses the weighted average of the past squared forecast error to estimate dynamic volatility. The ARCH model was later generalised by further adopting past squared conditional volatilities (Bollerslev, 1986). This method, labelled *Generalised AutoRegressive Conditional Heteroskedasticity* (GARCH), improved the overall in-sample performance of the ARCH method, although it still has serious limitations, such as failure to accurately capture structural breaks, inability to explain all asymmetries of the financial data, prediction of greater persistency of volatility than is the case, short memory, or poor out-of-sample performance (Marcucci, 2005), (Bauwens, Preminger, & Rombouts, Theory and inference for a Markov switching GARCH model, 2010), (Škrinjarić & Šego, 2016). To tackle some of these problems (e.g. asymmetry and memory issues), several extensions were proposed, such as *Exponential GARCH* (EGARCH), which prevents negative variance (Nelson D. B., 1991), *Threshold GARCH* (TGARCH), which allows for different responses on positive and negative shocks (Zakoian, 1994), or *Factionally Integrated GARCH* (FIGARCH), which captures long memory patterns (Baillie, Bollerslev, & Mikkelsen, 1996). The academic literature evidences at least 14 main GARCH extensions (and a plethora of additional enhancements) developed through time. Interestingly, none has been found to outperform the standard GARCH model when applied to foreign exchange, although there is strong evidence that the standard GARCH model is inferior relative to its main extensions when applied to US equities (Hansen & Lunde, 2005).

Even though the GARCH model has been applied to other financial time series, such as international equities (Franses & Van Dijk, 1996), (Gokcan, 2000), commodities (Bernard, Khalaf, Kichian, & McMahon, 2008), or foreign exchange (West & Cho, 1995), there is sufficient evidence in literature that the Markov-switching model is more efficient than the standard GARCH model. This is primarily due to its capturing key features in financial time series that the GARCH model fails to (e.g., switching variances, differences in distribution of higher moments of returns over time) (Timmermann, 2000), (Choi & Hammoudeh, 2010), (see also

Section 2.2.1). One solution to tackle this gap between Markov-switching regression and GARCH was to introduce a blend of both models, called MS-GARCH, which extended the standard GARCH framework with the possibility of sudden jumps from one volatility regime (e.g., high) to the other (e.g., low). However, this method has been found to underperform the standard GARCH technique in estimating the long-term volatility of US stock market data (Marcucci, 2005), though, on the other hand, there has been significant evidence of the high utility of MS-GARCH in terms of asset allocation, a topic which will be discussed further in Section 2.3.3 on portfolio allocation methods.

Regardless of their limitations, which overall led to the decision not to use them in the work of this thesis, GARCH models remain a popular method of regime detection. In addition, GARCH models have been constantly tweaked to improve their utility, a good example of which could be a recent work on the Heterogeneous AutoRegressive model, which has been applied with success to high-frequency data (Wilms, Rombout, & Croux, 2021), a topic not covered by this thesis.

**Change point detection.** In contrast to regime switching, which involves the data evolving between different states or regimes continuously, change point detection techniques focus on identifying specific points in time where abrupt shifts occur in the underlying structure of the data. These methods can be broadly categorized into online and offline approaches, where offline change point detection analyses the entire dataset at once to detect change points retrospectively, while online methods continuously monitor incoming data to detect changes in real-time.

Literature describes various statistical tests that have been utilised in both online and offline change point detection to identify change points, including the CUSUM tests (e.g., (Page, 1954), (Brodsky & Darkhovsky, 2013)), the Chow test (Chow, 1960), end-of-sample cointegration breakdown test (Andrews & Kim, 2003), the subspace identification (Kawahara, Yairi, & Machida, 2007), the multiple linear regression approach (Bai & Perron, 2003), the likelihood ratio tests (e.g., (Hinkley, 1970), (De & Mukherjee, 2022), (Habibi, 2022)), and Bayesian methods like Bayesian Change Point Analysis (Adams & MacKay, 2007).

In economics and finance, change point detection techniques have various applications such as, similarly to the regime switching methods, identifying shifts in market regimes (Niu, Hao, & Zhang, 2016), detecting structural breaks in non-stationary (Davis, Lee, & Rodriguez-Yam, 2006), as well as strictly financial time series (Dias & Embrechts, 2002), and monitoring changes in asset volatility (Habibi, 2022).

One advantage of change point detection over regime switching is its ability to pinpoint specific moments of structural changes in the data, providing more precise information about when these changes occur. However, regime switching models offer a more flexible framework for capturing continuous transitions between different states, which can be advantageous when dealing with complex and evolving systems (Jedelyn, Adolf, Tuerlinckx, Kuppens, & Ceulemans, 2018). This thesis will not pursue change point detection techniques any further; however, they may be interesting to consider for use in future work.

**Other regime detection models.** Apart from the Markov-switching regression and GARCH models, the literature primarily mentions three other frameworks capable of detecting regime shifts: the *Threshold Autoregressive* (TAR) model (with extensions, such as the *Self-Exciting Threshold Autoregressive* (SETAR) model), the *Dynamic Conditional Correlation* (DCC) model, and the *Directional Change* (DC) model. This thesis will not discuss them beyond the brief summaries below; however, they may be interesting to consider for use in future work.

The TAR model is a set of time series regression models where the predictors are associated with the outcome in a threshold-dependent way. By introducing a threshold parameter, TAR can model certain kinds of non-linear relationships between the outcome and a predictor. TAR primarily aims at estimating structural breaks within a time series which potentially serve as signals of a regime change. In particular, the SETAR model, an extension of the TAR model, allows for more flexibility, as the threshold parameter is allowed to vary over time, which can capture more complex patterns in the data; this also makes the SETAR model more suitable for detecting abrupt regime shifts. However, Threshold Autoregressive

models have not been widely used in practice since the threshold variable cannot be identified with certainty, the modelling procedures available are too complex, and the prior knowledge of major economic events could lead to a bias in inference (Tong & Lim, 1980), (Tsay, Testing and Modeling Threshold Autoregressive Processes, 1989), (Procacci & Aste, 2019).

DCC models are a specific type of multivariate GARCH framework built upon the *Constant Conditional Correlation* model, which assumed volatilities to be time-varying but conditional correlations to be constant; this is also why it is important to note that DCC models are not regime-switching framework explicitly but can be used for such studies. DCC changes the latter assumption, which allows the description of dynamic correlations between asset returns over time (Engle, 2002). The reason why DCC is treated as a separate traditional regime detection technique instead of one of the GARCH extensions is that DCC has become a separate branch of multivariate GARCH models; the baseline model has been extended in multiple directions, dealing with parametrisation, asymmetry, and alternative estimation methods (Caporin & McAleer, 2013). DCC is particularly useful for asset allocation, as its application is relatively simple and it is possible to work with large portfolios (Fiszeder & Fałdziński, 2019). However, the literature also mentions the utility of DCC in terms of detecting regimes, primarily after blending with the Markov-switching model (Pelletier, 2006). The *Regime Switching Dynamic Correlation* (RSDC) model has been shown to be a helpful method when applied to discovering contagion effects during adverse market regimes (Rotta & Valls Pereira, 2016) or working with asymmetric and fat-tail stock returns (Paoletta, Polak, & Walker, 2019). RSDC and its extensions will not be discussed in this thesis further, mainly due to the fact that their enhanced performance is caused by the Markov-switching factor.

DC was initially introduced as an alternative way to sample data; instead of focusing on fixed-time windows (i.e., an asset price change of a month or a week), DC concentrates on the size of a price change that occurs between price peaks and troughs, thus separating time series into irregular states. This can effectively separate underlying time series into a bullish (trough-to-peak) and bearish (peak-to-trough) regimes which can subsequently be used in trading strategies. However,

DC is particularly sensitive to the percentage change factor that indicates by how much the price must move between peak to trough (or trough to peak); a too low percentage rate introduces overly-frequent whipsaws, whereas a too large one may miss periods of smaller, although significant, slumps and rallies which can prove to be costly for asset managers. In addition, Directional Change is limited to two states, which may be problematic when it is applied to asset prices with long transitory regimes (i.e., the potential third state) (Chen & Tsang, 2020).

Traditional regime detection methods remain popular in the academic literature due to their relative ease of use and a long history of successful applications; this is particularly so in the case of the Markov-switching framework, which can also be applied without an attempt to predict ahead to build vanilla regime-based portfolios (see Section 2.3.3). However, as computing power has increased over the years, machine learning solutions have gained significant attention, which has also presented opportunities for novel approaches to regime switching detection. Two particularly popular machine learning methods are thus briefly discussed below: clustering and  $\ell_1$ -filtering, although due to their limitations in application to detecting regime shifts in financial time series, they will not be used in the work of this thesis. In addition, it is again important to point out that these frameworks can be used to detect regimes while they are not explicitly regime detecting methods.

### *Machine learning regime detection methods*

**Clustering.** In machine learning, clustering is a form of unsupervised learning, where the user has no prior knowledge of data patterns or labels. Time series clustering has been applied in multiple scientific domains, such as economics, finance, or engineering, in order to identify homogenous groups of series with common features that make them distinct from other groups. In addition, clustering can be used for exploratory data analysis, summary generation, and as a pre-processing technique for other data mining tasks, such as outlier detection (Aghabozorgi, Shirkhorshidi, & Wah, 2015). Popular clustering algorithms are K-Means, agglomerative (hierarchical) clustering, affinity propagation, spectral

clustering, DBSCAN, and BIRCH<sup>13</sup>. In finance, K-Means and hierarchical clustering are the most commonly applied; in the case of K-Means this is due to the algorithm's simplicity, speed, and robust performance (Babu, Geethanjali, & Satyanarayana, 2012), (Bini & Mathew, 2016), whereas hierarchical clustering, even though computationally expensive, has shown much use in constructing risk parity portfolios (more details will follow in Section 2.3.3) (López de Prado, Building Diversified Portfolios that Outperform Out of Sample, 2016). Notably in the context of this thesis, K-Means will be applied in Chapter 3 as a helper algorithm to optimise the novel regime detection framework.

Clustering algorithms can also be effectively used in studying regime-switching patterns. The literature mentions a successful application of clustering to discover state shifts in artificial datasets (Oates, Firoiu, & Cohen, 1999), (Samé, Chamroukhi, Govaert, & Akni, 2011), as well as real-world data, such as energy consumption, (Samé, Chamroukhi, Govaert, & Akni, 2011), (Iglesias & Kastner, 2013), wind regimes (Dias, Vermunt, & Ramos, 2015), or electric vehicles (Hallac, Vare, Boyd, & Leskovec, 2017). Clustering methods have also been used to discover different regime patterns in economics and finance. For example, clustering can be effectively used with retail data to discover seasonal patterns (Kumar, Patel, & Woo, 2002). In addition, clustering algorithms can be helpful in determining the timings of transitions between different regimes in inflation and unemployment (Ormerod, Rosewell, & Phelps, 2013). In finance specifically, clustering is often applied to decrease risk or improve portfolio returns, a topic that will be discussed further in Section 2.3.3. However, there have also been attempts to use clustering as a method to detect regime switches in stock returns. The literature reveals that this can be achieved by either blending clustering with the Hidden Markov Model (Dias, Vermunt, & Ramos, 2015) or establishing a novel method capable of classifying time series into different states with the use of, for instance, inverse covariance matrices, a method known as *Inverse Covariance Clustering* (ICC) (Procacci & Aste, 2019). ICC clusters financial time series via a maximum-likelihood model that uses a likelihood measure adjusted for temporal coherence. Even though the model shows high

---

<sup>13</sup> For details on each of these algorithms, refer to <https://scikit-learn.org/stable/modules/clustering.html>. Accessed on 04/03/2022.

efficiency in-sample, the analysis has been limited only to two regimes (bullish and bearish), as well as out-of-sample results show lower true negative ratio<sup>14</sup>, indicating over-assigning the more frequent bullish class to the bearish class. Other methods, such as a sliding window-based multi-stage clustering applied to NASDAQ price data (Ren, Wei, Cui, & Du, 2017), detect change in market trends via a pattern discovery; however, this method focuses on short term trends and does not provide any useful information about longer term trends. In addition, the above framework does not distinguish multiple regimes, rather just discovers upward and downward change in a trend. Finally, another recent approach (Bilokon, Jacquier, & McIndoe, 2021) proposes a data-driven clustering framework based on the path signatures that encode financial time series into easy-to-describe objects; however, the results have been limited to synthetic market data without showing real-world performance.

Clustering algorithms are not the only machine learning method capable of detecting regimes. Another interesting technique,  *$\ell_1$ -filtering*, has also been applied to financial datasets, though not without constraints.

**$\ell_1$ -filtering.**  $\ell_1$ -filtering is a machine learning method used to detect changes in a trend, which was originally built upon a similar technique called  $\ell_2$ -filtering (also known as Hodrick–Prescott Filtering).  $\ell_2$ -filtering has been more useful in economics to identify slow changes in, for instance, GDP, whereas  $\ell_1$ -filtering has been specifically developed to aid in detecting sharp trend variations in financial time series, such as equities (Kim, Koh, Boyd, & Gorinevsky, 2009), (Dao, 2014). The difference between the methods lies in the type of penalty function ( $L_1$  for  $\ell_1$ -filtering and  $L_2$  for  $\ell_2$ -filtering)<sup>15</sup> used to control the competition between the smoothness of the trend and the noise factor (i.e., the difference between the underlying curve and the trend).  $\ell_1$ -filtering, through its ability to produce smooth

---

<sup>14</sup> True negative ratio (TNR) is calculated as  $TNR = \frac{TN}{TN+FP}$ , where TN stands for true negative and FP for false positive.

<sup>15</sup> In  $\ell_2$ -filtering, the trend estimate  $x_t$  is chosen to minimize the weighted sum objective function  $\sum_{t=1}^n (y_t - x_t)^2 + \lambda \sum_{t=2}^{n-1} (x_{t-1} - 2x_t + x_{t+1})^2$ , whereas in  $\ell_1$ -filtering the second term is replaced by  $\lambda \sum_{t=2}^n |x_{t-1} - x_t|$ , where  $y_t$  is the return series and  $\lambda \geq 0$  is the regularization parameter used to control the trade-off between smoothness of  $x_t$  and the size of the residual  $y_t - x_t$  (Kim, Koh, Boyd, & Gorinevsky, 2009).

estimates of a piecewise linear trend, generates kinks in slopes which subsequently can be interpreted as regime shifts, such as a sharp movement from an upward to downward heading price.

In finance,  $\ell_1$ -filtering has been particularly useful in creating momentum strategies. For instance, when applied to the S&P 500 index (Dao, 2014) it has been shown to outperform buy-and-hold, as well as a simple moving average and the  $\ell_2$ -filtering strategy, although not on a cost-adjusted basis. Similarly, a momentum strategy using  $\ell_1$ -filtering (Mitra & Rohit, 2020) generated significant risk-adjusted returns, particularly for emerging markets, even after including transaction costs in performance analysis.

However,  $\ell_1$ -filtering has also been shown to suffer from both technical and practical limitations. The technical issue relates to the regularisation hyperparameter  $\lambda$  that balances the  $L_1$  penalty function, as well as the loss function itself. The  $\lambda$  hyperparameter must be optimised, which in turn depends on the choice of a number of kinks (or trend changes) in the final result; the smaller the  $\lambda$ , the more kinks in the piecewise trend, and thus, the larger the frequency of regime shifts (Dao, 2014), (Mitra & Rohit, 2020). With regards to the penalty function, it cannot handle heavy-tailed errors, as well as outliers which are usually an essential part of financial time series (Wen, Gao, Song, Sun, & Tan, 2019).

The practical caveat stems from the inability of  $\ell_1$ -filtering to distinguish transitioning regimes. Similarly to the two-state Markov-switching model or multiple other traditional methods,  $\ell_1$ -filtering is only capable of finding two regimes (in this case, bullish or bearish) which has been shown to decrease asset allocation returns during periods which are neither bullish nor bearish (Mitra & Rohit, 2020).

### 2.3.2 Regime prediction methods

From a practical perspective, regime detection is only useful to perform studies on discovered regimes (e.g., investigate cross-asset correlations during market crashes

or bullish periods), with the possibility to then make assumptions about the future based on the results. However, there is a large gap between assuming and *de facto* predicting an event ahead, the topic which will now be considered. As with regime detection methods, techniques for predicting regimes can be separated into classic and machine learning techniques. These methods will be reviewed below, with a focus on tools to be used in the work of this thesis, Random Forest, which will be used as a main framework for regime predictions, and the Hidden Markov Model, which will serve as a main benchmark. Similarly to the previous section on regime detection methods, it is important to note here that the techniques below, apart from Hidden Markov Model and switching Kalman filtering, can be used to predict regimes but they are not regime-switching predictive techniques by definition.

### *Traditional regime prediction methods*

**Probit/Logit model.** Two frameworks stemming from the *Generalised Linear Model* (GLM) family, Probit and Logit, were the first multivariate attempts to predict recessions on a binary basis. The difference between them relates to the choice of link functions; the Logit model uses the cumulative distribution function of the logistic distribution, whereas the Probit model utilises the cumulative distribution function of the standard normal distribution<sup>16</sup>. The models also differ in terms of interpretation; however, ultimately, they will arrive at very similar (although not identical) results (Aldrich & Nelson, 1984).

Early works apply the Probit model to predict economic activity (Estrella & Hardouvelis, 1991) and later US recessions (Estrella & Mishkin, 1998) using NBER classification. The latter work provided evidence that US recessions are mainly driven by the slope of the yield curve, among other factors such as interest rates, equity prices, and monetary aggregates. This research was also extended to other countries, such as Germany, Canada, Japan, and United Kingdom, arriving at similar conclusions (Bernard & Gerlach, 1998). In line with the Probit model, the Logit

---

<sup>16</sup> The logit link function is defined as  $\text{logit}(p) = \ln\left(\frac{p}{p-1}\right)$ , whereas the probit link function, also known as the inverse normal link function, is defined as  $\text{probit}(p) = \Phi^{-1}(p)$ , where  $p \in (0, 1)$  and  $\Phi$  is the CDF of the standard normal distribution.

model was also applied to predict US (Birchenhall, Jessen, Osborn, & Simpson, 1999) and UK (Birchenhall, Osborn, & Sensier, 2001) recessions; interestingly, it has been shown that any attempts at predicting regimes (as opposed to detecting) using Markov-switching regression will be less successful than the Logit model, an issue already discussed in Section 2.2.3 (Birchenhall, Jessen, Osborn, & Simpson, 1999). However, more recent work comparing the Probit model with MSR shows that in terms of detecting NBER recessions, MSR performs better than the Probit model, although with a delay (Fossati, 2016).

The Probit model has had extensions in form of a dynamic specification (i.e., adding a lagged dependent variable), and an allowance for autocorrelated errors or multiple break points across business cycles (Chauvet & Potter, 2005), (Kauppi & Saikkonen, 2008). However, the rise of machine learning algorithms has put the Probit/Logit models aside, as even the simpler ML methods, such as ElasticNet and LASSO regressions (described in the next section), as well as *k-Nearest Neighbours* (KNN) and naïve Bayes, have been able to predict recessions significantly better in the short-, medium-, and long-term (Vrontos, Galakis, & Vrontos, 2021). This is also why currently the Probit/Logit models are used instead as naïve benchmarks in regime predictive tasks, a topic which will also be discussed in the section below on machine learning methods used for regime detection.

**(Switching) Kalman filter.** The *Kalman filter* (KF), invented by Rudolf E. Kalman in the 1960s (Kalman, 1960), was initially used to track a moving target from noisy measurements of its position and to predict its future position. Today, the KF is used in various areas, such as location and navigation systems, control systems, computer graphics, target tracking, as well as finance. An important extension to the KF, the *switching Kalman filter* (SKF) allows for the modelling of time-varying parameters or latent states in a dynamic system. This approach extends the traditional Kalman filter by incorporating the concept of regime shifts, where the underlying dynamics of the system can switch between different states or regimes over time (Murphy, 1998). The switching Kalman filter is particularly useful in capturing complex and nonlinear relationships that may exist in economic and financial time series data (Elliott & Osakwe, 2018).

The basic idea behind switching Kalman filtering is to model the transition between different regimes using a Markov chain, where each regime is associated with a specific set of state equations and observation equations. The filter then estimates the latent states and parameters of the system by recursively updating its beliefs about the current regime based on both past observations and prior information (Murphy, 1998). In terms of the latent states estimation methods, switching Kalman filtering can be approached from both Bayesian and frequentist perspectives. Bayesian methods, such as importance sampling (Billio & Monfort, 1998) or Monte Carlo Markov Chain (e.g., (Frühwirth-Schnatter, 2001), (Giordani & Kohn, 2008)), allow for the incorporation of prior beliefs and uncertainties into the estimation process, while frequentist methods focus on maximizing likelihood functions to estimate model parameters. Both approaches have their strengths and weaknesses, with Bayesian methods offering more flexibility in handling complex models and uncertainties, while frequentist methods may provide more straightforward interpretations and computational efficiency (Bauwens, Lubrano, & Richard, 2000).

One possible extension of the SKF is the forgetting (or resetting) switching Kalman filter which introduces a mechanism to gradually forget past observations as new data becomes available. This extension can be particularly useful in scenarios where historical data may become less relevant over time or when there are structural changes in the underlying dynamics of the system (Bracegirdle & Barber, 2011).

Applications of (switching) Kalman filtering in economics and finance are vast and diverse. In economics, (switching) Kalman filtering is commonly used for tracking macroeconomic variables, forecasting economic indicators, and estimating parameters in dynamic economic models (Pasricha, 2006). In finance, (switching) Kalman filtering is widely applied in areas such as asset pricing models, portfolio optimization, risk management, and high-frequency trading strategies (e.g., (Lautier, 2002), (Vo, 2014), (Liang, Thavaneswaran, Yu, Hoque, & Thulasiram, 2020)).

While the SKF offers significant advantages in capturing time-varying dynamics and regime shifts in complex systems, it also comes with several

disadvantages. One major drawback is the increased complexity and computational burden associated with estimating multiple regimes and switching probabilities. This can lead to challenges in model identification, parameter estimation, and interpretation of results (Billio, Casarin, & Sartore, 2007); however, the traditional Kalman filter takes centre stage in Chapter 5, providing a detailed explanation of this method and expanding the literature review to encompass additional financial applications.

**Other traditional models.** Alternative classic approaches to predicting regimes concentrate on forecasting GDP, which can be helpful in determining potential state shifts, such as moving from an economic boom to a recession (in case of two-quarter consecutive decline of GDP), and vice-versa. An early method of predicting GDP dating back to the 1960s was to simply follow Conference Board's *Composite Index of Leading Indicators* (CLI)<sup>17</sup> which in theory should give signs of imminent changes in a business cycle; however, this has usually been ineffective and required enhancements (Niemira & Klein, 1994). One such improvement was a first attempt to develop a formal statistical model that aimed at generating the probability of a recession with the use of CLI. The Neftçi model simply converted CLI into probability readings, which after crossing a certain threshold gave a signal of an upcoming economic downturn (Neftçi, 1982). The Neftçi model had higher accuracy in predicting regimes than the simple CLI following rule; however, the major flaw was that it merely gave a likelihood of a recession sometime in the future, as well as allowing only one independent variable at a time. The Probit model, in fact, does better than the Neftçi model in that it helps predict recessions at a particular forecast horizon with higher accuracy and on a multivariate basis (Filardo, 1999).

---

<sup>17</sup> The CLI is a weighted average of ten leading indicators: average weekly hours in manufacturing, average weekly initial claims for unemployment insurance, manufacturers' new orders for consumer goods and materials, vendor performance measured by slower deliveries diffusion index, manufacturers' new orders for nondefense capital goods, building permits for new private housing units, S&P 500 stock price, M2 money supply, 10-year Treasury bond yield less federal funds rate, and index of consumer expectations (Filardo, 1999).

Apart from the binary models discussed on the previous page, the literature mentions other regression attempts to predict GDP. One promising approach used by monetary and fiscal authorities is the *Dynamic Stochastic General Equilibrium* (DSGE) model, which has also been implemented to predict inflation (Del Negro & Schorfheide, 2013). DGSE was initially a descriptive tool that allowed economists to think about business cycles and carry out hypothetical policy experiments; however, it was very limited in terms of forecasting due to being too minimalistic. This perception changed after the introduction of a DGSE model which adopted Bayesian techniques for estimation. The new DGSE framework managed to outperform its direct competitors such as the Random Walk, VAR<sup>18</sup>, and Bayesian VAR models with regards to predicting US and Euro area GDPs, an achievement that gained significant attention from central bankers around the world (Smets & Wouters, 2007), (Christoffel, Coenen, & Warne, 2010).

The DGSE model has been further extended to improve GDP forecasting performance; this hybrid model was a blend of DGSE with Factor Augmented VAR model and proved to outperform all the DGSE alternatives (Consolo, Favero, & Paccagnini, 2009). However, both the standard DSGE model and the hybrid model are still linear and do not incorporate time-varying parameters that could account for inherent non-linearities and capture the adaptive underlying structure of the economy in a robust manner (Bekiros & Paccagnini, 2016), unlike, for instance, Markov-switching regression once applied to regime detection tasks (see Sections 2.2.3 and 2.3.1).

The described traditional models used in regime prediction have one thing in common: they all predict recessions from either a NBER or GDP point of view. This is in fact problematic from a practical perspective, for the following reasons: 1) as shown in Figure 20, economic recessions are rare and markets can go through multiple smaller corrections between them, exposing portfolios to unwanted losses; 2) by focusing solely on recessions, the NBER or GDP labelling method categorises every other period as a single “bullish” regime. However, financial markets differ from a state economy in that they are not simply separated into expansion and

---

<sup>18</sup> VAR stands for Vector AutoRegressive, is a stochastic process model used to capture the relationship between multiple quantities as they change over time. For more details on the VAR model and its application to economics, refer to (Bagliano & Favero, 1998).

contraction cycles but also include (sometimes prolonged) transitory regimes, which were discussed in Section 2.2.1 and will be addressed in Chapter 3. This is why applying a classic regime detection technique that is able to capture all relevant states in the underlying asset price data to generate labels is a better solution for predictive analytics involving financial markets (Mullainathan & Spiess, 2017), (Uysal & Mulvey, 2021). The choice of algorithm to predict these labels is a separate point of discussion that will be taken up in the next section; however, as will be shown below (and was mentioned in the Probit model discussion), machine learning algorithms tend to be superior to traditional methods in terms of predicting financial regimes ahead of time.

### *Machine learning regime prediction methods*

As discussed in the section above, machine learning algorithms have significantly advanced research on economic regime prediction and have challenged the classic approaches to this problem (Athey, 2019), (Vrontos, Galakis, & Vrontos, 2021). The literature mentions many algorithms that can generate accurate predictions in terms of regime shifts, from the already described in Section 2.2.4 Hidden Markov Model and Random Forest to Support Vector Machines (SVMs) and artificial neural networks. The current section will focus on these latter two methods in particular, due to their proven effectiveness and later relevance to the work of this thesis; however, it is noted that there are other techniques available which could be of interest to a reader.

**Hidden Markov Model.** As pointed in Section 2.2.4, this thesis categorises the Hidden Markov Model as a machine learning framework due to its similar classification by other academics, as well as its parameter estimation mechanism (Baum-Welch algorithm) which closely resembles the training processes of ML algorithms.

Since its introduction in the 1960s (Baum & Petrie, 1966), the HMM has been successfully applied in the areas of the telerobotics (e.g., speech recognition, mobile robot motion (Rabiner, 1989), (Yang, Xu, & Chen, 1994)), biology (e.g., DNA

sequencing, protein structure prediction, ECG analysis (Eddy, 1996), (Koski, 1996)), customer relationships (Netzer, Lattin, & Srinivasan, 2008), and recognition of fraudulent credit card activities (Mhatre, Almeida, Mhatre, & Joshi, 2008), to name a few. Similarly, HMMs have been successfully utilised in financial markets, both as an asset price forecasting tool and as a technique for regime prediction. In terms of predicting asset prices, the HMM was some time ago shown as a promising method for forecasting daily probability distributions in equity returns (Weigend & Shi, 2000). Later, the HMM was used to predict airline stock returns and demonstrated a similar out-of-sample accuracy to artificial neural networks (Hassan & Nath, 2005). More recently, a multivariate HMM which used technical indicators created from opening, high, low, and closing prices has been applied to predict major US stocks returns; however, even though the out-of-sample accuracy was on average over 50%, the model was outperformed by the Support Vector Machine algorithm that will be outlined later below (Rao & Hong, 2010). Similarly, an alternative multivariate HMM framework with maximum a posteriori parameter estimation<sup>19</sup> was applied to stock returns for predictive tasks, but with rather mixed results, as the model did not outperform its competitors (e.g., ANN) in every example (Gupta & Dhingra, 2012). Finally, different-states HMMs were applied to predict monthly S&P 500 returns with the later purpose of trading the index (Nguyen, 2018); this research found the four-state model to be the most accurate framework relative to its competitors.

With regards to state shifts, a univariate HMM with two states was used to predict market turbulence (proxied by equity stock returns), inflation (US CPI), economic growth (real GDP; real industrial output), and market volatility (VIX Index) (Kritzman, Page, & Turkington, 2012), (Nguyen & Nguyen, 2015); however, the prediction results of these works were shown only in-sample, as the out-of-sample testing involved only asset allocation based on these predictions in a further and separate step. An HMM was also applied to predict the probability of a bear equity market using variables such as inflation, Dow Jones Industrial Average (DJIA) returns, corporate bonds, yield curve, and commodity data; the results showed that DJIA and inflation rate are the most useful in achieving this task. However, similarly

---

<sup>19</sup> Refer to (Gauvain & Lee, 1994) for technical details on maximum a posteriori estimation in HMM.

to previously described works, these predictions were carried out in-sample to be subsequently used in trading strategies (Nguyen, 2014), rather than being predictions of the future per se. Finally, in addition to equities, the HMM has been applied to commodities, foreign exchange, and cryptocurrencies; the model was able to infer future price trends (downward and upward) in the oil market (Legey, Silva, & Silva, 2010), Euro / US Dollar and Australian Dollar / US Dollar crosses (Lee, Ow, & Ling, 2014), as well as Bitcoin (Giudici & Hashish, 2020).

Even so, the HMM has its limitations, which were discussed in Section 2.2.4, as well as tending to underperform or perform similarly to other machine learning techniques, such as Support Vector Machines or artificial neural networks, in predictive tasks. Thesis will use the HMM only as a benchmark in Chapter 4, this choice being due to the widespread application of HMM in finance, and its flexibility with regards to setting the number of forecasted regimes, i.e., switching from a binary (two-state) to multiclass (three- or four-state) classification without increasing complexity.

**Random Forest.** The origins of the *Random Forest* (RF) model can be traced back to the Classification and Regression Tree (CART) algorithm (Breiman, Friedman, Olshen, & Stone, 1984) (also see Section 2.2.4). Though simple to apply and to interpret, CART had a high sensitivity to small perturbations in the data, causing the algorithm to overfit. As already described in Section 2.2.4, this limitation was later overcome in the form of the Random Forest model by introducing randomness into the decision tree algorithm and then aggregating the results of many decision trees, which proved to be more robust to overfitting than a single CART model (Breiman, 2001).

Random Forest has been one of the most popular machine learning ensemble algorithms to date, due to the vast success in its application to various datasets. In a recent study (Fernández-Delgado, Cernadas, Barro, & Amorim, 2014) RF has been shown, on average, to be the most accurate classification algorithm out of 17 different methods (e.g., including SVMs, artificial neural networks, linear models, boosted trees, and simple decision trees) when used on 121 different datasets, both

real and synthetic. Similar results were obtained when Random Forest and 14 other methods were applied to 115 real-life datasets (binary and multiclass) (Wainer, 2016), although this time the research included hyperparameter optimisation, with no significant difference in performance between the top three algorithms: RF, SVM, and gradient boosted trees.

Random Forest has also been successfully applied in finance and economics, particularly in problems of stock price and regime prediction. In terms of general price direction, RF was applied to over 5500 European stocks and subsequently compared with other algorithms (e.g., SVM, k-Nearest Neighbours, neural networks, and regularised Logistic Regression) (Ballings, Michel, Hespeels, & Gryp, 2015); its accuracy in predicting one-year ahead price direction with the use of fundamental and macro variables was shown to be the highest. In addition, compared to SVM and naïve Bayes (Milosevic, 2016), Random Forest achieved the highest F-Score in stock price correction (at least 10% upward or downward movement) prediction over the long term. Finally, good results have been obtained when RF was trained with technical indicators to predict price trend of several equities (Kumar, Dogra, Utreja, & Yadav, 2018), though its performance was slightly worse than that of Naïve Bayes when applied to a small dataset.

With regards to predicting regimes, Random Forest been shown to be a significant improvement in predicting bank crises in 17 countries compared to traditional approaches such as the Logit and Probit models (Ward, 2017). RF has also outperformed its ML competitors, such as gradient boosted trees, naïve Bayes, regularised linear models (described separately later), and the closely related *Extremely Randomised Trees* (ExtraTrees)<sup>20</sup> (Piger, 2020), when used as a predictor of regime turning points. In addition, Random Forest has achieved a near perfect accuracy in predicting US recessions, particularly more recent distressing periods, such as Covid-19 or US-China trade wars (Yazdani, 2020), compared to an SVM, single-layer neural network, eXtreme Gradient Boosted Trees model (discussed on the next page), the Probit model, or a regularised linear model. Finally, RF has also been used as a predictor for binary stock market regimes previously generated by

---

<sup>20</sup> Extremely Randomised Trees is a similar algorithm to Random Forest; however, ExtraTrees is built with decision trees that are more variable and less correlated than the trees in RF. For technical details and comparison, refer to (Geurts, Ernst, & Wehenkel, 2006).

the  $\ell_1$ -filtering method (Uysal & Mulvey, 2021); the signals predicted by Random Forest were subsequently used in asset allocation, a topic that will be discussed in the next section.

While Random Forest has clearly had significant success, the works described above do not test the algorithm's performance when applied to multiple states, such as bullish, bearish, and transitory, with the only discovered work treating multiple states being one with an engineering application (Lin, Fang, Xiaolong, & Shisheng, 2017), in a stacked procedure involving Random Forest, Extreme Learning Machines (ELMs, outlined later below), and Markov-switching regression. These results suggest the promise of a multi-state approach in terms of predicting multiple financial regimes.

Thus, based on the described performance in this section, this thesis will utilise Random Forest in regime prediction tasks, in a way which will be outlined in Chapter 4. However, as briefly stated in the introduction to this section, a number of other machine learning prediction methods will now be discussed, as they have seen substantial use for regime prediction, but their limitations for the purposes of this thesis will be explained, justifying the final choice of Random Forest.

**Gradient Boosted Trees.** Gradient boosting algorithms are ensemble machine learning method that has been successfully applied to multiple datasets. In contrast to a bagging algorithm such as Random Forest (see Section 2.2.4 for technical details on a bagging algorithm), gradient boosters apply a classification algorithm sequentially, re-weighting misclassified instances on each iteration, instead of averaging them. The classification method could be a simple decision tree or an extension, such as CART, which in combination with AdaBoost (Freund & Schapire, 1997), an early example of a boosting technique, showed a lower test error relative to a bagging algorithm (Friedman, Hastie, & Tibshirani, 2000).

*Gradient Boosted Trees* (GBT) have been widely applied in finance for predictive tasks; however, as was discussed earlier, GBTs generally tend to underperform Random Forest with regards to predicting stock price trend or economic recessions, though they can significantly outperform traditional regime

forecasting methods, such as the already described Probit model (Döpke, Fritzsche, & Pierdzioch, 2017). The suboptimal performance of GBTs in these areas has been suggested to be mainly dictated by an inability to capture all NBER recessions, generating “false alarms”, and producing delayed true positives (Ng S. , 2014), (Berge, 2015).

Recently, two powerful gradient boosting frameworks have been introduced: *eXtreme Gradient Boosting* (XGB) (Chen & Guestrin, 2016), and LightGBM (Ke, et al., 2017), the latter constructed to significantly improve training speed relative to XGB, particularly for large datasets. XGB, due to its overall speed, scalability, and accuracy has been widely used in various Kaggle competitions<sup>21</sup>, as well as having been successfully applied to predicting the direction of stock market prices (Dey, Kumar, Saha, & Basak, 2016), failure in the US banking sector (Carmona, Climent, & Momparler, 2019), and US recessions (Yazdani, 2020), though in the latter case shown to underperform Random Forest, as was also discussed above, thus providing evidence to support the choice of RF in this thesis.

**Regularised linear models.** As mentioned in the previous section on traditional predictive methods, linear models, such as logit and probit, even though simplistic, can be helpful in predicting regime shifts; however, incorporating regularisation techniques into these linear models can significantly enhance their predictive capabilities. Popular regularisation methods like L1 (Lasso), L2 (Ridge), ElasticNet, and Huber loss offer a systematic approach to address issues of overfitting and model complexity, which are common challenges faced when dealing with regime shifts in economic and financial data (Alessi & Savona, 2021).

L1 regularisation imposes a penalty on the absolute size of the coefficients, encouraging sparsity in the model by shrinking some coefficients to zero. This can help in feature selection and interpretability (Tibshirani, 1996). On the other hand, L2 regularisation penalises the squared magnitude of coefficients, leading to more stable and robust models by reducing multicollinearity (Ng A. Y., 2004). ElasticNet

---

<sup>21</sup> Refer to <https://www.kaggle.com/docs/competitions> for details on Kaggle and its competitions. Accessed on 04/03/2022.

goes one step further and combines both L1 and L2 penalties, offering a balance between variable selection and model stability. It is particularly useful when dealing with high-dimensional data where there are many correlated features (Zou & Hastie, 2005). Finally, utilising Huber loss as a regularisation technique adds robustness to outliers and balances between the mean absolute error and mean squared error commonly used as accuracy metrics (Huber, 1964).

Apart from simple linear regressions, another group of models that utilise regularisation methods is piecewise constant and linear solutions which involve representing relationships between variables with distinct constant or linear segments, respectively. Piecewise constant solutions break the data into segments where the relationship remains constant within each segment but can change abruptly at segment boundaries, offering flexibility for non-smooth patterns. In contrast, piecewise linear solutions approximate the relationship with linear segments, allowing for capturing complex data patterns by fitting multiple linear models to different segments. By integrating regularisation techniques such as L1 or L2 into piecewise constant and linear solutions, they can benefit from enhanced interpretability and flexibility in handling complex data structures (Rosset & Zhu, 2007), (Kato, 2010).

In economics and finance, regularised linear models have been successfully applied to various tasks, including asset pricing, risk assessment, as well as forecasting economic trends, including regime shifts (e.g., (Chan-Lau, 2017), (Alessi & Savona, 2021), (Hoang & Wiegertz, 2023)). While these solutions provide superior interpretability and explainability compared to tree-based algorithms, regularised linear models face challenges in capturing complex non-linear relationships within the data. In addition, they often underperform more advanced techniques like Random Forest in tasks related to predicting regime shifts, as previously discussed in the Random Forest subsection.

**Support Vector Machines.** Originally developed in the 1960s, SVMs gained larger attention decades later after introducing a “kernel trick” which helps in application of SVM to datasets that are not linearly separable (Boser, Guyon, &

Vapnik, 1992). Similarly to Random Forest and gradient boosting algorithms, Support Vector Machines have been applied to multiple datasets with success, though some experiments involving multi-topic databases (not all financial), as already discussed in the Random Forest paragraph on the previous page, found no significant difference between performances of these three methods (Wainer, 2016).

Support Vector Machines have been widely applied to financial datasets. For instance, an early work (Kim K.-j. , 2003) demonstrated SVMs to be a promising method of predicting future stock price direction, due to the algorithm's capability to outperform shallow artificial neural networks in this task. In addition, an optimised SVM using various fundamental and technical indicators has been shown to avoid overfitting in predicting future trend of emerging market stocks (Lin, Guo, & Hu, 2013). However, as was mentioned earlier, SVMs generally underperforms RF when used as a method to predict future stock price trends (Ballings, Michel, Hespeels, & Gryp, 2015), (Kumar, Dogra, Utreja, & Yadav, 2018).

In terms of regime shifts, SVMs have been proved to be useful in forecasting recessions, analogously to Gradient Boosted Trees and Random Forest algorithms; however, similarly to GBTs, SVMs have issues with generating "false alarms" (Gogas, Papadimitriou, Matthaio, & Chrysanthidou, 2015), as well as underperforming the Probit model in the short term (Plakandaras, Cunado, Gupta, & Wohar, 2017). Even so, there have been successful attempts to improve on these limitations, such as blending SVMs with a dimensionality reduction technique called 'neighbour rough set' (Wang, Xiao, Zhao, Ni, & Li, 2019). However, SVMs are still an inferior technique relative to Random Forest with regards to predicting regimes (Yazdani, 2020), a topic already discussed earlier. In addition, SVMs do not support multiclass classification natively, as the algorithm was originally designed for a binary classification. Though there are ways to tackle this issue, such as breaking a multiclass problem into several binary classifications in a "one-versus-one" manner<sup>22</sup>, they may adversely impact the model's complexity and extend training

---

<sup>22</sup> This is the standard approach when implementing SVM using a previously mentioned package scikit-learn. Refer to <https://scikit-learn.org/stable/modules/svm.html> for technical details. Accessed on 04/03/2022.

time without significant improvement in accuracy, particularly when working with large datasets (Wang, Zhang, & Wu, 2019).

**Artificial neural networks.** ANNs are computing systems inspired by the way the human brain operates. The ANN framework is comprised of connected units or nodes called 'artificial neurons' (or, frequently in this context, just 'neurons') which form layers that taken together can approximate any function to some degree of accuracy. One of the simplest and most common ANN architecture is referred to as a feed-forward architecture, wherein connections between neurons always go forward, beginning from the input layer and ending on the output layer. Artificial neural networks had already been known in the 1940s, although critical milestones were achieved decades later after introducing the perceptron model in 1958 (Rosenblatt, 1958), backpropagation for the calculation of errors in 1974 (Werbos, 1990). In addition, since the 1980s parallel computing has significantly sped up an ANN's training time.

In finance and economics, the most popular topics of application of ANN are portfolio optimisation (as will be discussed in the next section), as well as the prediction of asset prices and market recessions. Relating to the latter topic, academic works from as early as the mid-1990s tried to forecast turning points in economic cycles using feed-forward ANNs, although initially just in-sample (Vishwakarma, 1994). Later the research was extended to out-of-sample performance and generated very clear signals of recessions and expansions (Qi, 2001). More recently, ANNs have also been compared with Markov-switching models in relation to the prediction of regime switches. When both models were applied to an equity index (Liu & Zhang, 2010), ANNs were shown to be a preferable method to MSR, in line with expectations based on works mentioned in previous paragraphs (e.g., while describing the Probit method). However, MSR can serve as a useful component in a modelling framework that also involves artificial neural networks. For instance, applying the Markov-switching model to determine states of the input features and subsequently feeding the MSR output to an ANN

architecture called a Fuzzy-Neural System<sup>23</sup> has been shown to predict stock market volatility with a high accuracy relative to a standard feed-forward ANN (Kristjanpoller & Michell, 2018).

Another type of ANN recently applied to finance with some success is the Extreme Learning Machine (ELM), which was first introduced to significantly improve the efficiency and speed of a single-layer feed-forward ANN. An ELM does not require hidden neurons to be tuned; it randomly assigns hidden nodes, constructs biases and input weights of hidden layers, and determines the output weights using least squares methods (Huang, Zhu, & Siew, 2006). In finance, the ELM has been mostly applied as a price prediction tool, whether to equities (Li, et al., 2016) or to crude oil (Wang, Athanasopoulos, Hyndman, & Wang, 2016), and in these works found to outperform standard feed-forward ANNs or other machine learning techniques, such as Support Vector Machine, in this task. However, to the best of the author's knowledge, there have been no attempts to apply an ELM to predict financial regimes.

Apart from feed-forward nets, ELMs, and fuzzy nets, the academic literature contains several other kinds of ANN architectures with application to finance, with the recurrent type the most promising for time series. In particular, the *Long Short-Term Memory* (LSTM) framework<sup>24</sup> has been shown to outperform other recurrent neural network (RNN) architectures, such as the *Gated Recurrent Unit* (GRU)<sup>25</sup> model, and a plain RNN when applied to forecasting financial assets (Di Persio & Honchar, 2017) and equity prices (Roondiwala, Patel, & Varma, 2017). The performance of recurrent neural networks in terms of predicting stock prices can be increased by using a LSTM-GRU architecture; this involves constructing the LSTM layer first and then passing its output to a GRU layer to generate the final predictions (Hossain, Karim, Thulasiram, Bruce, & Wang, 2018). Finally, in terms of predicting regimes instead of stock prices, a recurrent neural network architecture built using a bi-directional Long-Short Term Memory algorithm with an autoencoder has been

---

<sup>23</sup> For a granular review of fuzzy neural networks, refer to (Musilek & Gupta, 2000).

<sup>24</sup> Refer to (Hochreiter & Schmidhuber, 1997) for technical details on recurrent neural networks, particularly LSTM.

<sup>25</sup> Refer to (Chung, Gulcehre, Cho, & Bengio, 2014) for an empirical study of the GRU.

shown to accurately predict the beginning and end of economic recessions (Wang, Li, Xia, & Liu, 2021).

However, as mentioned earlier, in general artificial neural networks are subpar relative to Random Forest when used for stock prediction (Ballings, Michel, Hespeels, & Gryp, 2015) and recession forecasting (Yazdani, 2020) tasks. The literature does not contain any strong evidence to support their use for the regime prediction task of Chapter 4. Thus, this thesis will not pursue ANN as an algorithm for predicting regimes further; however, being a possible framework for portfolio allocation, the topic of Chapter 5, artificial neural networks will be reviewed further in Section 2.3.3.

**Other machine learning methods.** Other machine learning techniques used in regime prediction mentioned in the literature can be separated into non-complex frameworks, such as k-Nearest Neighbours and naïve Bayes, and more advanced methods, such as reinforcement learning (RL) and Learning Vector Quantisation (LVQ). These techniques will not be discussed at length here, for reasons which will be given below; however, due to their demonstrating some limited success in predicting financial regimes, they are briefly discussed.

With regards to the simpler frameworks, in general their performance in stock price or regime prediction task has been found to be inferior to the other machine learning methods described earlier in this section (particularly Random Forest), although with some exceptions. The KNN algorithm, which assumes that predictions can be generated using datapoints in proximity<sup>26</sup>, showed almost similar performance to Random Forest and even slightly outperformed it over a long term when used in a recession prediction task (Vrontos, Galakis, & Vrontos, 2021); however, it still falls short in stock price (Ballings, Michel, Hespeels, & Gryp, 2015) and market trend prediction tasks (Kumar, Dogra, Utreja, & Yadav, 2018), as mentioned in the section dealing with Random Forest. In addition, KNN can train very slowly on large datasets, as well as being very sensitive to redundant and

---

<sup>26</sup> For technical specifications on k-Nearest Neighbours, refer to (Altman, 1992).

irrelevant features, an issue that is easily tackled by Random Forest (see Section 2.2.4) (Imandoust & Bolandraftar, 2013).

The second simpler algorithm, naïve Bayes<sup>27</sup>, has also been compared to other machine learning methods in regime prediction tasks. Even though its performance is almost always subpar relative to Random Forest (except for application to small datasets), as was described in the Random Forest section, a recent study showed its significant outperformance relative to other simple frameworks, such as regularised logistic regression and the Probit model (Davig & Hall, 2019). Due to its simplicity, high training speed, and interpretability, naïve Bayes remains a popular method used in various classification tasks; its accuracy can *de facto* be increased by coupling with kernel density estimation<sup>28</sup> which is used to estimate conditional marginal densities of each class (Piryonesi & El-Diraby, 2020).

As mentioned above, the other group of algorithms that can be useful tools in predict regimes or forecast stock prices involve more complex solutions, such as LVQ and reinforcement learning. LVQ is a classifier somewhat similar to KNN, in that it forms predictions on the basis of proximity of the dependent features; however, the key difference is that in the case of LVQ these features are not collections of training sample data, but instead are endogenously learned from the training sample data (Kohonen, 2001). LVQ has been recently used to predict NBER recessions one-step-ahead and has shown promise in this task, particularly when considering its high run-time performance (Giusto & Piger, 2017).

Finally, reinforcement learning has been tried as an approach to predict asset prices, as well as regime switches. Reinforcement learning is not an algorithm *per se*, rather a subcategory of machine learning which does not require labels for training purposes. Instead, RL is a type of dynamic programming that trains algorithms using a system of rewards and penalties. The agent, which serves as a learning system, explores the unknown territory, and exploits its gained knowledge to receive rewards for performing correctly or to receive penalties for performing

---

<sup>27</sup> Refer to (Rish, 2001) for an empirical study of the naïve Bayes classifier.

<sup>28</sup> Kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable. In case of naïve Bayes, functions as Gaussian or Epanechnikov can be used for local data smoothing when the algorithm is coupled with KDE (Piryonesi & El-Diraby, 2020).

incorrectly (Sutton & Barto, 1998). Financial literature on RL has been overwhelmingly dedicated to portfolio optimisation and asset trading, which will be discussed in the next section. In terms of forecasting prices, the RL framework has been successfully applied to predict the Korean stock market with the use of technical indicators as inputs (Lee J. W., 2001); the research showed this approach to be a useful tool to generate trading signals. In addition, a specific RL architecture called the Actor-Critic method<sup>29</sup> has outperformed a recurrent neural network when applied to the US markets over a short period (Li, Dagli, & Enke, 2007). Finally, RL has also been tried to handle the regime-switching nature of financial time series. A novel framework called *regime-switching recurrent reinforcement learning*<sup>30</sup> has been shown to be a promising way to predict regime shifts in time series and use them in investment decision-making (Maringer & Ramtohul, 2012). However, due to the high complexity of RL frameworks, low training speed, and the substantial datasets required to train the agent, this interesting technique will not be explored further in this thesis.

### 2.3.3 Portfolio allocation methods related to regime switching

This section focuses on various methods of portfolio allocation (also referred to as *asset allocation* and *portfolio optimisation*). The classic methods and their limitations were already discussed in Section 2.1.2; this section focuses on portfolio allocation techniques which are associated with the classic regime detection and some prediction methods discussed in Sections 2.3.1 and 2.3.2. Subsequently, limitations of these techniques are given, and potential methods offered to solve them.

---

<sup>29</sup> Refer to (Grondman, Busoniu, Lopes, & Babuska, 2012) for technical details on the Actor-Critic method.

<sup>30</sup> Recurrent reinforcement learning uses recurrent neural networks inside its framework. The regime-switching part has been obtained from blending the entire architecture with the TAR model, mentioned in Section 2.3.1 under the Other regime detection models paragraph.

### *Traditional portfolio allocation methods associated with regime switches*

Regime-switching portfolio allocation relies on *ex ante* signals generated using various regime detection or prediction methods. Even though prediction techniques are more valuable from a portfolio manager's point of view (as was also discussed in Section 2.3 of related work and in Section 2.2.3 on the limitations of MSR), plain detection methods (i.e., with no attempt to predict a regime switch ahead of time) have also been applied to construct vanilla portfolios. The application to portfolio optimisation of the most important traditional concepts, namely the GARCH method, the Markov-switching model, and the Hidden Markov Model, will now be described, as well as, briefly, their limitations.

**Portfolio allocation using GARCH.** As was discussed in Section 2.3.1, the standard GARCH model has multiple limitations which have spurred the development of its extensions. One of these, the Markov-switching GARCH (MS-GARCH) model, has shown the most promise with regards to its application to asset allocation. For example, the MS-GARCH model has been applied to US, Asian, and Balkan equities to reduce variance and increase portfolio utility (Lee & Yoder, Optimal hedging with a regime-switching time-varying correlation GARCH model, 2007), (Sheu, Lee, & Lai, 2013), (Sheu & Lee, 2014), (Škrinjarić & Šego, 2016). Similar results have been obtained for commodity portfolios, particularly energy commodities, whose hedging by the use of the MS-GARCH model resulted in a reduction of portfolio risk (Lee & Yoder, 2007), (Alizadeh, Nomikos, & Pouliasis, 2008). Finally, a novel extension of the MS-GARCH model called Jump-GARCH has made it possible to improve the decision-making process for an average trader by generating buy and sell signals in precious metals (Carpinteyro, Venegas-Martínez, & Aali-Bujari, 2021).

Even though the MS-GARCH model has found success in asset allocation, it appears it is the Markov-switching component that is responsible for these positive results. In fact, asset allocation using the Markov-switching regression model alone has been shown to outperform standard GARCH models when applied to US equities (Alizadeh & Nomikos, 2004). Further evidence has been provided by applying the

extended Hidden Markov Model to European equities (Salhi, Deaconua, Lejaya, Champagnat, & Navet, 2016). Thus, it would currently be fair to assume that regime-switching allocation using Markovian methods alone is more beneficial than using their MS-GARCH equivalent. Examples of building portfolios by the use of Markovian methods, namely Markov-switching Regression (MSR) and Hidden Markov Methods (HMM), are given below.

### **Portfolio allocation using the Markov-switching regression model (MSR).**

One of the earliest pieces of research on adapting Markov-switching regression models to asset allocation was carried out in 2002, and later extended in 2004 for the international equity markets. In the first work (Ang & Bekaert, 2002), MSR was used to determine that highly volatile regimes are linked to higher return correlation among equities, which emphasised the higher need for diversification during bearish periods. In the second work (Ang & Bekaert, 2004), it was found that regime-switching strategies established with the use of MSR generated higher out-of-sample Sharpe ratios than classic mean-variance optimisation (outlined in Section 2.1.1). Similar studies were performed in 2003 on German, Japanese, US, and UK markets (Graflund & Nilsson, 2003), where it was concluded that ignoring regimes in the long term is associated with significant losses, as optimal portfolio weights for dynamic portfolio rebalancing are clearly dependent on a correctly detected state. Later, in 2005, two-state and three-state MSRs were applied to UK equities and bonds, respectively (Guidolin & Timmermann, 2005), to demonstrate persistence of different regimes within both classes, and subsequently suggest optimal portfolio allocations under these changing conditions. This study was later extended to the application of four-state MSR (Guidolin & Timmermann, 2007), which was set up to capture the joint distribution of stock and bond returns and be used to optimise portfolio weights between equities and bonds. Finally, by using a regime-switching strategy that combined MSR with the Capital Asset Pricing Model<sup>31</sup>, it was shown that regime switches are a valuable timing signal for portfolio rebalancing (Hess, 2006).

---

<sup>31</sup>  $E(R_i) = R_f + \beta_i(E(R_m) - R_f)$ , where  $E(R_i)$  is a capital asset expected return,  $R_f$  is a risk-free return,  $\beta_i$  is a beta of the investment (sensitivity), and  $(E(R_m) - R_f)$  is market risk premium.

Even though some of the above studies presented out-of-sample results, these techniques have not necessarily been practical in the industry. Frequent portfolio rebalancing, due to the number of state changes (Graflund & Nilsson, 2003), (Hess, 2006), or the complete exclusion of transaction costs in academic studies (Ang & Bekaert, 2004), (Guidolin & Timmermann, 2005), (Guidolin & Timmermann, 2007) tends to significantly lessen the potential for real-world regime-switching portfolio performance improvements (Bulla, Mergner, Bulla, Sesboüé, & Chesneau, 2011). In addition, the above-described works did not attempt to predict regime switches ahead of time, which is a critical obstacle from the practical perspective (see Section 2.2.3 on MSR limitations); however, the detection vs. prediction issue can be partially tackled using Hidden Markov Model, as was pointed out in Section 2.2.4 on machine learning models for regime prediction. Relevant applications of HMM to regime-switching portfolio allocation are outlined below.

**Portfolio allocation using the Hidden Markov Model (HMM).** Similarly to MSR, by the use of HMM it is possible to determine asset allocation in portfolios affected by regime switches. Indeed, there is evidence that HMM can select assets that help avoid risk during left-tail events (Bae, Kim, & Mulvey, 2014), as well as dynamically choose between different asset classes (equities, bonds, and alternatives) outperforming on a risk-adjusted basis Markowitz, naïve (1/N), and 60%/40% (S&P 500 and US Treasuries) portfolios (Kim, Jeong, & Lee, 2019). In addition, applying HMM to currency futures improves carry trade<sup>32</sup> performance by avoiding the significant losses that carry trade strategies usually sustain during financial or currency crashes (Reus & Mulvey, 2016). Buying and selling S&P 500 ETFs after identifying market regimes with HMM has also been found to yield better performance comparing to factor models widely known in the industry and academia, such as Fama-French Three-Factor Model, Carhart Four-Factor Model, and AQR Factor Model (Wang, Lin, & Mikhelson, 2020). Finally, the Hidden Markov Model has also been applied to macroeconomic variables, such as inflation,

---

<sup>32</sup> A carry trade is a trading strategy that involves borrowing at a low interest rate and investing in an asset that provides a higher rate of return. A carry trade is typically based on borrowing in a low interest rate currency and converting the borrowed amount into another currency.

industrial production, and VIX, which helped allocating funds to stocks that, on average, have been well rewarded during adverse regimes, such as high inflation, low industrial production, or high volatility (Nguyen & Nguyen, 2015).

Analogously to the Markov-switching regression model, in research discussed in the previous sub-section, the HMM has been applied to various assets to generate signals for subsequent portfolio allocation. Indeed, the earlier-described works used HMM to predict regimes, and subsequently allocated higher weights to assets that, for instance, had been particularly resistant to left-tail events or tended to generate significant gains during market rallies (Bae, Kim, & Mulvey, 2014), (Nguyen & Nguyen, 2015), (Reus & Mulvey, 2016), (Kim, Jeong, & Lee, 2019). However, some research (Bae, Kim, & Mulvey, 2014), (Nguyen & Nguyen, 2015), (Kim, Jeong, & Lee, 2019) excluded transaction costs in calculating portfolio risk-adjusted returns, which is worrisome in terms of applying the HMM framework in the real-world environment. In addition, the HMM, as was indicated in Section 2.2.4, suffers from the inability to predict regimes ahead of time, without experiencing the onset of the new regime. Thus, even though the HMM has been a valuable tool in constructing regime-switching portfolios, it still suffers from serious limitations which make it, similarly to MSR, a suboptimal solution for a portfolio manager in the real-world environment. The works described above indeed suffered from this lack of ability to predict, although attempted to tackle it by overlaying the inferred signals with rule-based strategies (e.g., Markowitz mean-variance, semi variance optimisation) and cross-validation; such an approach effectively trains the portfolio weights and applies them going forward until the signal switches. However, this is only a partial solution to the problem, since the HMM or MSR (or other method, such as GARCH) will continue to lag the reality, and thus the signal of a regime switch will be generated too late for its optimal use.

### *Regime-based optimisation dilemma*

As was stressed several times, the above methodologies provide only a signal. Beyond this, and of vital importance to practitioners, is the question of how this signal should be used. This major issue has suggested two paths a practitioner can

follow, in the case of optimising portfolios using regime-based frameworks (HMM, MSR).

The first solution is an attempt to generate signals by the use of lagging models, such as HMM and MSR, and to apply optimal control solutions for asset weight optimisation. This approach can include model constraints, such as transaction costs and signal delay, and effectively predict how to dynamically allocate weights for the future based on inferred probabilities. One work which followed this route was (Boyd, et al., 2017), who implemented multi-period optimisation (MPO) with model predictive control (MPC) for trading, which was also used, on the top of *ex ante* generated signals from an HMM, by (Nystrup, Madsen, & Lindström, 2018). MPC dynamically optimised portfolio allocation based on inferred regime probabilities, and included selected constraints, such as transaction costs and risk aversion. The proposed framework not only could outperform single-period optimisation techniques on a risk-adjusted basis, such as the ones described in Section 2.1.2, but also benchmark portfolios, such as MSCI World Index, rule-based asset allocation based on HMM predictions, and a classic 1/N portfolio; however, the work involved allocation between an equity index and zero-interest cash only. A further attempt to apply the HMM-MPC framework to a multi-asset environment (Nystrup, Boyd, Madsen, & Lindström, 2019) has showed promise in terms of beating the naïve 1/N portfolio on a risk-adjusted basis in long-only and long-short portfolios; however, both portfolios resulted in Sharpe ratios slightly below 1 (which is undesirable in the industry), even though the authors assumed low transaction costs, which could have further decreased the Sharpe ratio. Other attempts to apply multi-period optimisation with model predictive control to the HMM output (e.g., (Oprisor & Kwon, 2020), (Li, Uysal, & Mulvey, 2022)) arrived at alike conclusions in terms of the superiority of the multi-period optimisation over the single-period optimisation technique.

The non-optimal results of the MPC papers above may stem from blending lagging with less accurate signal-generation models, which has also been indicated in (Li, Uysal, & Mulvey, 2022). This leads to a second path that a practitioner can follow. To improve the results of the regime-based portfolio optimisation the practitioner can apply a predictive algorithm first, and then use an optimal control

solution for asset weight optimisation. This approach corrects the “lagging” limitation of the regime-detecting method while increasing the accuracy of inferred probabilities, and thus can potentially improve the overall portfolio performance constructed with the use of a technique such as MPC. Chapter 5 of this thesis will investigate this problem further by feeding results of a predictive framework from Chapter 4 into a highly promising MPC algorithm.

### **2.3.4 Does technical analysis add value in building portfolios?**

The value of technical analysis in portfolio optimisation, either by improving profits, mitigating risk or predicting markets, has been a topic of heated debate between academics and industry professionals. Early work (pre-1990s) provided evidence of sizeable net profits due to applying technical analysis to trading, although only for foreign exchange and commodity markets. However, these results were disputed. Arguments that would render these results unreliable have focused on issues with testing procedures (such as excluding transaction costs), avoiding risk measures, *ex post* selection of trading rules, lack of out-of-sample results, and, primarily, data-snooping bias. Data snooping has been a common error in work that provides evidence that high Sharpe ratio portfolios can be built with the use of TA. This bias occurs when too many parameters are refined to improve a model’s (or strategy’s) performance on a single data set. Trading rules that are tested on the same data set are prone to exaggerate the results which ultimately leads to failure once the same strategy is applied to different data (e.g., different asset class) (Zhu & Zhou, 2009). However, many modern works (after late 1990s) addressed the majority of these caveats and even so indicated that applying TA to trading strategies could generate consistent profits in various markets, such as equity, foreign exchange, and commodity futures. However, at the same time, it should be noted that there was an indication of strategy profitability decay in some areas; many studies showed that technical trading strategies generated higher returns for US stock markets until late 1990s but not thereafter. Similarly, with regards to foreign exchange, using technical analysis resulted in consistent profits up until early 1990s, after which these profits

declined or disappeared completely. Overall, out of 95 studies between 1988 and 2004, 56 provided positive evidence that TA added value to asset allocation. Among the research arguing that technical analysis can add value to the investment process compared to a simple buy-and-hold strategy are (Brock, Lakonishok, & LeBaron, 1992), (Lo, Mamaysky, & Wang, 2000), and (Gunasekarage & Power, 2001). However, the rest of this work could be categorised as mixed (19) or even negative (20) in its conclusions about the value of TA (Park & Irwin, 2007).

Interestingly, more modern reviews of TA papers show much favourable results with regards to the profitability, predictability, and risk of using technical analysis in portfolio optimisation; out of 85 recent academic works, only 6 did not support TA (Nazário, Lima e Silva, Sobreiro, & Kimura, 2017). This stemmed from the fact that recent papers included more studies on the use of TA strategies on emerging markets, which were already mentioned in (Park & Irwin, 2007) as being able to generate substantial profits. In addition, with the rise of machine learning algorithms and artificial neural networks, technical analysis has served to generate features able to enhance asset return predictability, particularly with regards to price movement direction (Huang, Nakamori, & Wang, 2005), (Ming, 2009), (Chang, Wang, & Zhou, 2012), (Ratto, et al., 2018), a phenomenon that was substantially omitted in the review of (Park & Irwin, 2007). Finally, either by itself or in combination with other factors—for instance, macroeconomic factors—technical indicators have been found to show statistically and economically out-of-sample predictive power in relation to equity risk premium, a critical component in modelling portfolio return expectations and thus determining asset allocation (Neely, Rapach, Tu, & Zhou, 2014).

Out of a variety of technical indicators tested in the quoted papers, trading rules using simple moving averages have been the most popular and profitable, even after correcting for the data snooping bias (Hsu & Kuan, 2005), (Zhu & Zhou, 2009), (Pavlov & Hurn, 2012). Even so, moving averages have issues with the choice of window length (see Section 2.1.3); a too-short window will capture changes in trend faster but crosses the price too often, increasing transaction costs and thus decreasing portfolio performance, whereas a too-long window avoids frequent whipsaws but lags the price too much, thus causing the trader to miss out on gains

or increase losses (Ellis & Parbery, 2005), (Shynkevich, 2012). Notably, however, Kaufman's Adaptive Moving Average (KAMA, see Section 2.2.2) corrects for that lag issue and has been found to outperform strategies based on simple moving averages (Ellis & Parbery, 2005). However, KAMA, as a standalone component responsible for timed trend detection, cannot be solely used to capture regime switches in data, due to its inability to detect variance shifts. The Markov-switching regression model outlined in Section 2.2.1, and further described on following page, is the missing piece needed to complete the whole regime-switching framework.

### **3. Improving on the Markov-switching regression model by the use of an adaptive moving average**

As described in Section 2.1.3, an accurate labelling technique is relevant for subsequent signal generation that is a critical part of the entire quantitative trading system. Section 2.3.1 discussed initial promise of applying Markov-switching model to this task; however, despite its success in financial applications, the two-state Markov-switching regression model does not perform flawlessly. Often the transition from low- to high-volatility regimes comes smoothly rather than abruptly, which results in a period of medium variance. In addition, volatility itself is not a definite indication of up or down trending markets; these can also rise during increased volatility periods, as well as fall throughout decreasing-variance regime. This phenomenon has spurred alternating the Markov-switching regression framework with three states to capture the full market dynamics and improve the model's efficacy. However, this approach can suffer from instability by introducing overly-frequent regime switches. Subsequent prediction of such shifts and chaotic signal generation may adversely impact portfolio returns by increasing costs of trading, thus potentially reduce the need for real-world applications.

The aim of this chapter is to address this shortcoming by balancing the smoothness of two-state Markov-switching regression with a method that is both stable (avoiding spurious regime shift detections) and accurate in locating the times of onset of regime switches. Since merely moving from two to three states is not optimal, it is proposed to add KAMA (see Section 2.2.2) to the two-state model in order to hopefully more accurately discern regimes within various markets.

This study is structured as follows. First, background and relevant literature is reviewed, including other models used in the study for comparison purposes. Second, data and experimental methodology are outlined. Finally, the results are discussed and the work is concluded mentioning potential future extensions.

### 3.1 Background and related work

#### 3.1.1 Popular extensions to the two-state Markov-switching regression model

The “standard” two-state Markov-switching regression model (see Section 2.2.1) has been extended by a third, medium volatility regime to detect periods with neither low nor high volatility (Boldin, 1996), (Kim, Nelson, & Startz, 1998). By segmenting the medium variance period, it has been possible to potentially improve the accuracy of the two-state model by extracting regimes that were harder to determine. Due to the three-state model’s ability of selecting both persistent (low, high) and transitory (medium) regimes, two versions of these models are later considered as comparison models: three-state Markov-switching dynamic regression and the three-state variance-switching regression models of (Kim, Nelson, & Startz, 1998). The former merely takes the two-state Markov-switching dynamic regression model and alternates the number of states, so that in the following equation (same as Equation 1):

$$r_t = \mu_{S_t} + r_{t-1}\beta_{S_t} + \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1),$$

$S_t \in \{1,2\}$  changes to  $S_t \in \{1, 2, 3\}$ . The latter eliminates the intercept and lag terms, leaving just the state-dependent volatility. Using the same annotation as in the Equation 1, the Kim, Nelson, & Startz equation is expressed as

$$r_t = \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1), \quad (16)$$

where  $S_t \in \{1,2,3\}$ .

Still, the literature outlined below mentions limitations of applying three-state models to financial datasets; however, there has also been a promising approach that uses technical analysis in order to overcome some of the issues of the two-state models which deemphasises the use of the three-state models.

### 3.1.2 Related work

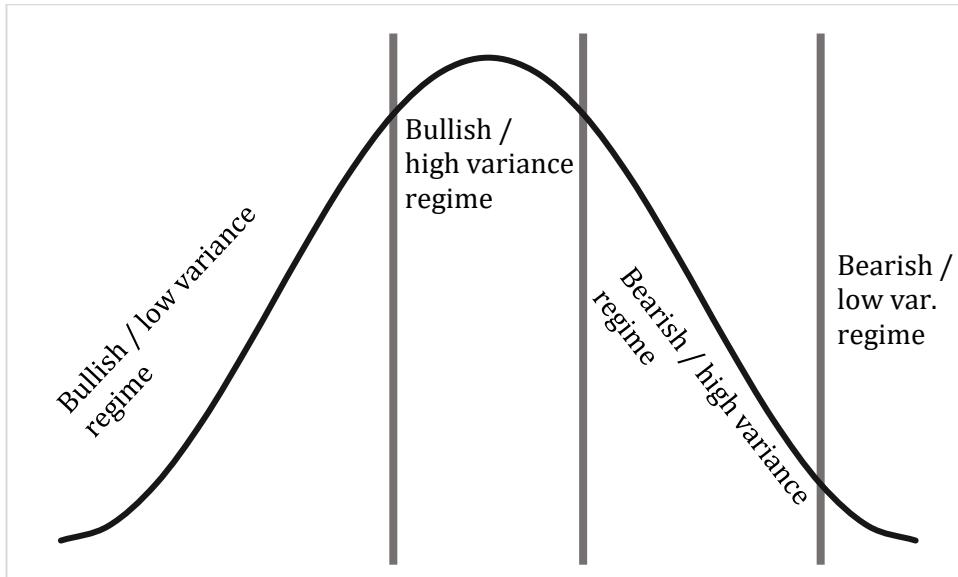
#### *Extensions to the two-state Markov-switching regression model*

The original, two-state model has become a starting point for further upgrades, since it is not optimal due to its failure to capture transitory regimes (Boldin, 1996). As described in Section 3.1.1, one promising solution for detecting both persistent and transitory states is increasing the number of regimes within the model from two to three. This method, however, comes at a cost, since generally the three-state Markov-switching regression models are considered to be unstable, as they may shift regimes too frequently (Hauptmann, Hoppenkamps, Min, Ramsauer, & Zagst, 2014). There have also been many attempts to increase the number of states in MSR to four, six, or even eight (Guidolin, 2011). Even though the four-state MSR model looks promising (e.g., (Billio, Casarin, Ravazzolo, & van Dijk, 2012)), there are concerns about potential overfitting, as it has been found that knowing the number of regimes existing in the underlying time series beforehand has a critical influence on a model's performance (Bulla, 2011). Given a universe of different assets and asset classes, in order to check if the three- or more-state MSR model is suitable, one would have to iteratively apply each version of the MSR model to each asset in order to find the optimal number of states, hoping that the same number of regimes will continue in the future, a complex process that has clear potential to overfit the data. One potential solution to tackle this overfitting could be to run the adequate tests for optimal number of regimes, such as the unit root test or the likelihood ratio test proposed in (Nelson, Piger, & Zivot, 2001) and (Kasahara & Shimotsu, 2018), respectively; however, this still does not solve the issue of overly-frequent switches—on the contrary, there is a high potential of increasing them.

Another potential solution is to overlay the two-state Markov model with moving averages in order to detect transitory regimes without frequent state switches (Srivastava & Bhattacharyya, 2018). The moving averages divide asset prices into bullish and bearish periods (see Section 2.1.4), and two-state Markov-switching regression split series into low and high volatility regimes, resulting in separating the underlying series into four regimes. This is in line with the classic

Wyckoff theory (see Figure 23 below) of four price cycles which distinguishes the following states within asset prices: advance (low variance / bullish), distribution (high variance / bullish), decline (high variance / bearish), and accumulation (low variance / bearish) (Wyckoff, 1937), (Srivastava & Bhattacharyya, 2018).

Figure 23: Wyckoff Price Cycle.



### Model of Srivastava and Bhattacharyya

An interesting model was presented by (Srivastava & Bhattacharyya, 2018) that indeed uses the Wyckoff's theory, and by blending the two-state Markov model with a technical overlay, smoothly extracts the described four states over global rather than local price swings. This model will be referred to here as the *WorldQuant Model* (WQ model). However, although the WQ model has been found to add value in equity strategies, it is constrained by the data required to calculate the technical indicator. The model uses a custom Keltner Channel (Ruiz-Franco, Jiménez-Gómez, & Lambis-Alandete, 2018) constructed in this instance with a triangular moving average and an average true range indicator (ATR) to split series into bullish and bearish regimes. The computation of ATR, a volatility factor in the Keltner Channel (loosely similar to ER in KAMA), requires period-high and low prices. This significantly limits an application of this model to assets with a short history of high and low prices, even when their closing prices have been available for a much longer time. Such issue causes the training of the WQ model to miss major economic events in the past,

e.g., the 1990s Asian crisis in case of emerging market equities or the dot-com bubble for the US IT sector. Thus, even though the results of applying the WQ model to equity markets show promise, it may only be a partial solution for the problem of accurately detecting persistent and transitory regimes without frequent shifts.

The work of this report has taken inspiration from the WQ model by replacing the Keltner Channels with the Kaufman's Adaptive Moving Average, enabling the new method to solve the issue of shorter price history while being less likely to damage the model's performance. In addition, the work of this report can be expanded from the original area of application of the WQ model, equities, to other asset classes to which the Markov switching method has been applied, such as commodities, foreign exchange, and fixed income. It will be seen that the new method has a wide-ranging ability to distinguish regimes.

## 3.2 Data and methodology

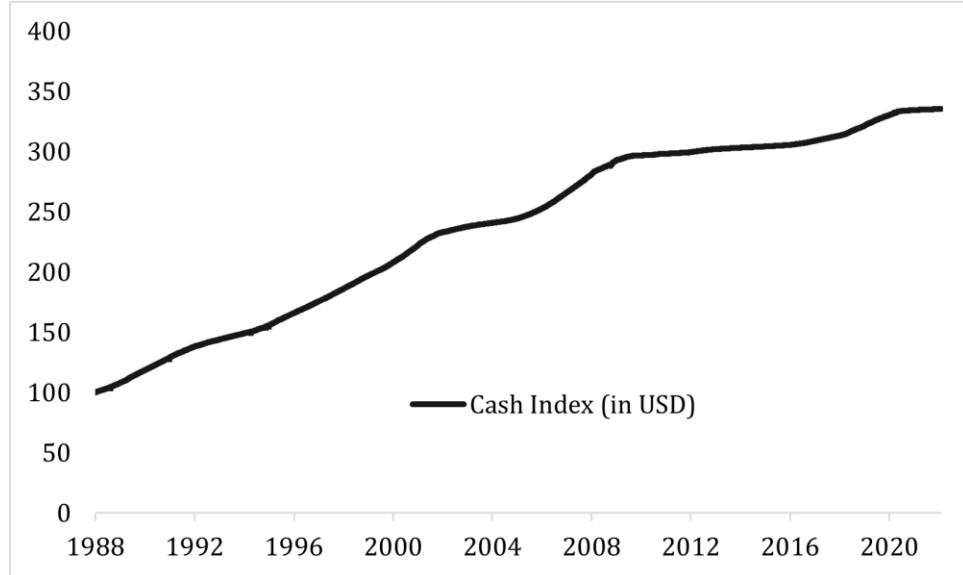
### 3.2.1 Data

Setting up each of the models described in the previous section (3.1) requires daily closing price  $P$  data, apart from the WQ Model which additionally necessitates daily highest  $P_h$  and lowest  $P_l$  prices to calculate Keltner Channels. All Markov-switching regressions utilise daily log returns  $r_t = \ln P_t - \ln P_{t-1}$  instead.

There are 56 assets in total divided into four classes: equities (24), exchange rates (FX) (13), commodities (12), and fixed income (7). Their choice is based on extent of use in the financial industry (for instance, major equity markets, G10 countries for FX), thus there will be sufficient liquidity to minimise the risk of market manipulation. In addition to the above-listed assets, a cash index is also used in the trading strategy detailed later in Section 3.2.2 on methodology. The rationale behind the use here of cash index stems from its low volatility and ever-rising trend. With the rare exception of periods of negative interest rates, such as after the financial

crisis of 2008-2009, it is nearly impossible to lose whilst betting on cash in the longer term, which makes it a useful “money-parking” tool (see Figure 24 below).

*Figure 24: The history of holding and rebalancing cash every 3 months (in USD).*



The table below presents asset names, instruments, time horizons, and data sources for the data used in the experiments of this chapter. For each asset, that 85% of the data closest to the start date is used for optimisation of the KAMA model, using procedures to be described below, while that 15% of the data closest to the end date is used for out of sample testing.

*Table 3: Data used in the work of Chapter 3.*

Asset class / asset	Instrument type	History: Phase 1	History: Phase 2	Source
<b>Equities</b>				
MSCI USA	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, MSCI
MSCI Emerging Markets	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, MSCI
NASDAQ 100	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, NASDAQ
Stoxx 600	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, STOXX
Eurostoxx	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, STOXX
DAX	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, Xetra
CAC 40	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, Euronext
FTSE MIB	ETF	01/07/2003 - 08/01/2021	01/01/2000 - 26/03/2021	Bloomberg, LSE
FTSE 100	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, LSE

Nikkei 225	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, TSE
Hang Seng	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, HSE
Shenzen Composite	ETF	31/12/2004 - 08/01/2021	06/01/2002 - 26/03/2021	Bloomberg, SSE
TSX Composite	ETF	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, S&P
ASX 200	ETF	31/01/2001 - 08/01/2021	31/05/1992 - 26/03/2021	Bloomberg, S&P
MSCI USA Consumer Discretionary	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
MSCI USA Consumer Staples	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
MSCI USA Communication Services	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
MSCI USA Energy	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
MSCI USA Financials	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
MSCI USA Healthcare	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
MSCI USA Industrials	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
MSCI USA Information Technology	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
MSCI USA Materials	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
MSCI USA Utilities	ETF	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg, MSCI
<b>Foreign Exchange</b>				
Australian Dollar / Japanese Yen	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
Swiss Franc / US Dollar	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
DXY	Index	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
Euro / Russian Rubel	Spot	31/01/2001 - 08/01/2021	01/01/2000 - 26/03/2021	Bloomberg
Euro / US Dollar	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
UK Pound Sterling / US Dollar	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
US Dollar / Australian Dollar	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
US Dollar / Canadian Dollar	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
US Dollar / Danish Krone	Spot	31/01/2001 - 08/01/2021	31/12/1994 - 26/03/2021	Bloomberg
US Dollar / Japanese Yen	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
US Dollar / Norwegian Krone	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
US Dollar / New Zealand Dollar	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
US Dollar / Swedish Krona	Spot	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg
<b>Commodities</b>				
Aluminium	Futures	17/05/2001 - 08/01/2021	17/05/2001 - 26/03/2021	Bloomberg, SHF
Brent Crude Oil	Futures	31/01/2001 - 08/01/2021	23/06/1988 - 26/03/2021	Bloomberg, ICE
Coffee	Futures	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, NYB
Copper	Futures	31/01/2001 - 08/01/2021	06/12/1988 - 26/03/2021	Bloomberg, CMX
Corn	Futures	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, CBT
Gold	Futures	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, CMX
Live Cattle	Futures	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, CME

Natural Gas	Futures	31/01/2001 - 08/01/2021	03/05/1990 - 26/03/2021	Bloomberg, NYMEX
Nickel	Futures	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, LME
Soybeans	Futures	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, CBT
Sugar	Futures	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, NYB
Wheat	Futures	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, CBT
<b>Fixed Income</b>				
US Long Bonds 7-10 Years	ETF	26/07/2002 - 08/01/2021	26/07/2002 - 26/03/2021	Bloomberg, iShares
US Short Bonds 1-3 Years	ETF	26/07/2002 - 08/01/2021	26/07/2002 - 26/03/2021	Bloomberg, iShares
US Ultra-Long Bonds 20+ Years	ETF	26/07/2002 - 08/01/2021	26/07/2002 - 26/03/2021	Bloomberg, iShares
US Treasury Inflation Protected Securities ("TIPS") Bonds	ETF	05/12/2003 - 08/01/2021	05/12/2003 - 26/03/2021	Bloomberg, iShares
US Dollar-denominated Emerging Markets Bonds	ETF	19/12/2007 - 08/01/2021	19/12/2007 - 26/03/2021	Bloomberg, iShares
US High Yield Corporate Bond	ETF	11/04/2007 - 08/01/2021	11/04/2007 - 26/03/2021	Bloomberg, iShares
US Investment Grade Corporate Bond	ETF	26/07/2002 - 08/01/2021	26/07/2002 - 26/03/2021	Bloomberg, iShares
<b>Supplementary</b>				
USD Cash 3-Month Rebalancing	Deposits	31/01/2001 - 08/01/2021	31/12/1987 - 26/03/2021	Bloomberg, JP Morgan

Two instruments require further clarification. By “currency” it is meant raw monies that can be bought and sold in any currency exchange. The only exception within the class of currencies is the DXY Index which tracks the overall strength of US Dollar by weighting six currencies in a basket: Euro, UK Pound Sterling, Japanese Yen, Canadian Dollar, Swiss Franc, and Swedish Krona. The other instrument used in this research that requires further explanation is an exchange-traded fund pertaining to the fixed income class. Even though the related work section (3.1.2) mentioned the implementation of Markov-switching regression models to interest rates, it was not appropriate to do this here, since yields cannot be traded directly, and therefore ETFs following selected bonds were employed instead.

### 3.2.2 Methodology

#### *Outline of the proposed work*

This section is constructed as follows: first, the process of combining the two-state MSR model with KAMA is described. This is also followed by a short paragraph on

turning multiple-state models into two-state with reasoning given. Second, the optimisation of the KAMA component within the combined model is outlined. Third, benchmark models used for performance comparison are briefly discussed, and finally the trading strategy, which assesses the tested models' abilities to separate regimes, is described. The trading strategy is additionally split into two phases:

**Phase 1: Comparison between regime detection abilities of the models.**

Here the proposed regime-switching model is compared to benchmark models in-sample (the 85% of the dataset used for training and validation) to distil the most promising candidates for out-of-sample testing (the remaining 15% of the dataset) which is performed in Phase 2 explained below. The reason for just an in-sample study is such that the data are restricted by the high and low-price data availability for the WQ model, an essential comparison model given its role in initially triggering the multi-state model work described here. To keep uniformity between assets during this phase, almost every series begins in January 2001 at the earliest, with some exceptions, such as fixed income, whose price data is not available since this period. Thus, the data used in this phase are not sufficient to have an out-of-sample test period, so while the results can be used to identify the most promising models from the available set, these can be tested fully on in the next phase.

**Phase 2: Use of most promising models for out-of-sample trading.** In this phase the WQ model will be set aside, since, while it was an important inspiration for the work of this report, it will be seen that the regime detection results from it are not strong. This allows for the collection of more data and for the provision of an out-of-sample test period. During this period, the ability of the better performing models to detect recent-past regime switches that should result in a change of trading strategy is tested, and the most profitable model determined. (Note: work on forecasting, as opposed to this current use of nowcasting, will be set aside as future work, for inclusion in later thesis chapters.)

## *Enhancing the two-state Markov-switching regression model signals with KAMA*

This section, which is relevant to both phases of the work, will look at the details of the way in which KAMA has been integrated into the Markov-switching framework, in line with the inspiration provided by the WQ model, as described at the end of the Section 3.1.2 on related work.

**Integration of KAMA with MSR.** The proposed regime-switching model is initiated with a two-state Markov-switching regression model (MSR) to detect high and low variance periods within each selected asset. Considering states  $S_t \in \{1, 2\}$  and the 50% level as a cut-off point for smoothed probabilities in MSR model, this results in the following separation:

- Low variance regimes, defined as ones where the filtered probability of state  $S_t = 1$  is higher than 50%.
- High variance regimes, defined as ones where the filtered probability of state  $S_t = 2$  is higher than 50%.

Kaufman's Adaptive Moving Average works as an overlay to subsequently divide low and high volatility periods into bullish and bearish regimes. However, its practical application also requires a mechanism to generate trading decisions, and for this purpose Kaufman's Adaptive Moving Average is embedded within a construction called the *filter* which generates a signal to enter or exit positions of interest. Over a  $t$ -day period, filter  $f$  is computed as

$$f_t = \gamma \sigma(KAMA_t), \quad (17)$$

where

$$\sigma(KAMA_t) = \sqrt{\left( \sum_{t=1}^n x_t^2 - \frac{\sum_{t=1}^n x_t}{n} \right)}, \quad (18)$$

and

$$x_t = KAMA_t - KAMA_{t-1}. \quad (19)$$

This essentially states that the filter  $f$  is a multiplication of the parameter  $\gamma$  and the standard deviation of the change in KAMA over  $n$  days, where  $n \leq t$ . The parameter  $\gamma$  is theoretically locked within bands of  $0.05 \leq \gamma \leq 1$ ; however, it is suggested it may oscillate around 0.15 (Kaufman, 1995), (Ellis & Parbery, 2005). Calculating both KAMA and the filter allows the construction of a strategy for trading in bullish and bearish regimes:

- Bullish, hence buy, when KAMA advances above its low over a prior period of  $n$  days by a value greater than the filter.
- Bearish, hence sell, when KAMA descends below its low over a prior period of  $n$  days by a value greater than the filter.

Thus, combining the above with the MSR model's results, the proposed regime-switching model separates the underlying price series into four regimes:

- Low variance and bullish when the filtered probability of state  $S_t = 1$  is higher than 50%, and KAMA rises above its low over a prior period of  $n$  days by a value greater than the filter.
- Low variance and bearish when the filtered probability of state  $S_t = 1$  is higher than 50%, and KAMA falls below its high over a prior period of  $n$  days by a value greater than the filter.
- High variance and bullish when the filtered probability of state  $S_t = 2$  is higher than 50%, and KAMA rises above its low over a prior period of  $n$  days by a value greater than the filter.
- High variance and bearish when the filtered probability of state  $S_t = 2$  is higher than 50%, and KAMA falls below its high over a prior period of  $n$  days by a value greater than the filter.

In this study, the low variance / bullish and high variance / bearish states are considered persistent, whereas low variance / bearish and high variance / bullish regimes are considered transitory. Hereafter, these four states are also referred to as "labels" or "classes". In addition, the proposed regime-switching model will also be referred to as the *KAMA+MSR* model.

The two phases described in the beginning of this section will incorporate two versions of KAMA+MSR model: four-state (low variance / bullish, high variance

/ bullish, low variance / bearish, high variance / bearish) and two-state (low variance / bullish and high variance / bearish). As will be explained and shown later in the results section (3.3), incorporating the two-state version of the model is necessary to fairly compare its performance particularly with the “standard” two-state Markov-switching regression.

**Turning multiple-state models into two-state.** This stage leaves the persistent states (low variance / bullish, high variance / bearish) within the KAMA+MSR model, while treating the transitory states (high variance / bullish, low variance / bearish) as periods in which no trading action should be taken. Such operation effectively turns the four-state KAMA+MSR model into a fine-tuned two-state model focusing only on persistent states. Similar procedure is done on the benchmarked WQ model in which the transitory states are removed, as well as on the three-state Markov models where the medium variance regimes are omitted, thus turning the original 4- and three-state models, respectively, into two-state.

**Optimisation of the KAMA Component of the Model.** In contrast to the two-state Markov-switching dynamic regression model, which is hyperparameter-free, KAMA isolates a vector of hyperparameters  $\boldsymbol{\theta}_h$  that needs to be optimised over the training period that comprises the initial 85% of the data<sup>33</sup>. To recall the description in Section 2.2.2 introducing KAMA, this parameter vector is given by

$$\boldsymbol{\theta}_h = (n, n_s, n_l, \gamma), \quad (20)$$

where  $n$  is a moving window for the efficiency ratio  $ER$  and the filter  $f$ ,  $n_s$  is a moving window for a short-term smoothing constant  $k_s$ ,  $n_l$  is a moving window for a long-term smoothing constant  $k_l$ , and  $\gamma$  is the control parameter in the filter  $f$  term. Clearly these KAMA parameters will influence the output of the model and its profitability, and hence will need to be treated as hyperparameters. (It is important to recall at this point that model comparison results for Phase 1 will be reported for

---

<sup>33</sup> The reason for 85%/15% data split comes from the fact that the 15% of out of sample data covers close to four years of data which spans over various important market events and is highly probable to include all four regimes studied.

validation data only, with out-of-sample testing for the most promising models done later, in Phase 2.)

Beginning with the initial vector of parameters  $\boldsymbol{\theta}_h$ , the algorithm constructs the proposed regime-switching model by overlaying the calculated KAMA on a two-state Markov-switching dynamic regression. Based on the outcome of this initial combination, the model splits asset price into multiple periods (“segments”) which are classified as one of the four regimes. Subsequently, the price slope (bullish vs. bearish) and log returns volatility (low vs. high variance) of each segment is computed, so that the optimised model’s accuracy can be appropriately inspected. The parameter values are then adjusted to optimise performance during the training period.

The test of meaningful separation of regimes is done using a K-Means clustering<sup>34</sup> with a 2-step walk-forward cross-validation and a custom scoring function. In order to implement cross-validation, the regime slope and volatility data is split into non-shuffled training and validation sets, whereby training is 50% and 75% of the training dataset in the first and second steps, respectively, and the consecutive 25% of the data are used for validation in both steps. The clustering ability of K-Means is employed on the sliced training sets to investigate whether the already refined regimes differ by slopes and volatilities. Given that this will be a four-state model, K-Means here assumes four clusters; the clustering algorithm additionally uses ten random initialisations of centroids, the same random seed for each model, and 300 maximum iterations per initialisation. The K-Means method then predicts clusters for the validation sets, and the generated predictions are contrasted with the model’s initially chosen regime labels.

To achieve this, the KAMA parameter optimisation algorithm uses a custom function called the misclassification score. This scoring function groups clusters and regime labels into a  $4 \times 4$  matrix in order to analyse the dominant number of

---

<sup>34</sup> As discussed in Section 2.3.1 on machine learning-based regime detection methods, K-Means is a clustering method that aims to partition  $n$  observations into  $k$  clusters in which each observation belongs to the cluster with the nearest mean. For technical details on K-Means, refer to (Arthur & Vassilvitskii, 2006).

detected segments (periods of time that fall into a given class) within each row and column, as in the example table below, and perform appropriate actions.

*Table 4: Example matrix grouping clusters and regime labels.*

Number of extracted segments	Low variance / Bullish	Low variance / Bearish	High variance / bullish	High variance / bearish
<b>Cluster 1</b>	1	0	0	0
<b>Cluster 2</b>	0	5	1	9
<b>Cluster 3</b>	0	0	2	0
<b>Cluster 4</b>	3	0	3	0

The misclassification score initially loops over rows to subtract the dominant number of segments from the row sum. Thus, in the example (Table 4), in Clusters 1 and 3, since there is only one regime detected, the score is 0, which indicates lack of misclassification within this cluster. However, in Cluster 2, there are 6 segments (out of 15) whose underlying slope and volatility characteristics do not resemble the dominant label, high variance and bearish, so the score is 6. In Cluster 4, there are two dominant labels, each mismatched (with respect to the other) with a score of 3, and we would in a case like this assign an overall score for the cluster of 3. The number of misclassifications per cluster is then summed up to indicate how many segments have been inaccurately labelled by the optimised model. The total number is finally divided by the length of the predicted dataset to form a ratio that can be compared between optimisation trials. Ideally, the final misclassification score should equal 0, representing a perfect alignment between the K-Means discovered clusters and the proposed regime-switching model, though this degree of alignment is unlikely in practice and hence the optimisation algorithm is run 50 times to find the best-available vector of parameters  $\theta_h$  for the training period, which are saved for use in the test period.

Figure 25 on the following page presents all four regimes distinguished by the described KAMA+MSR model on MSCI USA Index after optimising KAMA hyperparameters.

Figure 25: All four regimes separated by the KAMA+MSR model on MSCI USA Index.



#### *Preparing other models for the comparison stage*

As explained earlier, the KAMA+MSR model is tested in both Phases against variations of three different Markov-switching regression models, as well as, in Phase 1, the WQ Model, which mirrors the proposed method the closest. See Table 5 on the following page for a summary:

Table 5: Models used for comparison.

Name	Equation
<b>Two-state Markov-switching dynamic regression model (2S MSR, two-state MSR)</b>	$r_t = \mu_{S_t} + r_{t-1}\beta_{S_t} + \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1),$ where $S_t \in \{1, 2\}$
<b>Three-state Markov-switching dynamic regression model (3S MSR, three-state MSR)</b>	$r_t = \mu_{S_t} + r_{t-1}\beta_{S_t} + \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1),$ where $S_t \in \{1, 2, 3\}$
<b>Three-state Markov-switching dynamic regression model turned to two-state (3S 2S MSR)</b>	$r_t = \mu_{S_t} + r_{t-1}\beta_{S_t} + \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1),$ where $S_t \in \{1, 3\}$
<b>Kim, Nelson, &amp; Startz three-state variance-switching regression model (3S KNS, three-state KNS)</b>	$r_t = \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1),$ where $S_t \in \{1, 2, 3\}$
<b>Kim, Nelson, &amp; Startz three-state variance-switching regression model turned to two-state (3S 2S KNS)</b>	$r_t = \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1),$ where $S_t \in \{1, 3\}$
<b>WorldQuant model (WQ)</b>	$r_t = \mu_{S_t} + r_{t-1}\beta_{S_t} + \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1),$ + Keltner Channel overlay, where $S_t \in \{1, 2\}$
<b>WorldQuant model turned to two-state (WQ 2S)</b>	$r_t = \mu_{S_t} + r_{t-1}\beta_{S_t} + \sigma_{S_t}\varepsilon_t, \quad \varepsilon_t \sim N(0, 1),$ + Keltner Channel overlay, where $S_t \in \{1, 2\}$ and transitory states are omitted

To recall, setting up Markov-switching regressions does not require any parameter optimisation; thus, the benchmarks are merely fitted to the 56 different asset prices described in Section 3.2.2 on data. In case any three-state model fails to discern three regimes within price series, its application to this particular asset is discarded and it does not participate in further steps.

The WQ Model, on the other hand, is optimised identically to the proposed regime-switching model, so that it is not left at disadvantage. The optimisation algorithm tries to find the best hyperparameter combination within a vector  $\boldsymbol{\theta}_{h_{WQ}}$

$$\boldsymbol{\theta}_{h_{WQ}} = (n_{TRIMA}, n_{ATR}), \quad (21)$$

where  $n_{TRIMA}$  represents the triangular moving average period, whereas  $n_{ATR}$  is a time window for the average true range, both indicators being components of the Keltner Channel. The parameter vector  $\boldsymbol{\theta}_{h_{WQ}}$  of the trial with the lowest misclassification score is stored and exploited in the subsequent trading phase for replication.

## *Implementing a suitable trading strategy for performance testing*

**Initial considerations of the trading strategies.** Trading-based performance testing in Phase 1 (in-sample, all models considered) is divided into two stages: in the first, every model retains all states; in the second, models with more than three states use just the two persistent regimes, leaving the transition regimes inactive whilst making transactions, as discussed previously. In Phase 2, as will be seen, only two-state models will be considered, so the second of the above stages is not needed.

No matter the stage, the process is identical. The optimization process involves assigning different weights to two assets (a selected asset from Table 3 vs. USD Cash 3-Month Rebalancing, the supplementary asset from Table 3) with the total weight summing to 100%. This process is repeated 1000 times to determine the optimal allocation. The optimization is carried out using the earliest 85% of the available data for each asset. During this process, the effectiveness of 1000 random weight allocations is compared to determine the best possible allocation. The code subsequently calculates both returns and *adjusted Sharpe ratio* (ASR) (Israelsen, 2005) (explained later), on a per-segment basis, within both assets and cash, and multiplies them by the initiated weights.

The reason for calculating both metrics (returns and ASR) instead of just ASR stems from the fact that risk-adjusted returns (such as ASR) are not the principal means of comparison because of the use of cash, which is characterised by nearing-zero volatility (approximately 0.01% over the studied period). Instead of putting pressure on profit, the algorithm emphasises variance by safely allocating larger weights to cash, which simultaneously enlarges the adjusted Sharpe ratio, no matter the regime nor the model tried. This clearly distorts the comparison process, thus annualised weighted returns are also necessary and must take priority as a comparison measure.

It should also be noted that the asset returns are diminished by two-way transaction costs, i.e., during buying and selling. Table 6 on the following page lists the costs assumed in this work:

Table 6: Two-way trading costs per each asset class (as given in (Edelen, Evans, & Kadlec, 2013)).

Asset class	Brokerage commissions	Bid-ask spread	Market impact	Total cost
<b>Equities</b>	0.14%	0.13%	0.53%	0.8%
<b>Currencies</b>	0%	0.13%	0%	0.13%
<b>Commodities</b>	0.14%	0.13%	0%	0.27%
<b>Fixed Income (ETFs)</b>	0.14%	0.13%	0.53%	0.8%

Even though the costs presented in the table above pertain to average trading costs of US equity mutual funds over an eleven-year sample they have been spread to all asset classes for simplification purposes. However, the reason why they are not equal between the classes stems from the different sizes of these markets and a potential influence a trader may have on them. For instance, it is difficult to have any significant impact on foreign exchange, unless the trader is an intervening central bank. Additionally, since in this research raw currencies are traded, there are no brokerage commissions, and the only cost remaining is the bid-ask spread. In case of commodities, the trader must account for brokerage fees, as well as the bid-ask spread on futures contracts. However, only on rare occasions the trade would be large enough to significantly move the entire market. Regarding equity indices, as well as fixed income ETFs, the full cost is taken into consideration, since on the top of commissions and the bid-ask spread, the trading entity must account for a potential market impact.

**Assessing the trading strategy.** As mentioned previously, the algorithm calculates returns and the adjusted Sharpe ratio to assess the strategy. The annualised (assuming 252 trading days within a year) ASR is given by,

$$ASR = \frac{\mu(r_M)}{\sigma(r_M) / \mu|r_M|} \times \sqrt{252} \quad (22)$$

where  $\mu(r_M)$  are mean cost-adjusted returns  $r_M$  over the analysed period,  $\sigma(r_M)$  is the standard deviation of these cost-adjusted returns, and  $\mu|r_M|$  stands for mean cost-adjusted returns calculated from only positive returns. The adjusted Sharpe ratio penalises negative returns in particular, treating positive returns in the same manner as the standard Sharpe ratio.

The KAMA+MSR model is ultimately compared to the other models with a “winning score ratio”  $WS$ , calculated as

$$WS = \frac{(Z_{win} - Z_{runner})}{Z_{win}}, \quad (23)$$

where  $Z_{win}$  indicates the model with the highest weighted annualised returns<sup>35</sup> or adjusted Sharpe ratio, and  $Z_{runner}$  adheres to the second-best method. By computing the ratio in the above Equation 23, it is possible not only to display the winner, but also by how much better it is relatively to the closest “opponent”.

The winning score ratio is obtained for both weighted annualised returns and adjusted Sharpe ratio. The two results are subsequently combined to reveal the model with the most risk-reward approach to detecting financial regimes.

**Phase 2 in comparison to Phase 1.** In Phase 2 three best performing models based on strategy results from Phase 1 are tested on the holdout sample. The optimisation and scoring procedures are identical, however, the data is now split in an 85% training and validation sample, and a 15% holdout sample. Since the WQ model is excluded from this stage, there is no need for high and low prices, thus the majority of data begins in the late 1980s when the cash index used in the trading strategy commences. The major exception is fixed income whose price history remains the same due to data unavailability for the selected ETFs (see Section 3.2.1 on the data used in the experiment).

The optimisation algorithm tries to determine the best KAMA parameters for each asset over the 85% sample, and subsequently stores the results for the trading stage. The trading algorithm seeks the most profitable allocation weights also on the 85% sample first, however, later implements them on the 15% holdout sample to calculate the scoring statistics for the final comparison.

---

<sup>35</sup> Weighted annualised returns are calculated as:  $(w_1 r_1 + w_2 r_2) \times 252$ , since there are only two assets in each trial and 252 trading days.

### 3.3 Results

This section describes results for Phases 1 and 2, and is finished with a discussion.

#### 3.3.1 Phase 1

The implemented strategy from Section 3.2.2 results in multiple tables per model and asset which demonstrate statistics from the best in-sample trial. The MSCI USA Index is given as an example in Table 7 below.

*Table 7: Example results for the KAMA+MSR model.*

KAMA+MSR Model	Weights / scores
Bearish_High_Var	0.008
Bearish_Low_Var	0.434
Bullish_High_Var	0.102
Bullish_Low_Var	0.977
Adj_Sharpe_ratio	<b>0.979</b>
Mean_weighted_returns	<b>0.151</b>
Weighted_volatility	0.133

The data from tables like Table 7 are used to calculate the winning score (see Equation 23) which is subsequently averaged over each asset within its class (for details relating to each individual asset, Appendix A). The three tables to follow present the final results from the competitor models for mean weighted annualised returns, adjusted Sharpe ratio, and a sum of both. (Depending on an investor's preference either returns or Sharpe ratio could be more highly weighted, but a ratio of 1:1 is the most reasonable assumption in the absence of such a stated preference.

Table 8: Phase 1 (in-sample) results.

Highest Adjusted Sharpe Ratio – average Winning Score Ratio									
Asset classes	Proposed model	WQ model	Markov 2S	Markov 3S	Markov KNS	Proposed model 2S	WQ model 2S	Markov 3S 2S	Markov KNS 2S
<b>Equities</b>	0.01	0.00	0.00	0.00	0.00	<b>0.07</b>	0.01	0.02	0.00
<b>Commodities</b>	0.00	0.00	0.03	0.00	0.00	0.03	0.05	<b>0.10</b>	0.04
<b>FX</b>	0.01	0.00	<b>0.04</b>	0.00	0.00	0.03	0.02	0.01	0.03
<b>Fixed Income</b>	0.00	0.00	<b>0.07</b>	0.01	0.00	0.01	0.00	0.04	0.01
Highest Weighted Annualised Returns – average Winning Score Ratio									
Asset classes	Proposed model	WQ model	Markov 2S	Markov 3S	Markov KNS	Proposed model 2S	WQ model 2S	Markov 3S 2S	Markov KNS 2S
<b>Equities</b>	0.00	0.00	<b>0.07</b>	0.00	0.00	0.05	0.00	0.00	0.00
<b>Commodities</b>	0.00	0.00	0.03	0.04	0.00	<b>0.11</b>	0.01	0.00	0.03
<b>FX</b>	0.00	0.00	0.00	0.00	0.03	<b>0.09</b>	0.01	0.02	0.00
<b>Fixed Income</b>	0.00	0.00	<b>0.13</b>	0.00	0.00	0.05	0.00	0.07	0.00
Combined Winning Score Ratio									
Asset classes	Proposed model	WQ model	Markov 2S	Markov 3S	Markov KNS	Proposed model 2S	WQ model 2S	Markov 3S 2S	Markov KNS 2S
<b>Equities</b>	0.01	0.00	0.07	0.00	0.00	<b>0.12</b>	0.01	0.02	0.00
<b>Commodities</b>	0.00	0.00	0.05	0.04	0.00	<b>0.14</b>	0.06	0.10	0.07
<b>FX</b>	0.01	0.00	0.04	0.00	0.03	<b>0.12</b>	0.04	0.03	0.04
<b>Fixed Income</b>	0.00	0.00	<b>0.20</b>	0.01	0.00	0.06	0.00	0.11	0.01

There is one clear pattern emerging from the presented tables: two-state models generate higher risk-adjusted returns in-sample (i.e., in Phase 1) than the multi-state models. However, this is anticipated, since transitory regimes (i.e., bullish / high variance and bearish / low variance in KAMA+MSR and medium variance in three-state Markov models) are, on average, characterised by lower returns and higher variance which automatically reduce both annualised profit and adjusted Sharpe ratio.

In the “Highest Adjusted Sharpe Ratio” section of Table 8 the results are mixed. The KAMA+MSR model is a clear winner in equities; however, it loses significantly with three-state Markov-switching regression model turned to two states in commodities. In FX, even though two-state Markov-switching model achieved the highest ASR, it is not that distinct, as the KAMA+MSR and two-state Kim, Nelson, & Startz models produced almost similar results. Finally, in fixed income the two-state MSR wins, however, only by three percentage points over the three-state MSR model turned into two states.

The situation clarifies slightly in the “Highest Weighted Annualised Returns” section of the table, in which there are two clear winners: the proposed regime-switching and two-state Markov-switching DR model. Although the two-state Markov model prevails within equities, the outcome is only marginally higher than the KAMA+MSR model’s; this is primarily attributable to superior results in NASDAQ 100 and MSCI USA Information Technology Index (see Appendix A). On the other hand, in commodities and foreign exchange, the proposed model evidently stands out, whereas in fixed income the two-state Markov model wins again.

When both metrics are summed, assuming equal weights between them, the KAMA+MSR model emerges as a winner in three out of four asset classes; the only exception is fixed income where the two-state MSR clearly prevails. The third best model is the three-state Markov model turned into two states which has achieved particularly good results in commodities and fixed income. It is notable that while the WQ model was an important inspiration for the work of this report, it is evidently not as successful as the other models.

Phase 1 selected as the best models the KAMA+MSR model, the two-state MSR model, and three-state MSR turned into two states. These three will be tested on the holdout sample in Phase 2.

### 3.3.2 Phase 2

Similarly to Phase 1, Phase 2 resulted in three tables that contain winning score ratios, this time using in-sample allocation weights on out-of-sample data:

*Table 9: Phase 2 (out-of-sample) results.*

Highest Adjusted Sharpe Ratio – average Winning Score Ratio			
Asset classes	Markov 2S	Proposed model 2S	Markov 3S 2S
<b>Equities</b>	0.03	<b>0.33</b>	0.03
<b>Commodities</b>	<b>0.25</b>	0.13	0.24
<b>FX</b>	0.09	0.15	<b>0.29</b>
<b>Fixed Income</b>	0.06	<b>0.22</b>	0.01
Highest Weighted Annualised Returns – average Winning Score Ratio			
Asset classes	Markov 2S	Proposed model 2S	Markov 3S 2S
<b>Equities</b>	0.08	<b>0.10</b>	0.03
<b>Commodities</b>	0.02	<b>0.27</b>	0.12
<b>FX</b>	<b>0.22</b>	0.19	0.02
<b>Fixed Income</b>	<b>0.25</b>	0.18	0.00
Combined Winning Score Ratio			
Asset classes	Markov 2S	Proposed model 2S	Markov 3S 2S
<b>Equities</b>	0.11	<b>0.43</b>	0.06
<b>Commodities</b>	0.27	<b>0.40</b>	0.35
<b>FX</b>	0.31	<b>0.35</b>	0.31
<b>Fixed Income</b>	0.31	<b>0.40</b>	0.01

It is reassuring that the out-of-sample performance resemble the in-sample results, such that the highest ASR does not indicate a clear winner, the highest annualised returns slightly clarify the situation by isolating the proposed and two-state MSR models, and the combined score section points at the ultimate winner.

In the highest adjusted Sharpe Ratio, the KAMA+MSR model evidently performs best in equities and fixed income. The two-state Markov-switching regression marginally outperforms the three-state MSR turned into two states in commodities, whereas the latter model achieves the best results in FX.

In the highest annualised returns, the KAMA+MSR model again outperforms in equities, however, not considerably, particularly versus the two-state Markov regression. The proposed regime-switching model does clearly better in commodities, yet marginally underperforms in FX and fixed income the two-state MSR.

Finally, after combining the winning scores (given equal weights), the KAMA+MSR model demonstrates the highest outcome, specifically within equities and fixed income. In commodities and foreign exchange, the results are not that distinct, as the proposed model only slightly outperforms the other models.

## 3.4 Discussion

### 3.4.1 Comparison of performance across different asset classes

The out-of-sample analysis (Phase 2) demonstrates the robustness of the proposed model and indicates its high performance, particularly in case of equities.

#### *Equities*

This is the only asset class in which the KAMA+MSR model has outperformed the other models in both ASR and annualised returns. The notably higher adjusted Sharpe ratio for KAMA+MSR points at a good balance between returns and volatility which suggests a solid blend between the use of trend in predicting regime switches within KAMA and the use of variance for this purpose in Markov-switching regression. It is also worth pointing out that the proposed model achieves such results regardless of the equity region. Whether it is the USA, Europe, or Emerging Markets, the KAMA+MSR model seems to be resistant to both global and local shocks (though more indices, in particular from the Emerging Markets region, could strengthen this argument).

### *Commodities*

In this case, the proposed model has been the most profitable of those tested; however, these gains also create significant risk, which the ASR metric demonstrates. Thus, if the trader is risk-averse, it is possibly safer to pick the three-state MSR turned into two states, which generated almost as high ASR as the two-state MSR model and higher annualised returns than that model. However, the proposed model, KAMA+MSR, could add more value in case of singular assets, e.g., gold, sugar, or natural gas (see Appendix A).

### *Foreign exchange*

Analysis here reveals rather mixed results. Even though KAMA+MSR shows the most stable performance—that is, it is the second-best model when considering ASR and annualised returns separately—it ultimately depends on the trader which model is preferred. The risk-averse trader might select the three-state MSR turned into two states, whereas the risk-seeking trader could pick the two-state Markov-switching regression to build a regime-based strategy. Possibly, the mixed performance of all the models for foreign exchange may stem from eliminating the transitory regimes from the non-two-state models; however, the in-sample analysis (Phase 1) does not suggest it, as it also favours the two-state models. Section 3.1.2 on related work mentions the application of Markov-switching regressions to currencies, although the latest paper from (2007) indicates only modest returns increase. Thus, possibly, Markov-switching regressions may not be for FX overall, and other regime-detection methods should be pursued for a more robust performance. Nevertheless, similarly to commodities, KAMA+MSR seems to perform particularly well in cases of specific assets, such as, AUDJPY, USDAUD, USDDKK, EURUSD, and EURRUB currency pairs (see Appendix A).

### *Fixed income*

While KAMA+MSR was not the best for fixed income in Phase 1, in the out-of-sample Phase 2 it is the winner. It has outperformed other models in the ASR metric and has not achieved a significantly worse result for annualised returns. Interestingly, it failed to discern four regimes in Emerging Markets Credit ETF due to a small sample size after extracting 15% for the holdout, and still performed the best on average. In Phase 1, EM Credit was one of the few fixed income assets in which the KAMA+MSR model accomplished the highest winning score ratio (see Appendix A) which could imply similar achievement in the out-of-sample analysis if the sample size was sufficiently larger.

#### **3.4.2 Why are three- or four-state models sub-optimal?**

Even though the four-state KAMA+MSR does not outperform its “competitors”, by identifying the non-persistent regimes it is better able to detect persistent states than the other models, specifically the two-state Markov-switching regression in isolation. As stated earlier, the underperformance of the four-state KAMA+MSR model is understandable; however, this could be possibly changed by applying, for instance, neutral option strategies to the transitory regimes (high variance / bullish, low variance / bearish) instead of simply allocating weights between assets and cash. Nevertheless, without the non-persistent regimes, the proposed model would not be able to detect the persistent regimes (low variance / bullish, high variance / bearish) with such accuracy and smoothness. It is, thus, safe to assume that it is up to the trader what sort of strategy should be imposed on both transitory and persistent regimes to fully capture the regime-detecting capability of the KAMA+MSR model. In addition, and more importantly, the four-state KAMA+MSR model shows promise in terms of generating labels for subsequent ex-ante predictions of all the regimes, not just the two (low variance / bullish, high variance / bearish) over which the model has outperformed its benchmarks.

### 3.5 Conclusions

This chapter presented an enhancement of the two-state Markov-switching regression model with the Kaufman's Adaptive Moving Average (the KAMA+MSR model). The proposed model aims at detecting two transitory (high variance / bullish, low variance / bearish) and two persistent regimes two (low variance / bullish, high variance / bearish), with accurate timing of onset of each state, but without frequent regime switches. Owing to the use of both variance and trend factors to detect regime switches in the KAMA+MSR model it managed to outperform, on average, the comparison models in every analysed asset class. The proposed model has done particularly well in equities, where it achieved the highest average adjusted Sharpe ratio and mean annualised returns after implementing a simple asset allocation strategy on a holdout sample (Phase 2). Even though the transitory regimes were excluded from this trading strategy, their detection proved to be critical for better distinguishing the persistent regimes, in contrast with, for instance, the three-state Markov-switching regression model.

There are a number of ways in which the work of this chapter could be improved in the future. First, even though the proposed model outperformed the “opposing” models in Phase 2, it slightly did so with regards to foreign exchange and commodities. Possibly some other method, as opposed to Markov switching, could do significantly better in combination with KAMA with regards to these asset classes. Second, all models were evaluated using a rather naïve trading strategy which allocated weights to an asset or cash depending on the identified regime; more complex strategies could also be considered. For example, it is possible the power of the detected transitory regimes might be put to good use in derivatives-based hedging such as neutral option strategies. The fact that these states improved the detection of persistent regimes in the KAMA+MSR model may suggest that the transitory regimes have their own “hidden” ability worth investigating. In fact, due to their essential role in the novel framework described in this chapter, the signals generated by both permanent and transitory regimes will be further inspected in Chapter 4 which focuses on predicting distinguished regimes ahead of time.

## **4. Predicting financial regimes by the use of the Random Forest-KAMA+MSR framework**

Chapter 3 focused on applying the KAMA+MSR framework to detect financial regimes within various asset classes. An exchange between an asset and cash (i.e., overweight an asset in low variance / bullish regime, and underweight it in high variance / bearish regime) was applied to compare different regime detection frameworks to study the accuracy of each and, by adjusting the returns with costs, to see how frequent the detection model shifts regimes. It was shown that the regime could be done accurately; however, the potential use of a detection framework in trading may be hindered by the lag of the signal, as well as a lack of clarity as to the optimal strategy to be applied to fully exploit the detected regimes, making detection a sub-optimal basis for a trading strategy.

This chapter will attempt to predict the regimes ex-ante to avoid the above-mentioned lags; additionally, it will look at alternative ways to exploit the generated signals, with the aim of finding a more suitable trading strategy than a conventional asset-cash exchange whenever the regime shifts. Similarly to Chapter 3, this chapter will work with equities, commodities, and foreign exchange rates to study the robustness of the proposed method. The chapter is organised as follows. First, relevant technical background and related work are presented. This is followed by a methodology section, which includes a discussion of the data used in the experiments of this chapter. The chapter ends with presenting and discussing the results of the experiments.

## 4.1 Background and Related Work

### 4.1.1 Fractional differencing

As was argued in Chapter 2, one of the best machine learning algorithms to predict financial regimes is Random Forest, which this chapter will use. As with any machine approach to financial time series prediction, it is better here to operate on a transformed stationary series than the original data; this is because it is often difficult to learn patterns from raw, non-stationary series that is prone to structural breaks, such as sudden changes in trends or shifting variability. The classic statistical method to transform non-stationary series into stationary is to apply differencing or percentage change. However, though it is a common solution, this creates another issue, that of completely erasing the memory the time series holds (Tsay, 2010). In other words, adverse events like dot-com bubble, Great Financial Crash, Euro crisis of 2012, or finally the Covid bear market of 2020, may all be very similar to each other after first-differencing the data, even though, from the macroeconomic perspective, they were completely different events. In addition, they may also be less distinguishable from technical market corrections that have not caused such widespread panic on the capital markets. The major advantage of retaining non-stationary time series is that all these adverse events, whether severe or moderate, can be easily distinguished, owing to the memory of the data. The same logic applies to up trending, bullish markets; the so-called bear rally, as well as the bullish regime when prices keep upward momentum, will look the same when prices are first-differenced, while these regimes significantly differ when the data is non-stationary, the former being a technical rally within detrimental macroeconomic environment used by traders for short-term gains, whereas the latter is an opportunity to increase returns by holding an asset for a longer period due to optimistic views on the economy.

The problem described above is also known as the stationarity vs. memory dilemma (López de Prado, 2018), and can be solved by transforming the time series data using *fractional differencing* (Hosking, 1981). Assuming a non-stationary time

series  $s$  throughout time  $t$ , such that  $s = \{s_t, s_{t-1}, s_{t-2}, \dots, s_{t-k}\}$ , the first-differencing procedure of the preceding paragraph can be computed as

$$\Delta s_t = s_t - s_{t-1}, \quad (24)$$

which transforms the non-stationary time series  $s$  into a stationary one, effectively erasing its memory. If a backshift operator  $B$  is defined, such that

$$B^k s_t = s_{t-k}, \quad (25)$$

where  $k \geq 0$  and  $t > 1$ , Equation 24 (*first-order differencing*) can be redefined as

$$\Delta s_t = s_t - s_{t-1} = s_t - Bs_t = (1 - B)s_t. \quad (26)$$

Situations may occur in which the first-differenced data, after applying Equation 26, may even so not become stationary; such time series would require *second-order differencing* to achieve the desired effect, as following,

$$\Delta^2 s_t = \nabla(s_t - s_{t-1}) = (s_t - s_{t-1}) - (s_{t-1} - s_{t-2}) = s_t - 2s_{t-1} + s_{t-2}, \quad (27)$$

which, using the backshift operator  $B$ , can be expressed as

$$(1 - B)^2 s_t = s_t - 2s_{t-1} + s_{t-2}, \quad (28)$$

since

$$(1 - B)^2 = 1 - 2B + B^2 \quad (29)$$

and

$$B^2 s_t = s_{t-2}. \quad (30)$$

More generally, assuming the order of differentiation is  $d$ , Equation 28 can be represented as

$$\Delta^d s_t = (1 - B)^d s_t. \quad (31)$$

Using a binomial equation

$$(1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k, \quad (32)$$

where  $d$  can be any real number, Equation 29 (and thus Equation 28) can be expanded to

$$(1 - B)^d = \sum_{k=0}^{\infty} \frac{\prod_{i=0}^{k-1} (d - i)}{k!} (-B)^k = \sum_{k=0}^{\infty} (-B)^k \prod_{i=0}^{k-1} \frac{d - i}{k - i}, \quad (33)$$

which is effectively equal to

$$1 - dB + \frac{d(d-1)}{2!} B^2 - \frac{d(d-1)(d-2)}{3!} B^3 + \dots. \quad (34)$$

To see how the order of differentiation  $d$  preserves memory in the transformed time series, first some further calculations must be performed. Assuming that the current value  $s_t$  in the time series is a function of all the past values occurring before time  $t$ , a weight parameter,  $\omega_k$ , can be assigned to each past datapoint, such that

$$s_t = \sum_{k=0}^{\infty} \omega_k s_{t-k}. \quad (35)$$

Applying fractional differencing to the time series  $s$  allows the finding of weights  $\omega_k$  for each past value  $s_{t-k}$ , such that

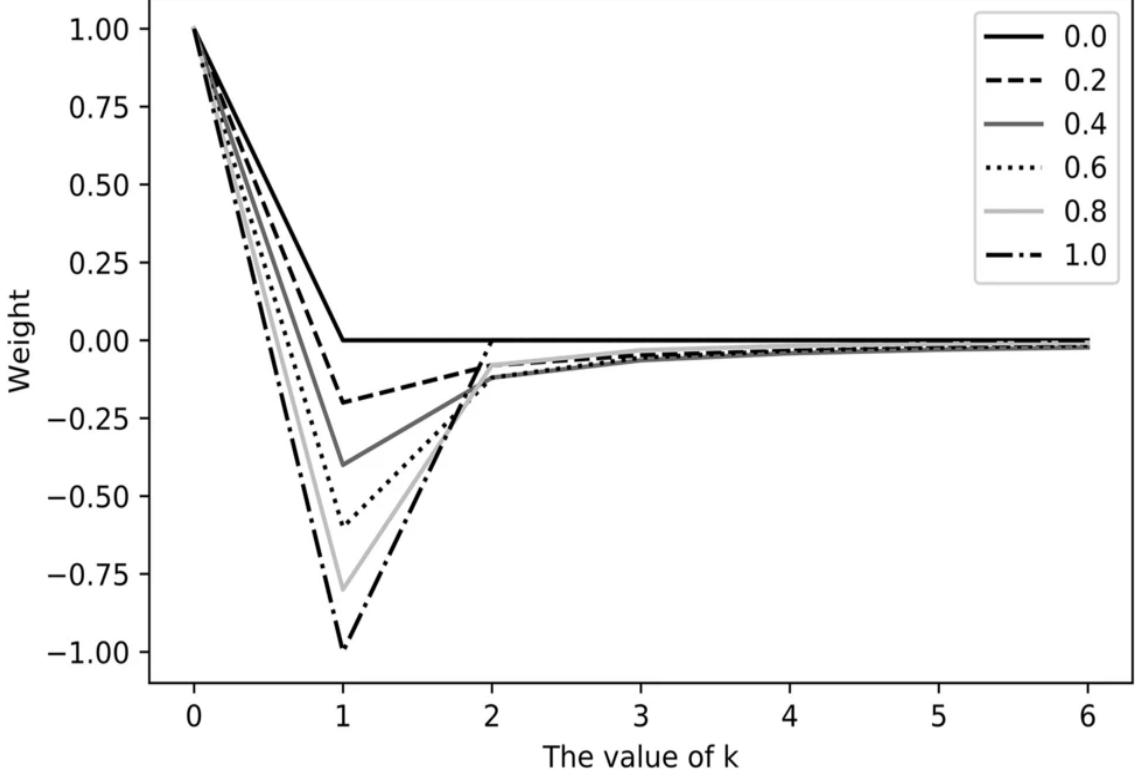
$$\omega = \left\{ 1, -d, \frac{d(d-1)}{2!}, -\frac{d(d-1)(d-2)}{3!}, \dots, (-1)^k \prod_{i=0}^{k-1} \frac{d - i}{k - i} \right\}. \quad (36)$$

Assuming the order of differentiation  $d \in \mathbb{R}^+$ , there is a moment when  $d$  will be equal to  $k$ , so that  $d - k = 0$ , as well as

$$\prod_{i=0}^{k-1} \frac{d - i}{k!} = 0, \quad (37)$$

which effectively means that memory of time series  $s$  beyond that point is removed. If  $d = 1$ , the set of weights  $\omega$  from Equation 36 will become  $\omega = \{1, -1, 0, 0, \dots, 0\}$ . Figure 26 on the following page depicts this logic.

Figure 26: Changing weights  $\omega_k$  on y-axis, as the values of  $k$  on x-axis increase. Each line is associated with a particular value of the order of differentiation  $d \in [0,1]$  in 0.1 increments. Source: (Walasek & Gajda, 2021).



It can be seen in Figure 26 that if  $0 < d < 1$ , and  $k \in \mathbb{Z}^+$ , all weights  $\omega_k$  are non-zero; thus the memory in the transformed time series has been preserved, i.e., all weighted past values  $s_{t-k}$  contribute to the current value  $s_t$ , while the time series  $s$  transitions from non-stationary to stationary at the same time.

For the above derivation, the weights  $\omega_k$  can be generated iteratively as

$$\omega_k = -\omega_{k-1} \frac{(d - k + 1)}{k}. \quad (38)$$

It has been previously stated that for  $d = 1$  all weights in  $\omega$  become 0, except from  $\omega_0$  and  $\omega_1$ , which become 1 and -1, respectively. This is a classic first-order differencing, when the time series is stationary. Similarly, for  $d = 0$ , only  $\omega_0 = 1$ , while the rest of the weights become 0. Such a series remains untransformed, thus coinciding with the original time series. Hence, in order to solve the previously mentioned stationarity vs. memory dilemma, the optimal order of differentiation  $d$  must be found, such that the transformed time series  $y$  preserves memory, while becoming stationary (López de Prado, 2018), (Walasek & Gajda, 2021). Section 4.2.2

on methodology will describe the application of fractional differencing to the data used in this chapter, including finding the optimal value of  $d$  for each feature used to predict the target variable.

Applying fractional differencing to financial time series has become particularly popular after its demonstrated success in (López de Prado, 2018). More recently, fractional differencing has been applied to various equity indices in order to show that even though the linear correlation coefficients between transformed and raw time series are high (above 0.99%), the *Kwiatkowski–Phillips–Schmidt–Shin* (KPSS) test<sup>36</sup> for stationarity, as well as the augmented Dickey–Fuller test<sup>37</sup> for unit root, prove stationarity of the fractional series. In addition, transforming time series by fractional differencing has improved the out-of-sample accuracy of a neural network algorithm predicting the asset's price one step ahead (Walasek & Gajda, 2021).

In addition to its use of fractional differencing, this chapter will also make use of feature selection; the section below describes in detail techniques that will be used for this process.

#### 4.1.2 Feature selection

As will be described in more detail in the methodology section (4.2.2), the predictive framework used in this chapter incorporates feature selection to improve the accuracy of Random Forest on the holdout set by reducing the dimensionality of the data. Even though Random Forest applies its own importance score to each feature utilised in the training, such a solution has been found to be suboptimal, as will be explained below.

Section 2.2.4 on Random Forest described methods to calculate the probability of misclassifying a predicted variable, such as Gini impurity. The same metric can be used to inspect the importance of each feature; the *mean decrease in*

---

<sup>36</sup> Refer to (Kwiatkowski, Phillips, Schmidt, & Shin, 1992) for technical details on the KPSS test.

<sup>37</sup> Refer to (Cheung & Lai, 1995) for technical details on the augmented Dickey–Fuller test.

*impurity* (MDI) calculates each feature's importance as the sum over the number of splits (across all trees) that include the feature, proportionally to the number of samples it splits. In other words, MDI measures how effective the feature is at reducing uncertainty in a classifier when creating decision trees within Random Forest (Breiman, 2001). Even though the creators of Random Forest, Leo Breiman and Adele Cutler, acknowledged that this mechanism helps identify the most important features, as well as being often robust to feature permutation<sup>38</sup>, more recently it has been found that this is not always the case. For instance, the importance score built into Random Forest tends to inflate the importance of continuous or high-cardinality categorical variables (Strobl, Boulesteix, Zeileis, & Hothorn, 2007). In addition, MDI in Random Forest is calculated using the fitted model, without considering the importance of each feature on the validation set; it thus focuses on the fit rather than on the predictive power of the predictors. Finally, relying on classification accuracy to categorise whether a feature is important or not has been found to be flawed, in that such a method is sufficient to indicate which feature is important; however, it is not sufficient to indicate which feature is unimportant (Kursa & Rudnicki, 2010).

The above issues can be solved by investigating feature importance using a recently popular feature selection package called BorutaShap<sup>39</sup>. BorutaShap is an extension of another popular package called Boruta (Kursa & Rudnicki, 2010), which is a wrapper algorithm that uses Random Forest to determine the most critical and most redundant features in terms of their predictive power. The main difference between BorutaShap and Boruta is that the former allows the wrapping of the algorithm around other machine learning algorithms than Random Forest (such as XGBoost), as well as using *Shapley values* to determine the average importance of each feature instead of MDI. Before describing BorutaShap in more detail, it is essential to understand the use of Shapley values in investigating feature importance, a method known as *SHapley Additive exPlanations* (SHAP) (Lundberg &

---

<sup>38</sup> Quoting the official repository: "Adding up the gini decreases for each individual variable over all trees in the forest gives a fast variable importance that is often very consistent with the permutation importance measure." Source:

[https://www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm#giniimp](https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#giniimp). Accessed on 28/07/2022.

<sup>39</sup> See <https://github.com/Ekeany/Boruta-Shap> for the official repository of BorutaShap. Accessed on 28/07/2022.

Lee, A unified approach to interpreting model predictions., 2017), which is briefly outlined below.

### *Shapley values and SHAP*

SHAP is a method to explain individual predictions by computing the contribution of each feature to the prediction. To achieve this, SHAP exploits Shapley values, a game theory concept coined by (Shapley, 1953), which is based on a fair attribution of a player's contribution to the result of a game. If a set of players cooperate on some task, they will work in a coalition to maximise their gains. Shapley values measure the total pay-off by computing the marginal contribution of each player to the result. This idea has been linked to machine learning and feature importance, in which context the game is represented by the prediction of a single instance  $x_i$ , and the players are the collaborating features used for a prediction of the instance  $x_i$ . Formally, the contribution  $\phi$  of a feature  $j$  to the prediction of instance  $x_i$  (Shapley value) can be expressed as

$$\phi_j(v) = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|! (M - |S| - 1)!}{M!} [v(S \cup \{j\}) - v(S)], \quad (39)$$

where  $v$  is some value function (such as a machine learning algorithm) that takes the subset (or coalition) of features  $S$  and returns the total pay-off of the game (i.e., predictions),  $N$  is the set of all input features,  $M$  is the number of input features, and  $\phi_j \in \mathbb{R}$ . Thus, in Equation 39, the Shapley value is calculated by computing a weighted average pay-off gain that the feature  $j$  provides when included in all coalitions that exclude the feature  $j$ . In practice, Shapley values are estimated rather than calculated directly, as the more the features used, the larger the number of possible coalitions, which grow exponentially<sup>40</sup>.

The SHAP method takes Shapley values further by representing them as an additive feature attribution method, a linear model  $g$ , that can be defined as

---

<sup>40</sup> See (Štrumbelj & Kononenko, 2014) for an example of approximating Shapley values with the use of Monte-Carlo sampling.

$$g(z') = \phi_0 + \sum_{j=1}^M \phi_j z'_j, \quad (40)$$

where  $z' \in \{0,1\}^M$  is a binary feature vector which represents a feature either observed ( $z'_j = 1$ ) or absent ( $z'_j = 0$ ), and  $\phi_j$  is the Shapley value of a feature  $j$  calculated in Equation 39. The additive feature attribution method  $g$  provides a unique solution that satisfies three desirable properties of a feature explanation technique: local accuracy, missingness, and consistency (Lundberg & Lee, A unified approach to interpreting model predictions., 2017):

- *Local accuracy* means that the sum of the feature attributions is equal to the prediction of the instance  $x_i$  that needs to be explained. Given this single instance  $x_i$  and a feature vector  $o'$  which is a subset of  $z'$  that only includes observed features (i.e.,  $z'_j = 1$ ), the local accuracy property can be expressed as

$$\hat{v} = g(o') = \phi_0 + \sum_{j=1}^M \phi_j o'_j = \mathbb{E}(\hat{v}) + \sum_{j=1}^M \phi_j, \quad (41)$$

where  $\hat{v}$  is the estimated function  $v$  given in Equation 39.

- *Missingness* means that features that are already missing (i.e.,  $z'_j = 0$ ) are attributed no importance. This property forces those missing features to get a Shapley value of 0 instead of some arbitrary number, and can be represented as

$$a'_j = 0 \Rightarrow \phi_j = 0, \quad (42)$$

where  $a'_j$  are features from set  $z'$  that are all absent.

- *Consistency* means that if a model changes so that the marginal contribution of a feature value increases or stays the same (regardless of other features), the Shapley value also increases or stays the same<sup>41</sup>.

On the top of representing Shapley values as an additive feature attribution method, SHAP proposes two additional advantages: (1) a fast Shapley value estimation technique called KernelSHAP and an alternative, more suited to tree-based algorithms, called TreeSHAP (Lundberg, Erion, & Lee, 2018); (2) the

---

<sup>41</sup> Refer to the original paper on SHAP (Lundberg & Lee, A unified approach to interpreting model predictions., 2017) for technical derivation and proof of consistency.

possibility to use global, instead of local (i.e., single instance  $x_i$ ), interpretation methods based on aggregations of Shapley values. These described advantages have made SHAP a popular way to discover feature importance, both in academia and industry, particularly by use of the Python SHAP<sup>42</sup> package. For instance, SHAP has been found to better recover influential features than the original feature importance methods embedded within Random Forest or XGBoost, which has led to more accurate classification models for financial time series, such as discrepancies within expert commentaries (El Mokhtari, Higdon, & Başar, 2019), as well as equity (Xiaomao, Xudong, & Yuanfang, 2019), (Man & Chan, 2021) and gold prices (Jabeur, Mefteh-Wali, & Viviani, 2021).

The feature selection package mentioned previously, BorutaShap, utilises SHAP within its framework to generate a list of features with the highest predictive power. The paragraphs below briefly describe this process.

### *BorutaShap package*

As already mentioned, BorutaShap is a feature selection algorithm that can be wrapped around a machine learning algorithm of choice, such as Random Forest, to determine the most critical features in terms of their predictive power. The package achieves this via the following steps:

1. A new dataset is created which is a copy of the original feature set  $F_s$  but randomly shuffled to remove their correlations with the response (original) feature; these permuted features are called *shadow features*.
2. The shadow features from the previous step are merged with the original dataset  $F_s$  in order to create a two times larger (column-wise) data table  $LF_s$  for the next step.
3. The  $LF_s$  dataset is split into a training set  $LF_t$  and a validation set  $LF_v$ .
4. A machine learning algorithm of choice, such as Random Forest, is fit to the training  $LF_t$  set and predicts the  $LF_v$  set. Subsequently, the SHAP algorithm is used to calculate aggregated Shapley values of each feature from the  $LF_v$  set.

---

<sup>42</sup> The official repository of the SHAP package can be found here: <https://github.com/slundberg/shap>. Accessed on 28/07/2022.

5. The computed Shapley values are z-scored<sup>43</sup>; the shadow feature with the largest z-score becomes a threshold above which a feature is deemed a “hit”. This procedure stems from the notion that a feature is only useful if it has a higher predictive power than the best randomised feature.
6. The issue with the process so far is that the features deemed as “hits” may end up so due to a pure chance (as step 1 incorporates random permutations which may influence the outcome of step 5). Therefore, BorutaShap repeats steps 1-5  $n$  times to make the feature selection process more reliable.
7. After completing all  $n$  trials, the number of hits is summed for each feature. However, this still does not clarify whether a feature is useful or not; for instance, what if a feature  $j$  has been deemed important in 13 trials out of 20 (has 13 hits) but unimportant in seven? This is where BorutaShap implements a binomial test<sup>44</sup> to check the probability of accepting or rejecting the feature after all  $n$  trials. The binomial test is sufficient, since each trial can give only a binary outcome (a hit or not); thus, a series of  $n$  iterations follows a binomial distribution.
8. The binomial test may result in a feature being tentative, such that even after all  $n$  trials it remains uncertain whether the feature can be accepted or rejected. This is usually the case when out of, for instance, 20 trials, the feature would have around ten hits. If such an event occurs, BorutaShap leaves a decision to the user whether the feature should be accepted or rejected. To help with the decision, BorutaShap optionally checks the median z-score of the feature and compares it to the median threshold (a maximum z-score of the shadow features); if the former is the feature is accepted, otherwise it is rejected.

Boruta – in combination with Random Forest – has been proved successful in reducing noisy and highly correlated variables in various studies, such as those involving remote sensing (Leutner, et al., 2012), gene selection (Deng & Runger, 2013), or domestic energy use (Candanedo, Feldheim, & Deramaix, 2017), as well as predicting bitcoin prices (McNally, Roche, & Caton, 2018), stock returns (Naik &

---

<sup>43</sup> Z-score (or a standard score) is calculated as  $\frac{x_i - \mu_s}{\sigma_s}$ , where  $x_i$  is the observed value,  $\mu_s$  is the mean of the sample, and  $\sigma_s$  is the standard deviation of the sample.

<sup>44</sup> For testing the hypothesis of the binomial test, BorutaShap assumes p-value of 0.05.

Mohan, 2019), or VIX (volatility index mentioned in Section 2.1.5 on regime switches) (Kim & Han, 2022). BorutaShap, which builds upon Boruta (which being newer<sup>45</sup> than its predecessor Boruta has not been so used so widely). However, it has been recently applied, for example, to predicting the outcomes of football matches (Geurkink, Boone, Verstockt, & Bourgois, 2021).

#### 4.1.3 Candidate features

Before applying a method such as BorutaShap it is of course necessary to decide on an initial pool of features from which the most useful features can then be selected. From the relevant literature there are two groups of features that can be used to predict the price movements of any asset. The first major group are technical indicators, which have already been argued in Section 2.3.4 to add value in building portfolios, as well as improve the predictability of asset returns. The second group are fundamental and macroeconomic features, though it should be noted that the fundamental features are mostly associated with equities.

A composite review (Gu, Kelly, & Xiu, 2020) of over 900 predictors of equity risk premia in combination with various machine learning algorithms, including Random Forest, has listed *price/earnings* (PE), *price/book* (PB), *price/sales* (PS), and *dividend yield* (DY), as the most important predictors of equity prices, on the top of technical indicators that can be divided into four groups, relating to momentum, volatility, volume, and liquidity. In terms of macroeconomic features helpful in predicting equity price movements, the literature mentions FX rates (e.g., (Hau & Rey, 2006), (Katechos, 2011)), credit spreads (e.g., (Gilchrist & Zakrajšek, 2012), (Friewald, Wagner, & Zechner, 2014)), bond yields, interest rates, inflation, economic distress (proxied by LIBOR-OIS spread, TED spread, and yield curve (e.g., (Fama & French, 1989), (Welch & Goyal, 2008)) (with Table 11, which can be found in Section 4.2.1, providing descriptions of these variables), and commodity prices (e.g., (Lombardi & Ravazzolo, 2016)).

---

<sup>45</sup> BorutaShap was released in 2020, whereas Boruta was released in 2010.

Technical indicators have also been found to have predictive power for commodity prices, as has been shown for the example of oil prices (Yin & Yang, 2016). Macroeconomic variables can also predict commodity returns with success, using similar predictors as for equities, such as volatility, bond yields, interest rates, foreign exchange rates, and yield curve (e.g., (Frankel, 2008), (Chen, Rogoff, & Rossi, 2010), (Dbouk & Jamali, 2018)); however, other predictors, such as trade-weighted US dollar, *Baltic Dry Index* (BDI), as well as a composite index of multiple commodity prices, *Commodity Research Bureau Index* (CRBI), can also be helpful in predicting commodity prices (Dbouk & Jamali, 2018) (Table 11 in the following section will describe these variables in more detail).

Finally, with regards to FX rates, a recent comprehensive study (Djemo, Eita, & Mwamba, 2021), employing Random Forest, named commodity prices, interest rates and interest rates differentials, inflation and inflation differentials, equity prices, forward exchange rate points, and terms of trade as the most important predictors of various foreign exchange rates, which is also in line with conventional beliefs. In addition, another study (Han, Wan, & Xu, 2020) named the Baltic Dry Index, a useful predictor of commodity prices, to also have significant predictive power in relation to FX rates.

As will be seen in the following data and methodology section, the features described above have been used to conduct the main experiment of this chapter, which is to predict financial regimes *ex ante*; the sub-section relating to data will outline the predictors used in more detail.

## 4.2 Data and methodology

### 4.2.1 Data

As was previously mentioned, Chapter 4 will focus on predicting regimes for three major asset classes: equities, commodities, and foreign exchange pairs. Table 10 lists the classes and assets that are their subcomponents, as well as the final start date of the data completing the data preparation step described in detail in the next section (4.2.2). Note that each asset class has a separate benchmark used only in out of sample model performance comparison, a process also outlined in the next section, and that all series, including the benchmarks, end on the same date, 29/04/2022.

*Table 10: List of asset classes, assets, and start dates. All data series terminate on 29/04/2022. The main benchmark for each asset class is given at the end of the list of assets for that class and indicated in bold. (BBG = Bloomberg.)*

Asset class	Asset	Asset label	Start date
<b>Equities</b>	S&P/ASX 200 ETF	ASX	03/05/2001
	CAC40 ETF	CAC	29/01/1996
	DAX ETF	DAX	29/01/1996
	FTSE 100 ETF	FTSE	03/04/2002
	FTSE MIB ETF	FTSEMIB	06/09/2004
	KOSPI ETF	KOSPI	24/06/1995
	MSCI China ETF	MSCI_China	25/12/2002
	NASDAQ 100 ETF	NASDAQ	31/01/2001
	NIFTY 50 ETF	NIFTY	02/02/2001
	Nikkei 225 ETF	Nikkei	28/02/1995
	Swiss Market ETF	SMI	31/01/1996
	S&P 500 ETF	SPX	30/01/1995
	S&P/TSX Composite ETF	TSX	28/02/1995
	TWSE ETF	TWSE	30/12/1996
	<b>ACWI Index</b>	<b>ACWI</b>	<b>06/03/2018</b>
<b>Commodities</b>	Aluminium Futures	Aluminium	12/08/2004
	Brent Crude Oil Futures	Brent	12/03/1993
	Coffee Futures	Coffee	21/04/1993
	Copper Futures	Copper	07/07/1993
	Corn Futures	Corn	23/08/1995
	Gold Futures	Gold	23/11/1993
	Live Cattle Futures	Live_cattle	01/10/1992

	Natural Gas Futures	<i>Natural_gas</i>	07/05/1993
	Nickel Futures	<i>Nickel</i>	10/01/2005
	Soybeans Futures	<i>Soybeans</i>	17/10/1997
	Sugar Futures	<i>Sugar</i>	17/02/1992
	Wheat Futures	<i>Wheat</i>	18/01/1996
	<b>BBG Commodity Index</b>	<b><i>BBG_Commodity</i></b>	<b>19/12/2017</b>
<b>Foreign Exchange</b>	AUDJPY Spot	<i>AUDJPY</i>	01/05/2000
	USDCHF Spot	<i>USDCHF</i>	27/05/1999
	USDEUR Spot	<i>USDEUR</i>	18/01/2000
	USDGBP Spot	<i>USDGBP</i>	13/02/1995
	USDAUD Spot	<i>USDAUD</i>	01/05/2000
	USDCAD Spot	<i>USDCAD</i>	13/02/1995
	USDDKK Spot	<i>USDDKK</i>	13/02/1995
	USDJPY Spot	<i>USDJPY</i>	13/02/1995
	USDNOK Spot	<i>USDNOK</i>	30/01/1996
	USDNZD Spot	<i>USDNZD</i>	03/09/1999
	USDSEK Spot	<i>USDSEK</i>	13/02/1995
	<b>BBG DXY Index</b>	<b><i>BBDXY</i></b>	<b>20-04-2018</b>

The starting dates of each asset within a single class differ due to either index or feature data availability. It was decided to use only a single benchmark in each asset class, but one that had a meaningful relation to the assets used for training (however, noting that the time period selected for out-of-sample testing will be different to that used for training). First, all equities from Table 10 are top members of the ACWI Index, excluding NASDAQ which has been selected as a proxy of growth. Similarly, the majority of commodities listed are used to build the Bloomberg Commodity Index, an important benchmark for industry experts to keep track of commodity prices worldwide. Finally, since all foreign exchange pairs (excluding AUDJPY<sup>46</sup>) use the US Dollar as a base currency, the Bloomberg DXY Index, which indicates the overall strength of the US Dollar against leading global currencies<sup>47</sup>, has been used as a benchmark.

The data used for the experiment conducted in this chapter contain features which are relevant to every asset class, as well as features relevant only for a certain

---

<sup>46</sup> AUDJPY currency pair is usually used in the industry as a contrarian indicator of a status of the global markets, where the elevated values of AUDJPY may indicate contracting markets to come, while low values are usually associated with the market rally to begin shortly.

<sup>47</sup> Here by leading global currencies the G10 countries are meant.

class, a topic already discussed in the previous section (4.1.3). Table 11 lists those variables used by all asset classes, which are further split into technical and macro features. The technical features were computed using the TA<sup>48</sup> package, a popular technical analysis tool coded in Python, as well as the tsfresh<sup>49</sup> package (Christ, Kempa-Liehr, & Feindt, 2016). It must be noted that technical features sourced from the tsfresh package were computed using a rolling window of maximum 252 days (a trading year), while the features calculated by the TA package use various rolling windows, indicated in the "formula and/or description" column of Table 11. In addition, all features are at a daily frequency (excluding weekends) with the exception of inflation rate and terms of trade data, which are originally obtained at a monthly frequency, but later resampled to daily by forward-filling the latest value over the given month. Finally, as there is no volume data for foreign exchange pairs, features that include volume do not apply to this particular asset class.

*Table 11: Features used in the prediction of each asset class (excluding volume-related features for FX).*

Feature	Feature label	Formula and/or description
<b>Technical features (source: TA package)</b>		
<b>Closing price</b>	<i>Close</i>	Asset's closing price $C$ of the current day.
<b>High price</b>	<i>High</i>	Asset's highest price $H$ of the current day.
<b>Low price</b>	<i>Low</i>	Asset's lowest price $L$ of the current day.
<b>Volume</b>	<i>Volume</i>	Asset's number of shares traded $V$ over the current day.
<b>Relative Strength Index</b>	<i>RSI</i>	$n = 14, C = \text{closing price},$ $\bar{C}_g = \text{average daily positive change in } C \text{ over } n,$ $\bar{C}_l = \text{average daily negative change in } C \text{ over } n,$ $RS = \frac{\bar{C}_g}{\bar{C}_l}, RSI = 100 - \frac{100}{(1+RS)}.$
<b>Ultimate Oscillator</b>	<i>Ultimate</i>	$L = \text{day's lowest price}, H = \text{day's highest price},$ $C = \text{closing price}, C_p = \text{prior closing price},$ $n_1 = 7, n_2 = 14, n_3 = 28,$ $BP = C - \min(L, C_p), TR = \max(H, C_p) - \min(L, C_p),$ $A_{n_1} = \frac{\sum_{t=1}^{n_1} BP}{\sum_{t=1}^{n_1} TR}, A_{n_2} = \frac{\sum_{t=1}^{n_2} BP}{\sum_{t=1}^{n_2} TR}, A_{n_3} = \frac{\sum_{t=1}^{n_3} BP}{\sum_{t=1}^{n_3} TR},$ $Ultimate = \left[ \frac{(A_{n_1} \times 4) + (A_{n_2} \times 2) + A_{n_3}}{4 + 2 + 1} \right].$

<sup>48</sup> See the official Github repository for TA: <https://github.com/bukosabino/ta>. Accessed on 03/08/2022.

<sup>49</sup> The official repository for tsfresh is available here: <https://github.com/blue-yonder/tsfresh>. Accessed on 03/08/2022.

<b>Awesome Oscillator</b>	<i>Awesome</i>	$L = \text{day's lowest price}, H = \text{day's highest price}$ , $M = (L + H)/2, n_1 = 5, n_2 = 34,$ $\overline{M}_n = \text{average } M \text{ over period of } n \text{ days},$ $\text{Awesome} = \overline{M}_{n_1} - \overline{M}_{n_2}.$
<b>Rate of change</b>	$ROC_{1m},$ $ROC_{6m},$ $ROC_{12m}$	$C = \text{closing price}, C_{p-n} = \text{prior closing price } n \text{ days ago},$ $n_1 = 21, n_2 = 120, n_3 = 252,$ $ROC_{1m} = \frac{C}{C_{p-n_1}} - 1, ROC_{6m} = \frac{C}{C_{p-n_2}} - 1, ROC_{12m} = \frac{C}{C_{p-n_3}} - 1.$
<b>Maximum return</b>	$Max\_Rets_{1m}$	$C = \text{closing price}, C_p = \text{prior closing price}, n = 21,$ $Max_{Rets_{1m}} = \max \left( \frac{C}{C_p} - 1 \right)_n.$
<b>Momentum change</b>	$Chg_{6m}$	$p = \text{prior value}, Chg_{6m} = ROC_{6m} - ROC_{6m_p}$
<b>Chande Momentum Oscillator</b>	$CMO$	$C_H = \text{closing price higher than the prior price},$ $C_L = \text{closing price lower than the prior price}, n = 21,$ $CMO = \frac{\sum_{t=1}^n C_H - \sum_{t=1}^n C_L}{\sum_{t=1}^n C_H + \sum_{t=1}^n C_L} \times 100.$
<b>Triple Exponential Average</b>	$TRIX$	$C = \text{closing price}, p = \text{prior value}, n = 21,$ $EMA_1 = \left[ C \times \left( \frac{2}{1+n} \right) \right] + EMA_{1p} \times \left[ 1 - \left( \frac{2}{1+n} \right) \right],$ $EMA_2 = \left[ EMA_1 \times \left( \frac{2}{1+n} \right) \right] + EMA_{2p} \times \left[ 1 - \left( \frac{2}{1+n} \right) \right],$ $EMA_3 = \left[ EMA_2 \times \left( \frac{2}{1+n} \right) \right] + EMA_{3p} \times \left[ 1 - \left( \frac{2}{1+n} \right) \right],$ $TRIX = \frac{(EMA_3 - EMA_{3p})}{EMA_{3p}}.$
<b>Average True Range</b>	$ATR$	$L = \text{day's lowest price}, H = \text{day's highest price},$ $C_p = \text{prior closing price}, p = \text{prior value}, n = 14,$ $TR = \max (H - L,  H - C_p ,  L - C_p ),$ $ATR = \frac{[ATR_p \times (n-1) + TR]}{n}.$
<b>Positive Directional Indicator<sup>50</sup></b>	$Plus\_DI$	$H = \text{day's highest price}, p = \text{prior value}, n = 21,$ $DM = H - H_p,$ $DM_s = \left( \sum_{t=1}^n DM \right) - \left( \frac{\sum_{t=1}^n DM}{n} \right) + DM,$ $Plus_{DI} = \left( \frac{DM_s}{ATR_n} \right) \times 100.$
<b>Negative Directional Indicator<sup>51</sup></b>	$Minus\_DI$	$L = \text{day's highest price}, p = \text{prior value}, n = 21,$ $DM = L_p - L,$ $DM_s = \left( \sum_{t=1}^n DM \right) - \left( \frac{\sum_{t=1}^n DM}{n} \right) + DM,$

<sup>50</sup> Positive Directional Indicator uses Average True Range (ATR) over the same period in its computations.

<sup>51</sup> Negative Directional Indicator uses Average True Range (ATR) over the same period in its computations.

		$Minus_{DI} = \left( \frac{DM_s}{ATR_n} \right) \times 100.$
<b>Bollinger Bands</b>	<i>BB</i>	$L = \text{day's lowest price}, H = \text{day's highest price},$ $C = \text{closing price}, n = 20, TP = \frac{H+L+C}{3},$ $\sigma_n = \text{standard deviation of } TP \text{ over } n \text{ days},$ $\mu_n = \text{average of } TP \text{ over } n \text{ days},$ $UB = \mu_n + 2 \times \sigma_n, LB = \mu_n - 2 \times \sigma_n, BB = \frac{C-LB}{UB-LB}.$
<b>Ulcer Index</b>	<i>Ulcer</i>	$H = \text{day's highest price}, C = \text{closing price}, n = 21,$ $Max_H = \max(H) \text{ over period of } n \text{ days},$ $PD = \frac{C-Max_H}{Max_H} \times 100, PD_s = \frac{\sum_{t=1}^n PD^2}{n}, Ulcer = \sqrt{PD_s}.$
<b>Mean price volatility</b>	<i>Mean_Std_M</i> <i>Mean_Std_Y</i>	$C = \text{closing price}, n_1 = 21, n_2 = 252, p = \text{prior value},$ $L = \text{day's lowest price}, H = \text{day's highest price},$ $TP = \frac{H+L+C}{3}, \sigma_n = \text{standard deviation of } TP \text{ over } n \text{ days},$ $EMA_{\sigma_n} = \left[ \sigma_n \times \left( \frac{2}{1+n} \right) \right] + EMA_{\sigma_{np}} \times \left[ 1 - \left( \frac{2}{1+n} \right) \right],$ $Mean_{Std_M} = EMA_{\sigma_{n_1}}, Mean_{Std_Y} = EMA_{\sigma_{n_2}}.$
<b>On-Balance Volume</b>	<i>OBV</i>	$C = \text{closing price}, p = \text{prior value}, V = \text{current volume},$ $OBV = OBV_p + \begin{cases} V, & \text{if } C > C_p \\ 0, & \text{if } C = C_p \\ -V, & \text{if } C < C_p \end{cases}$
<b>Force Index</b>	<i>Force_Inx</i>	$C = \text{closing price}, p = \text{prior value}, V = \text{current volume},$ $FI = (C - C_p) \times V.$
<b>Money Flow Index</b>	<i>MFI</i>	$L = \text{day's lowest price}, H = \text{day's highest price},$ $C = \text{closing price}, V = \text{current volume}, n = 21,$ $TP = \frac{H+L+C}{3}, RMF = TP \times V,$ $RMF_{pos} = \text{positive values of } RMF,$ $RMF_{neg} = \text{negative values of } RMF,$ $MFR = \frac{\sum_{t=1}^n RMF_{pos}}{\sum_{t=1}^n RMF_{neg}}, MFI = 100 - \frac{100}{1 + MFR}.$
<b>Moving Average Convergence Divergence</b>	<i>MACD</i>	$C = \text{closing price}, n_1 = 12, n_2 = 26, n_3 = 9, p = \text{prior value},$ $EMA_1 = \left[ C \times \left( \frac{2}{1+n_1} \right) \right] + EMA_{1p} \times \left[ 1 - \left( \frac{2}{1+n_1} \right) \right],$ $EMA_2 = \left[ C \times \left( \frac{2}{1+n_2} \right) \right] + EMA_{2p} \times \left[ 1 - \left( \frac{2}{1+n_2} \right) \right],$ $EMA_{diff} = EMA_1 - EMA_2,$ $EMA_3 = \left[ EMA_{diff} \times \left( \frac{2}{1+n_3} \right) \right] + EMA_{3p} \times \left[ 1 - \left( \frac{2}{1+n_3} \right) \right],$ $MACD = EMA_{diff} - EMA_3.$
<b>Mass Index</b>	<i>Mass_Inx</i>	$L = \text{day's lowest price}, H = \text{day's highest price},$ $n_1 = 9, n_2 = 25, p = \text{prior value}, P_{diff} = H - L,$ $EMA_1 = \left[ P_{diff} \times \left( \frac{2}{1+n_1} \right) \right] + EMA_{1p} \times \left[ 1 - \left( \frac{2}{1+n_1} \right) \right],$

		$EMA_2 = \left[ EMA_1 \times \left( \frac{2}{1+n_2} \right) \right] + EMA_{2p} \times \left[ 1 - \left( \frac{2}{1+n_2} \right) \right],$ $M = \frac{EMA_1}{EMA_2}, Mass_{Inx} = \text{rolling sum of } M \text{ over a period of } n_2.$
<b>Commodity Channel Index</b>	<i>CCI</i>	$L = \text{day's lowest price}, H = \text{day's highest price},$ $C = \text{closing price}, TP = \frac{H+L+C}{3}, n = 20, c = 0.015,$ $MA = \frac{\sum_{t=1}^n TP}{n}, MD = \frac{\sum_{t=1}^n TP - MA}{n},$ $CCI = \frac{TP - MA}{c \times MD}.$
<b>Schaff Trend Cycle</b>	<i>STC</i>	$C = \text{closing price}, n_1 = 23, n_2 = 50, n_3 = 3, n_4 = 10,$ $p = \text{prior value},$ $EMA_1 = \left[ C \times \left( \frac{2}{1+n_1} \right) \right] + EMA_{1p} \times \left[ 1 - \left( \frac{2}{1+n_1} \right) \right],$ $EMA_2 = \left[ C \times \left( \frac{2}{1+n_2} \right) \right] + EMA_{2p} \times \left[ 1 - \left( \frac{2}{1+n_2} \right) \right],$ $EMA_{diff} = EMA_1 - EMA_2,$ $EMA_{diff_{min}} = \text{rolling minimum of } EMA_{diff} \text{ over a period of } n_4,$ $EMA_{diff_{max}} = \text{rolling maximum of } EMA_{diff} \text{ over a period of } n_4,$ $S = 100 \times \frac{EMA_{diff} - EMA_{diff_{min}}}{EMA_{diff_{max}} - EMA_{diff_{min}}},$ $EMA_S = \left[ S \times \left( \frac{2}{1+n_3} \right) \right] + EMA_{Sp} \times \left[ 1 - \left( \frac{2}{1+n_3} \right) \right],$ $EMA_{S_{min}} = \text{rolling minimum of } EMA_S \text{ over a period of } n_4,$ $EMA_{S_{max}} = \text{rolling maximum of } EMA_S \text{ over a period of } n_4,$ $D = 100 \times \frac{EMA_S - EMA_{S_{min}}}{EMA_{S_{max}} - EMA_{S_{min}}},$ $STC = \left[ D \times \left( \frac{2}{1+n_3} \right) \right] + STC_p \times \left[ 1 - \left( \frac{2}{1+n_3} \right) \right].$
<b>Amihoud Illiquidity<sup>52</sup></b>	<i>Amih_L</i>	$C = \text{closing price}, V = \text{current volume}, V_d = C \times V, n = 21,$ $p = \text{prior value}, r = \frac{C}{C_p} - 1,$ $EMA_1 = \left[ V_d \times \left( \frac{2}{1+n} \right) \right] + EMA_{1p} \times \left[ 1 - \left( \frac{2}{1+n} \right) \right],$ $EMA_2 = \left[ r \times \left( \frac{2}{1+n} \right) \right] + EMA_{2p} \times \left[ 1 - \left( \frac{2}{1+n} \right) \right],$ $Amih_L = \frac{ EMA_2 }{EMA_1} \times 1000000.$
<b>Kyle's Lambda<sup>53</sup></b>	<i>Kyle_L</i>	$C = \text{closing price}, V = \text{current volume}, p = \text{prior value},$ $r = \frac{C}{C_p} - 1, n = 21, S_r = \begin{cases} 1 & \text{if } r_t > 0 \\ -1 & \text{if } r_t < 0 \end{cases}$ $V_d = S_r \times \ln(C \times V), R_{i,t} = \alpha_i + \beta_i V_d + \varepsilon_{i,t},$ $Kyle_L = \text{rolling } R_{i,t} \text{ over a period of } n.$

<sup>52</sup> The number 1000000 has been additionally put in the formula, as the original  $Amih_L$ , that is, without this number, is very small and often begins after sixth decimal place.

<sup>53</sup> Kyle's Lambda is a rolling OLS regression ( $R_{i,t}$ ) which is interpreted as the cost of demanding a certain amount of liquidity over a selected time period ( $n$ ).

<b>Corwin-Schultz</b>	<i>Corwin_Schultz</i>	$L = \text{day's lowest price}, H = \text{day's highest price},$
<b>Bid-Ask Spread Estimator<sup>54</sup></b>	<i>(CS)</i>	$n_1 = 5, n_2 = 21, HL_{sq} = \ln\left(\frac{H}{L}\right)^2, c = 3 - 2 \times 2^{\frac{1}{2}},$ $H_{max} = \text{rolling maximum of } H \text{ over a period of } n_1,$ $L_{min} = \text{rolling minimum of } L \text{ over a period of } n_1,$ $HL_{sqsum} = \text{rolling sum of } HL_{sq} \text{ over a period of } n_1,$ $B = \text{rolling mean of } HL_{sqsum} \text{ over a period of } n_2,$ $G = \ln\left(\frac{H_{max}}{L_{min}}\right)^2,$ $\dot{A} = \frac{\left(2^{\frac{1}{2}} - 1\right) \times \left(B^{\frac{1}{2}}\right)}{c}, A = \dot{A} - \left(\frac{G}{c}\right)^{\frac{1}{2}},$ $A = \begin{cases} A & \text{if } A \geq 0 \\ 0 & \text{if } A < 0 \end{cases}$ $CS = 2 \times \left(\frac{e^A - 1}{1 + e^A}\right).$

#### Technical features (source: tsfresh package)

<b>Aggregated autocorrelation</b>	<i>Agg_autocorrelation</i>	Calculates the value of an aggregation function $f_{agg}$ , where $f_{agg} \in \{\text{mean, variance, standard deviation, median}\}$ over the autocorrelation $R_l$ for different lags $l$ . $R_l = \frac{1}{(n-l)\sigma^2} \sum_{t=1}^{n-l} (X_t - \mu)(X_{t+l} - \mu),$ where $X_i$ are the values of the time series with length $n$ , and $\sigma^2$ and $\mu$ are estimators for its variance and mean. $f_{agg}(R_1, \dots, R_m)$ for $m = \max(n, maxlag)$ , where $maxlag$ is the maximum lag of choice.
<b>Approximate entropy</b>	<i>Approximate_entropy</i>	Implementation of a vectorised approximate entropy algorithm. Refer to (Yentes, Hunt, Schmid, McGrath, & Stergiou, 2013) for technical details on this algorithm. This feature uses default parameters in computations.
<b>Autoregressive process coefficient</b>	<i>Arr_coefficient</i>	This feature is obtained by fitting the unconditional maximum likelihood of an autoregressive $AR_k$ process, where $k$ is the maximum lag of the process. $X_t = \varphi_0 + \sum_{i=1}^k \varphi_i X_{t-i} + \varepsilon_t$ , where $X_i$ are the values of the time series, and $\varphi_i$ is the autoregressive process coefficient.
<b>Augmented Dickey-Fuller test statistic</b>	<i>Augmented_dickey_fuller</i>	This feature is obtained by applying the augmented Dickey-Fuller test on the values of the time series $X_i$ , and extracting the test statistic. Refer to (Cheung & Lai, 1995) for technical details on the augmented Dickey-Fuller test.
<b>Binned entropy</b>	<i>Binned_entropy</i> (BE)	$mb = \text{maximal number of bins},$ $n = \text{length of the time series } X_i,$ $p_k = \text{the percentage of samples in bin } k,$

<sup>54</sup> Corwin-Schultz Bid-Ask Spread Estimator allows to estimate bid-ask spreads from daily high and low prices, hence can serve as a measure of liquidity of the underlying asset.

		$BE = - \sum_{k=0}^{\min(mb,n)} p_k \log(p_k) \times 1_{(p_k > 0)}.$
<b>Time series complexity</b>	<i>Cid_ce (CID)</i>	<p><i>CID</i> is an estimate for a time series complexity which is proxied by numbers of peaks and troughs (such that the higher the number of peaks and troughs, the more complex the time series).</p> $CID = \sqrt{\sum_{t=1}^{n-1} (x_t - x_{t-1})^2},$ where $x_t$ is a value from the time series $X_i$ and $n$ is its length.
<b>Count above mean</b>	<i>Count_above_mean</i>	This feature returns the number of values of time series $X_i$ that are above its mean.
<b>Count below mean</b>	<i>Count_below_mean</i>	This feature returns the number of values of time series $X_i$ that are below its mean.
<b>Energy ratio by chunks</b>	<i>Energy_ratio_by_chunks (ERC)</i>	<p>This feature is a sum of squares of values in the chunk <math>i</math> out of the total number of chunks <math>N</math> expressed as a ratio of this sum of squares (<math>Ch_{i,N}</math>) over the full series energy parameter <math>FE</math>.</p> $FE = \sum_{t=1}^N x_t^2,$ $Ch_{i,N} = \sum_{t=1}^i x_{t,i}^2.$
<b>Aggregated FFT spectrum</b>	<i>FFT_aggregated</i>	<p>This feature returns the spectral centroid (mean), variance, skew, and kurtosis of the absolute Fourier transform spectrum.</p> <p>Refer to the official tsfresh package Github<sup>49</sup> page for calculations.</p>
<b>First location of maximum</b>	<i>First_location_of_maximum</i>	This feature returns the first location of the maximum value of time series $X_i$ , where the position is computed relatively to the length of $X_i$ .
<b>First location of minimum</b>	<i>First_location_of_minimum</i>	This feature returns the first location of the minimum value of time series $X_i$ , where the position is computed relatively to the length of $X_i$ .
<b>Fourier entropy</b>	<i>Fourier_entropy</i>	<p>This feature returns the binned entropy of the power spectral density of the time series with the use of the Welch method.</p> <p>Refer to (Welch P., 1967) or the official tsfresh package Github<sup>49</sup> page for full calculations.</p>
<b>Index mass quantile</b>	<i>Index_mass_quantile</i>	<p><i>Index mass quantile</i> calculates the relative index <math>i</math> of time series <math>X_i</math> where quantile <math>q</math> of the mass of <math>X_i</math> lies to the left of <math>i</math>. For example, if <math>q = 50\%</math>, this feature will return the mass centre of the time series <math>X_i</math>.</p>
<b>Kurtosis</b>	<i>Kurtosis (G2)</i>	<p>This feature returns a kurtosis for the values of time series <math>X_i</math>, such that</p> $M_4 = \frac{\sum_{t=1}^n (x_t - \mu)^4}{n}, M_2 = \frac{\sum_{t=1}^n (x_t - \mu)^2}{n}, \alpha_4 = \frac{M_4}{M_2^2}, g = \alpha_4 - 3,$

		$G2 = \frac{n-1}{(n-2)(n-3)}[(n+1) \times g + 6],$ where $x_t$ is a single value of $X_i$ , $\mu$ is a mean of $X_i$ , and $n$ is a length of $X_i$ .
<b>Last location of maximum</b>	<i>Last_location_of_maximum</i>	This feature returns the last location of the maximum value of time series $X_i$ , where the position is computed relatively to the length of $X_i$ .
<b>Last location of minimum</b>	<i>Last_location_of_minimum</i>	This feature returns the last location of the minimum value of time series $X_i$ , where the position is computed relatively to the length of $X_i$ .
<b>Lempel-Ziv complexity</b>	<i>Lempel_ziv_complexity</i>	A feature that is a complexity estimate based on the Lempel-Ziv algorithm. Refer to (Lempel & Ziv, 1976) or the official tsfresh package Github <sup>49</sup> page for calculations.
<b>Linear trend timewise</b>	<i>Linear_trend_timewise</i>	This combined feature returns p-values, correlation coefficients, intercepts, slopes, and standard errors of an estimate from a rolling OLS regression applied to the values of time series $X_i$ .
<b>Longest strike above mean</b>	<i>Longest_strike_above_mean</i>	This feature returns the length of the longest consecutive subsequence in the values of time series $X_i$ that is bigger than its mean.
<b>Longest strike below mean</b>	<i>Longest_strike_below_mean</i>	This feature returns the length of the longest consecutive subsequence in the values of time series $X_i$ that is smaller than its mean.
<b>Mean change</b>	<i>Mean_change</i>	<i>Mean change</i> is an average of the differenced values of time series $X_i$ over a period of $n$ , such that $Mean_{change} = \frac{1}{n-1} \sum_{t=1}^n x_t - x_{t-1},$ where $x_t$ is a value of time series $X_i$ .
<b>Mean absolute change</b>	<i>Mean_abs_change</i>	<i>Mean absolute change</i> is an average of the absolute differenced values of time series $X_i$ over its length $n$ , such that $Mean_{change} = \frac{1}{n-1} \sum_{t=1}^n  x_t - x_{t-1} ,$ where $x_t$ is a value of time series $X_i$ .
<b>Mean second derivative central</b>	<i>Mean_second_derivative_central (MSDC)</i>	This feature returns the mean value of a central approximation of the second derivative, such that $MSDC = \frac{1}{2(n-2)} \sum_{t=1}^{n-1} \frac{1}{2} (x_{t+2} - 2 \cdot x_{t+1} + x_t),$ where $x_t$ is a value of time series $X_i$ and $n$ is its length.
<b>Number CWT peaks</b>	<i>Number_cwt_peaks</i>	This feature returns the number of different peaks in time series $X_i$ . This feature is obtained by first smoothing the values of $X_i$ using ricker wavelet for widths ranging from 1 to $n$ , and then finding the number of peaks that occur at enough width scales and with sufficiently high Signal-to-Noise-Ratio (here set

		to 1 by default). Refer to (Du, Kibbe, & Lin, 2006) for further details on calculating Number CWT peaks that has been used in the tsfresh package.
<b>Partial autocorrelation</b>	<i>Partial_autocorrelation</i>	<p>This feature calculates the value of the partial autocorrelation <math>\alpha_k</math> function at the given lag <math>k</math>. Assuming the values used to calculate <math>\alpha_k</math> come from time series <math>X_i</math>, the value of <math>\alpha_k</math> can be expressed as</p> $\alpha_k = \frac{cov(R_{i,t}, R_{i,t-k} R_{i,t-1}, \dots, R_{i,t-k+1})}{\sqrt{var(R_{i,t} R_{i,t-1}, \dots, R_{i,t-k+1})var(R_{i,t-k} R_{i,t-1}, \dots, R_{i,t-k+1})}}$ <p>where <math>R_{i,t}</math> is an OLS regression applied to the values of time series <math>X_i</math>.</p>
<b>Permutation entropy</b>	<i>Permutation_entropy</i>	<p>This feature calculates permutation entropy which measures complexity of time series <math>X_i</math> by capturing order relations between its values, as well as by extracting a probability distribution of the ordinal patterns. Refer to (Bandt &amp; Pompe, 2002) for details on computing permutation entropy that has been used in the tsfresh package.</p>
<b>Sample entropy</b>	<i>Sample_entropy</i>	<p>This feature returns a sample entropy of the values of time series <math>X_i</math>. Sample entropy assesses the complexity of the underlying time series. Refer to (Richman &amp; Moorman, 2000) for details on calculating sample entropy that has been used in the tsfresh package.</p>
<b>Skewness</b>	<i>Skewness (G1)</i>	<p>This feature returns the sample skewness of the values of time series <math>X_i</math>, such that</p> $M_3 = \frac{\sum_{t=1}^n (x_t - \mu)^3}{n}, M_2 = \frac{\sum_{t=1}^n (x_t - \mu)^2}{n}, g = \frac{M_3}{M_2^{\frac{3}{2}}}$ $G1 = \frac{\sqrt{n(n-1)}}{n-2} \times g,$ <p>where <math>x_t</math> is a single value of <math>X_i</math>, <math>\mu</math> is a mean of <math>X_i</math>, and <math>n</math> is a length of <math>X_i</math>.</p>
<b>Spectral Welch density</b>	<i>Spkt_welch_density</i>	<p>This feature first calculates an estimate of the cross power spectral density of the values of time series <math>X_i</math> at different frequencies and returns the power spectrum of the different frequencies. Refer to (Welch P., 1967) or the official tsfresh package Github<sup>49</sup> page for full calculations.</p>
<b>Time reversal asymmetry statistic</b>	<i>Time_reversal_asymmetry_statistic (TRAS)</i>	<p>TRAS is calculated as</p> $TRAS = \frac{1}{n-2k} \sum_{t=1}^{n-2k} (x_{t+2k}^2 \cdot x_{t+k}) - (x_{t+k} \cdot x_t^2),$ <p>where <math>x_t</math> is a single value of time series <math>X_i</math>, <math>k</math> is a selected lag, and <math>n</math> is a length of <math>X_i</math>.</p>

<b>Variation coefficient</b>	<i>Variation_coefficient</i>	This feature returns a variation coefficient $CV$ which is calculated as $CV = \frac{\sigma}{\mu}$ , where $\sigma$ is a standard deviation of values of time series $X_t$ , and $\mu$ is their mean.
<b>Macroeconomic features (source: Bloomberg and Datastream)</b>		
<b>US government ten-year bond yield</b>	<i>US10Y_yield</i>	Yields on ten-year US government bonds.
<b>Effective Federal Funds Rate</b>	<i>Fed_rate</i>	This feature is a daily Effective Federal Funds Rate calculated by the Federal Reserve Bank of New York and serves as a dynamic proxy for the main interest rate set up by the Federal Reserve.
<b>US Consumer Price Index</b>	<i>US_CPI</i>	Seasonally adjusted Consumer Price Index (an inflation rate) as reported by the US Bureau of Labour Statistics.
<b>Baltic Dry Index</b>	<i>BDI</i>	Baltic Dry Index serves as a benchmark for the cost of shipping goods worldwide.
<b>Commodity Research Bureau Index</b>	<i>CRB_Spot</i>	CRBI acts as a representative indicator of commodity (mainly industrial) prices worldwide.

Table 12 to follow below lists all those features, referred to in this chapter as “external features”, that are applicable only to a specific asset class, with some exceptions pertaining to two of the three classes (e.g., a certain feature may be used both for predicting equity and FX regimes but not commodity regimes).

Table 12: Features usually applicable only to a specific asset class (rarely, to two).

Feature	Feature label	Formula and/or description
<b>Equities (source: Bloomberg and Datastream)</b>		
<b>Price / Sales ratio</b>	<i>PS</i>	The aggregated price / sales ratio $PS$ for the entire index is calculated by first: $PS_m = \frac{MC_m}{RV_m}$ , where $MC_m$ is a member's market capitalisation and $RV_m$ is member's total revenues over last 12 months, and then: $PS = \frac{\sum_m w_m PS_m}{\sum_m w_m}$ , where $w_m$ is a member's weight in the equity index.
<b>Price / Earnings ratio</b>	<i>PE</i>	The aggregated price / earnings ratio $PE$ for the entire index is calculated by first: $PE_m = \frac{P_m}{EPS_m}$ , where $P_m$ is a member's share price and $EPS_m$ is member's earnings per share over last 12 months, and then: $PE = \frac{\sum_m w_m PE_m}{\sum_m w_m}$ , where $w_m$ is a member's weight in the equity index.
<b>Price / Book ratio</b>	<i>PB</i>	The aggregated price / book ratio $PB$ for the entire index is calculated by first: $PB_m = \frac{P_m}{BPS_m}$ , where $P_m$ is a member's share price and $BPS_m$ is member's average book value per share over last 12 months, and then:

		$PB = \frac{\sum_m w_m P B_m}{\sum_m w_m}$ , where $w_m$ is a member's weight in the equity index.
<b>Dividend yield</b>	<i>DY</i>	The aggregated dividend yield <i>DY</i> for the entire index is calculated by first: $DY_m = \frac{P_m}{DPS_m}$ , where $P_m$ is a member's share price and $DPS_m$ is member's average dividend per share over last 12 months, and then: $DY = \frac{\sum_m w_m \frac{1}{DY_m}}{\sum_m w_m}$ , where $w_m$ is a member's weight in the equity index.
<b>FX rate</b>	<i>FX_rate</i>	FX rates of assets' countries of origin versus US dollar.
<b>Corporate bonds (credit) yield</b>	<i>Credit_YTW</i>	Yields on investment grade corporate bonds of different origin, such that US assets (e.g., S&P 500) have US credit yields, Eurozone assets (e.g., DAX Index) have Euro credit yields, or emerging markets assets (e.g., MSCI China Index) have emerging markets credit yields.
<b>Global corporate bonds (credit) yield</b>	<i>Global_credit_YTW</i>	An aggregated global corporate bond yield dominated by US credit.
<b>VIX Index</b>	<i>VIX</i>	Volatility Index <sup>5</sup> which stands for implied volatility of S&P 500.
<b>TED spread</b>	<i>TED_Spread</i>	TED spread is a difference between short-term US government debt (here the three-month treasury bill) and an interest rate on interbank loans (here the three-month LIBOR). This feature is often used in the industry as a measure of credit risk.
<b>LIBOR-OIS spread</b>	<i>LIBOR_OIS_Spread</i>	LIBOR-OIS spread is a difference between the three-month LIBOR, i.e., an interest rate on interbank loans, and the overnight index swap rate, set by the Federal Reserve. This feature serves as an indicator of potentially worsening credit conditions in a country, when the spread widens, and vice-versa, when the spread narrows.
<b>Yield curve</b>	<i>Yield_curve_2-10 Yield_curve_3M-2</i>	The yield curve is a difference between the country's shorter-term and longer-term government bond yields. In the work of this chapter there are two yield curves: a difference between two-year and 10-year bond yields and a difference between three-month and two-year bond yields. The yield curve is often used in the industry to predict economic contractions, such as whenever the difference becomes negative; it is usually associated with an economic recession to come.
<b>DXY Index</b>	<i>DXY</i>	DXY Index is an aggregated measure of US dollar strength versus Japanese yen, British pound sterling, Canadian dollar, Swedish krona, and Swiss franc. Note, DXY Index is not Bloomberg DXY Index, which, even though it also represents the strength of the US dollar, has a different currency basket that changes more often, as it depends on the US largest trading partners at a specific time period. Due to Bloomberg DXY Index having a shorter history, DXY Index had to be used instead.
<b>Eurozone Consumer Price Index</b>	<i>Euro_CPI</i>	Seasonally adjusted Consumer Price Index (an inflation rate) in the Euro area as reported by the Eurostat.
<b>Merrill Lynch Option Volatility Estimate (MOVE) Index</b>	<i>MOVE</i>	MOVE Index measures the future volatility in US government bond yields implied by current prices of options on bonds of various maturities. Similarly to the VIX Index, higher values of MOVE Index imply jittery economic conditions.
<b>J.P. Morgan Emerging Markets Bond Index (EMBI) Global spread</b>	<i>EMBI</i>	EMBI spread measures the difference between the aggregated yield of the dollar denominated debt of over 60 emerging market governments versus the yield on US government bonds of similar maturity. EMBI spread is often used in the industry to measure the credit risk in emerging markets relatively to the US.
<b>Commodities (source: Bloomberg and Datastream)</b>		

<b>Trade-weighted US dollar</b>	<i>USD_TWI</i>	Trade-weighted US dollar is a weighted average of the foreign exchange value of the US dollar against the currencies of a broad group of major US trading partners. The difference between DXY Index and USD TWI is that the latter is more commodity-driven; that is, it includes countries that rely heavily on global commodity prices.
<b>VIX Index</b>	<i>VIX</i>	See the description above.
<b>Yield curve</b>	<i>Yield_curve_2-10</i> <i>Yield_curve_3M-2</i>	See the description above. For commodities, only the US yield curve is used.
<b>Foreign Exchange (source: Bloomberg and Datastream)</b>		
<b>Trade-weighted US dollar</b>	<i>USD_TWI</i>	See the description above.
<b>Government ten-year bond yield</b>	<i>10Y_bond_yield</i>	Ten-year government bond yields from the country of the quote currency. For instance, if the FX pair is USDGBP, then British ten-year government bond yields are selected.
<b>Consumer Price Index</b>	<i>CPI</i>	Seasonally adjusted CPI (inflation) rate from the country of the quote currency. For example, if the FX pair is USDJPY, then the Japanese CPI is selected.
<b>FX three-month forward points</b>	<i>Fwd_points</i>	The number of basis points added (forward premium) to or subtracted from (forward discount) the current spot rate of a currency pair. Three-month FX forward rates reflect short-term interest rate differentials between both countries and indicate which currency (base or quote) is more "attractive" to be held over this time period.
<b>Long-term interest rate differentials</b>	<i>Interest_rate_diff</i>	Similar to FX forward points; however, this feature is a difference between ten-year interest rates (proxied by government bond yields) of base and quote currencies.
<b>Inflation rate differentials</b>	<i>Inflation_diff</i>	Difference between CPI rates of base and quote currencies. For instance, if the FX pair is USDCAD, then this feature will be a difference between US and Canadian CPIs.
<b>Terms of trade</b>	<i>Terms_of_trade</i>	<i>Terms_of_trade</i> is the relative price of exports in terms of imports and is defined as the ratio of export prices to import prices. This feature calculates the terms of trade of the quote currency (e.g., in USDCHF, Swiss terms of trade is computed).
<b>Equity index</b>	<i>Equity_inx</i>	The main equity index of the quote currency in a FX pair. For example, in case of USDEUR, Eurostoxx Index is selected, whereas in case of USDSEK, the Swedish OMX Stockholm 30 Index is selected.
<b>S&amp;P 500 Index</b>	<i>SPX</i>	The main equity index of USA, which also serves as a proxy of the current and implied condition of the US economy. S&P 500 Index has been selected as an additional feature, since US dollar is a base currency in almost every FX pair.

The data from Tables 11 and 12 goes through the data preparation procedure described in Section 4.2.2, on the following page. However, before this step is done, the data is split into training and testing sets, where the "training set" also incorporates validation sets. The training set is 85% of the entire dataset, while the testing (holdout) set is the remaining 15%. Such a separation is in line with the final dataset split done in Chapter 3. The 15% holdout set is equal to roughly four and a half years of out of sample data in each asset class, which may be considered a sufficient length of time to evaluate each model's long-term performance.

## 4.2.2 Methodology

This section will describe the preparation of the data for modelling, and details of the modelling process. The data preparation comprises of feature engineering, fractional differencing, and label generation (which builds on the outcome of Chapter 3), and finally the stacking of the data together. The modelling process comprises of the cross-validation setup, hyperparameter tuning, feature selection, and finally the testing of the performance of the models on the holdout sets. All of these elements of the methodology will be described in more detail below.

### *Data preparation*

**Feature engineering.** The feature engineering step first involves creating the technical indicators listed in Table 11 in Section 4.2.1 on data from *High-Low-Close-Volume* (HLCV) data, apart from FX rates, for which there is no volume data, thus only take HLC data and no volume-based technical indicators are computed. Once the technical features are complete, the closing price of each asset is taken and used for creation of over 100 statistical features and its variants (also listed in Table 11) by the tsfresh package.

The engineered data from Table 11, as well as the external features listed in Table 12, are subsequently put together. Any NaN values between values are filled using the medians over rolling fixed (weekly) windows of past data, to avoid data leakage and smoothly interpolate between the first and last available values. In addition, the NaN values associated with monthly macro releases that has been unavailable in the most recent months are forward-filled. Other NaN values, such as those associated with a feature of a shorter history, are simply dropped, to avoid their being filled using datapoints coming from the future.

The obtained data frame is then ready for the next step, which involves fractional differencing.

**Fractional differencing.** The data from the previous step is scaled using the fractional differencing process described in Section 4.1.1, using the Fracdiff<sup>55</sup> package which allows fractionally differenced features to be calculated quickly, as well seeking the optimal differentiation order that makes time-series stationary while also preserving its memory. The optimisation of the differentiation order is done on the training set only (85% of the data), whereas the holdout set (15% of the data) is transformed using the selected order.

Another advantage of the Fracdiff package is that it uses a rolling fixed window to find the optimal order, as well as scale the data. This builds upon (López de Prado, 2018), where it has been noted that such a solution avoids any data leakage, as well as being robust to live data “flowing in”. Thus, even if the main training set (85%) is split into a number of training-validation sets (as will be outlined in the modelling section on p. 161), the fractionally differenced features remain exactly the same, i.e., the scaling process always goes forward without changing the past. Thus, there is no need to additionally fit and transform the algorithm on the newly created folds of training and validation sets.

Once fractional differencing has been done, the closing price of an asset is taken to generate labels that will be predicted by Random Forest.

**Label generation and prediction.** To recall, the KAMA+MSR model creates four regime classes: bullish / low variance, bullish / high variance, bearish / low variance, and bearish / high variance, though with only the first and the fourth of these being used for trading. Such a separation is based on the results from Chapter 3 (see Section 3.3.2), which showed the KAMA+MSR framework to perform best when initial four regimes are turned into three, where the third regime combines bearish / low variance and bullish / high variance into the “other” regime not suitable for trading. However, a minor caveat associated with this approach is that once costs are accounted for, a bullish / low variance or bearish / high variance period may not be profitable after all, since the costs can wipe out the gains. In

---

<sup>55</sup> See <https://github.com/fracdiff/fracdiff> for the official Github repository. Accessed on 10/08/2022.

addition, bullish / high variance or bearish / low variance states, which become the “other” regime, contain periods of up-trending and down-trending markets, respectively. This often happens when after, for instance, a bullish / low variance regime, a price of a certain asset, such as a commodity, soars significantly and then suddenly drops, thus effectively becoming a bullish / high variance state due to volatile movement of the market. Similarly, subsequent to a bearish / high variance regime, a bearish / low variance state may occur when the volatility of the asset’s price becomes milder, and it initially falls but rebounds afterwards (a typical “bear rally” already mentioned in Section 4.1.1). Such periods could confuse the learning algorithm, as well as potentially result in a model which produces less profitable strategy (outlined below in the modelling section), since it is solely focused on either bullish / low variance or bearish / high variance regimes. Thus, this experiment enhances the original labels by extending the bullish / low variance and bearish / high variance states to the peak and trough of bullish / high variance and bearish / low variance regimes, respectively. Finally, each new regime is adjusted for costs attributable to the selected asset class from Table 6 in Chapter 3 to avoid situations in which a market may be identified as, for instance, bullish / low variance, but where once costs are accounted for, would not, in fact, generate positive returns but ones that were either flat or negative. If such a situation occurs, this state is categorised as “other regime”, instead. In effect the classification labels  $y$  become:

- 0 = “other” regime,
- 1 = “BL / LV + extension to the peak of the next BL / HV” regime,
- 2 = “BR / HV + extension to the trough of the next BR / LV” regime,

which simplifies to:

- 0 = “other” regime,
- 1 = “bullish” regime,
- 2 = “bearish” regime,

where “BL / LV” is bullish / low variance, “BL / HV” is bullish / high variance, “BR / HV” is bearish / high variance, “BR / LV” is bearish / low variance state, and both 1 and 2 regimes are cost-adjusted.

Each model uses the time series feature space  $X_c$  to predict  $y_{t+1}$ , where  $c$  is the feature set (listed in Tables 11 and 12) attributable to one of the three asset classes used in this chapter's experiment.

After label generation, the data comes through to the final stage of preparation, stacking, before it is fed to the training and validation algorithm.

**Stacking of the datasets.** The above-described feature engineering, fractional differencing, and label generation steps are done for each asset from the class. Thus, there is a loop in the code that goes through each asset and prepares it for modelling. Once each iteration is done, this new data frame is stacked upon the completed data frames (excluding the very first loop that serves as a base) to create a panel-like dataset, identified by date index and asset name. See Table 13 below for an example:

*Table 13: Example of a panel-like dataset ready for modelling.*

Date_index	Asset_name	Target_variable	Feature_1	Feature_2	Mutual_feature_1
01/01/2000	Asset_1	Regime_asset_1	Ft_1_asset_1	Ft_2_asset_1	Common_ft_1
02/01/2000	Asset_1	Regime_asset_1	Ft_1_asset_1	Ft_2_asset_1	Common_ft_1
03/01/2000	Asset_1	Regime_asset_1	Ft_1_asset_1	Ft_2_asset_1	Common_ft_1
01/01/2000	Asset_2	Regime_asset_2	Ft_1_asset_2	Ft_2_asset_2	Common_ft_1
02/01/2000	Asset_2	Regime_asset_2	Ft_1_asset_2	Ft_2_asset_2	Common_ft_1
03/01/2000	Asset_2	Regime_asset_2	Ft_1_asset_2	Ft_2_asset_2	Common_ft_1

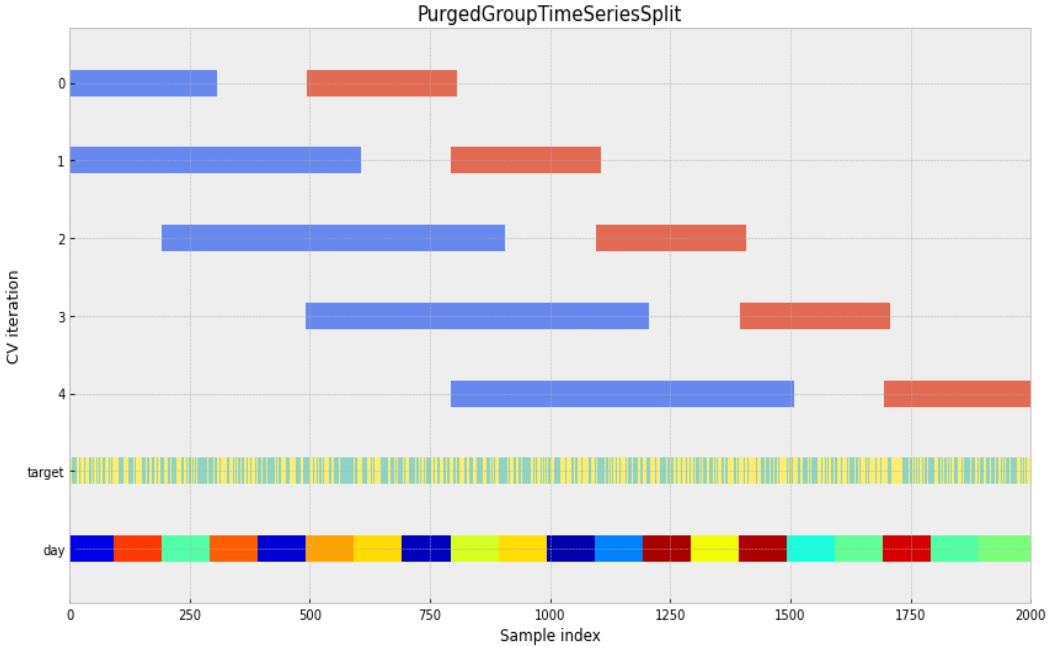
The panel-like dataset is then sorted on the basis of date index and asset name (as can be seen in Table 13 above). Finally, the same data is sorted so that the asset with the oldest history goes to the top of the stacked dataset, while the asset with the shortest history goes to the bottom. This step is necessary to avoid cross-validation errors, such as using data from the future to validate on the past.

The dataset is now ready for training and validation, which will be described in detail in the following section.

## *Modelling*

**Cross-validation setup.** Before implementing hyperparameter optimisation, it is necessary to describe how the 85% of the training sample was split into different training and validation sets. As was previously shown, the dataset used in this experiment has a panel-like form which is a blend of cross-sectional (different assets) and time-series (assets ordered by time) data. Such dataset cannot simply use a K-fold cross-validation which merely splits data into shuffled training and validation sets, as the sequential nature of the data will lead to overfitting of the machine learning algorithm (i.e., the model might be validated on the past using a sample from the future). Another issue is the use of lagged statistical features derived from the tsfresh package which may leak from the validation to the training set in a situation when the former commences right after the latter. Such data leakage may also lead to overfitting, as the model might find it easier to predict certain samples close to the beginning of the validation set. This may erroneously indicate that the optimised model is accurate, since the scoring metric will be inflated by that part of the predictions close to the end of the training set. To tackle both issues in cross-validation, the work of this chapter implements a method called *Purged Group Time-Series Split* (PGTS) recommended by (López de Prado, 2018). PGTS makes sure the model is trained on the past and validated on the future while leaving a gap between training and validation sets. The “group” term in the name refers to the handling of panel-like datasets that are subdivided into different groups, such as assets. Figure 5 on the following page gives an outline of the PGTS process.

Figure 5: Outline of the PGTS process, where blue bars are training and orange bars are validation sets. The target stands for the predicted label, while the day is a maximum number of days within training and validation sets (here, each block is a single day).



PGTS requires the specification of the number of splits, the maximum training and validation day windows, as well as the day window for the gap between training and validation sets (blue and orange bars in Figure 5 above). Table 14 lists the settings used in creating models for this chapter.

Table 14: PGTS settings.

Parameter	Value
<b>Number of splits</b>	4
<b>Maximum training window</b>	3024 days (roughly 12 trading years)
<b>Maximum validation window</b>	1008 days (roughly 4 trading years)
<b>Gap window</b>	21 days (a trading month)

The validation window has been set to roughly four years because this is also the approximate length of the holdout set of each asset class (thus, each model). The maximum training window was picked after a preliminary analysis which showed that a too-short window ends up with less samples to learn from, while a too-long window poses a problem for assets with a short history span, as they may end up

being validated on just one split instead of at least two, for example. The splitting framework of Table 14 is also consistent with the 75%-25% training-validation splits used during optimisation of the KAMA component of the KAMA+MSR model described in Chapter 3.

PGTS is subsequently used for hyperparameter tuning, feature selection, and, finally, the modelling process, which are described next.

**Hyperparameter tuning.** The hyperparameters of Random Forest, described in Section 2.2.4, need optimising in order to tackle the bias vs. variance trade-off, a common machine learning issue also discussed in Section 2.2.4. In this experiment, the optimal hyperparameters are found with the use of the Optuna<sup>56</sup> package. Table 15 presents the search spaces of the hyperparameters, which apply to every constructed model.

*Table 15: Hyperparameters of Random Forest which are optimised in every experiment.*

Hyperparameter	Type of data	Search space
<i>n_estimators</i>	Integer	{10,...,300}
<i>max_depth</i>	Integer	{1,...,20}
<i>min_samples_split</i>	Integer	{1,...,100}
<i>min_samples_leaf</i>	Integer	{1,...,100}
<i>max_samples</i>	Float	{0.1,...,1.0}
<i>min_weight_fraction_leaf</i>	Float	{0.0,...,0.05}
<i>max_features</i>	Float	{0.2,...,1.0}

The reason for choosing these particular search space intervals stems from recommendations of (López de Prado, 2018), as well as common approaches to Random Forest optimisation, and the need to reduce the time needed to train each model, which was particularly impacted by a number of estimators. As was also mentioned in Section 2.2.4, while giving descriptions of each Random Forest hyperparameter, the rest of the hyperparameters are assigned their default values,

---

<sup>56</sup> Optuna is a popular hyperparameter tuning package that is model agnostic, fast, and can dynamically construct the search spaces for the hyperparameters. See (Akiba, Sano, Yanase, Ohta, & Koyama, 2019) for technical details on Optuna.

with the exception of class weights, which were chosen to be "balanced". This is because of discrepancies between the number of instances of each class, with the "bearish" regime being the rarest, and the "other" regime being the most common. Assuming balanced class weights in Random Forest makes sure the algorithm does not focus on the most frequent regime label while disregarding the other two, which will decrease its out-of-sample performance.

The optimal hyperparameters are found during the validation of the model on the PGTS splits described earlier. By "optimal", it is meant that set of hyperparameters which maximises the scoring metric outlined in the strategy back-testing section to follow. The optimisation process is initially performed using a limited number of trials to loosely tune the model using all available variables for the feature selection procedure described briefly below. After the feature selection stage, the subset of variables with the largest predictive power is selected for the final modelling. This step involves a higher number of optimisation trials to select the optimal hyperparameters. Table 17 in Section 4.3.1 lists the selected hyperparameters after carrying out the optimisation for each created model.

As briefly mentioned above, between the loose hyperparameter tuning and final hyperparameter optimisation, a feature selection step is performed; this process is described in the paragraph to follow.

**Feature selection.** Feature selection uses the BorutaShap method outlined in Section 4.1.2 to find those features with the largest predictive power, in order to reduce the dimensionality of the data. The problem with the default setup of BorutaShap is that it is not prepared for time-series cross-validation, as the original package uses K-fold splits for validation (and thus for picking the most important features). For the purpose of implementing BorutaShap in this experiment, the original code was tweaked to include the PGTS splits described previously in order to analyse feature importance on the same validation splits used in loose tuning and final optimisation. To achieve this, a separate loop with the number of iterations equal to the number of validation splits was implemented. Each iteration takes a single training and validation split from PGTS, runs BorutaShap, and stores the

features selected to be important by the algorithm. After completing all iterations, the number of times a certain feature was deemed important is summed, so that only the features which had the largest predictive power throughout the whole time (over the entire loop) are ultimately selected for use in the final hyperparameter optimisation. This makes sure the selected features have been continuously significant, regardless of the time period they were validated on. In other words, features which were never important, or may have been important in the past but lost their predictive power over time, are dropped.

Thanks to the tweaked BorutaShap algorithm, it has been possible to pick a set of features whose predictive power does not decay over time. This final set of features per each model will be shown in Section 4.3, while presenting charts with feature importance scores.

Before wrapping up the methodology section, it is necessary to also describe how each model's performance has been analysed. The next section discusses strategy back-testing, which incorporates the validation metric that the hyperparameter tuning package aims to maximise, as well as out of sample trading performance comparison between the trained models and selected benchmarks.

#### *Back-testing strategy and comparing models to benchmarks*

**Acting on the models' predictions.** Initially, it was assumed that the models' predictions would be used to establish a trend-following strategy, that is, a high probability of a "bullish" regime would give a buying signal, whilst a high probability of a "bearish" regime would give a shorting signal. (A high probability of the "other" regime after previously having entered a long or short position in an asset would indicate a point of time to close this position.) However, after preliminary results it was discovered that the models' predictions had in fact produced contrarian signals. In other words, a high probability of a "bullish" regime here indicated an exhausted and overbought market, which should spur an action to enter a short position in the underlying asset, while a high probability of a "bearish" regime here pointed at oversold market conditions, thus recommending entering a long position in the underlying asset. Contrarian models are commonly used in the industry; many

trading strategies act on, and many indicators provide an alarm about, overbought and oversold conditions of the market, and recommend assuming a contrarian position, shorting when overbought or buying when oversold.

Following the discovery that a contrarian interpretation of the models' signals was advisable, the following strategy was adopted:

- When the “bullish” regime is predicted, a short position should be assumed in the underlying asset.
- When the “bearish” regime is predicted, a long position should be assumed in the underlying asset.
- When the “other” regime is predicted, an asset should be sold (if it was bought) or bought back (if it was shorted).

**Outline of the strategy.** To check how each model performs, a back-testing strategy based on the generated regime predictions was implemented. Since each model predicts regimes for only one asset class, three portfolios were set up: equity-only, commodity-only, and FX-only. As was mentioned shortly before, every time a model predicts a regime shift in a specific asset of a class, the same asset is either bought (long position), shorted (short position), sold (if bought before), or bought-back (if shorted before). In the case of shorting an asset, the strategy chooses not to use a negative weight that could be used to acquire the assets that simultaneously received a "buy" signal. If there are multiple regime shifts predicted at the same time—that is, there is a predicted regime switch for several assets from one class—the mean weight is assumed across the assets. Regardless of the position assumed within an asset, whenever the model predicts the “other” regime, the open position is closed. As each model predicts the probability of “bullish”, “bearish”, and “other” regimes one-step-ahead ( $y_{t+1}$ ) the current closing price is used to calculate returns per trade.

Back-testing is performed first on validation sets, and, after training and validation steps are completed, on the holdout set (last 15% of the entire dataset). The model uses a given day's data to predict regime shifts for the next day. Given all

$t = 1 \dots T$  trading days in the analysed period (e.g., validation or holdout), the daily cost-adjusted trading return  $r_M(t)$  has been calculated as

$$r_M(t) = \sum_{\substack{\text{assets} \\ i}} \left( w_{L,i}(t)r_{L,i}(t) + w_{S,i}(t)r_{S,i}(t) \right), \quad (43)$$

in which  $r_{L,i}(t)$  is the return associated with longing (similarly, shorting) asset  $i$  on day  $t$ , and  $w_{L,i}(t)$  is the weight associated with longing (and similarly, for shorting) asset  $i$  on day  $t$ . Note, the returns are diminished by trading costs attributable to each asset class; for commodities and FX these are assumed to be the same as in Table 6 from Chapter 3. However, with regards to equities, the trading costs were halved after having consulted their original amounts with industry experts who indicated that the assumed equity trading costs from Table 6 in Chapter 3 are too conservative, and, in fact, not realistic for assets of such liquidity. Finally, the weights from Equation 43 are given by

$$w_{L,i}(t) = \begin{cases} \frac{1}{n_{L,t}} & \text{if asset } i \text{ is longed on day } t \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

where  $n_{L,t}$  is the number of assets longed (similarly, shorted) on day  $t$ . Note that an asset does not need to be traded, either longed or shorted, on a given day; this condition is achieved by setting  $w_{L,i}(t) = w_{S,i}(t) = 0$ .

The daily cost-adjusted trading returns are a critical part of the validation metric that the hyperparameter optimisation package aims to maximise. The paragraph below describes this metric in more detail.

**Validation / scoring metric.** The experiment conducted in this chapter does not use a typical machine learning scoring metric for hyperparameter optimisation. Since the purpose of each model is to predict regime switches ex ante and generate profits while acting on these signals, a financial metric has instead been selected to be maximised by the hyperparameter tuning package. Specifically, the daily cost-adjusted trading returns  $r_M$  are used to calculate an annualised *Sortino ratio* (ST), as defined on the following page. The choice of the Sortino ratio has been based on the

fact the underlying regimes differ significantly in terms of volatility. In Chapter 3 this issue has been tackled by mixing assets with cash, thus calculating a Sharpe ratio was sufficient; however, here the trades do not include cash that could tame the increased variance during, for instance, the bearish regime. Thus, the calculated Sharpe ratio may end up being decreased, even if the trade has been successful, due to significant spikes in volatility (to recall, a Sharpe ratio is calculated as mean returns divided by the standard deviation of these returns). The Sortino ratio instead uses the semi-variance of the underlying returns, which do not incorporate positive swings. The annualised ST (assuming 252 trading days in a year) can be expressed as

$$ST = \frac{\mu(r_M - r_f)}{\sigma_{neg}(r_M - r_f)} \times \sqrt{252}, \quad (45)$$

where  $r_f$  is the risk-free return (here assumed to be 0),  $\mu(\cdot)$  denotes the mean value of the daily-varying quantity (here the difference between  $r_M$  and  $r_f$ ), and  $\sigma_{neg}(\cdot)$  its negative semi-variance (downside deviation), over all  $t = 1 \dots T$  trading days of the analysed period.

The hyperparameter tuning process tries to optimise parameters that maximise the average ST over PGTS validation splits. Additionally to the ST, though not optimised within the model, the *Matthews Correlation Coefficient* (MCC) metric has been used to check the robustness of the model from a machine learning perspective. In the multiclass case, the MCC can be computed directly from the confusion matrix  $C$  for  $K$  classes as

$$MCC_m = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) \times (s^2 - \sum_k^K t_k^2)}}, \quad (46)$$

where  $t_k = \sum_i^K C_{ik}$  (the number of times class  $k$  truly occurred),  $p_k = \sum_i^K C_{ik}$  (the number of times class  $k$  was predicted),  $c = \sum_k^K C_{kk}$  (the total number of samples correctly predicted), and  $s = \sum_i^K \sum_j^K C_{ij}$  (the total number of samples). For binary classification, the multiclass MCC ranges between -1 and 1 (noting that a result of -1 would correspond to every example being classed as its opposite, a situation not normally encountered in practice unless there is a problem in the code); in the multiclass case, assignment of all examples to one or other of the alternate classes

(again, unlikely in practice) would typically lead to a multiclass MCC somewhere between -1 and 0.<sup>57</sup>

Equation 46 can, however, be criticised for potentially hiding the fact that one class in particular was badly predicted; it is also unhelpful if for some reason one of the classes is either more important or less importance than the others. This is in fact the case here. The main purpose of the model was to predict “bullish” and “bearish” regimes, while the “other” regime was only used to close the open position; an investor would thus be more interested to know when the asset potentially enters a “bullish” or “bearish” regime rather than merely “other”. For this reason, it is also valuable to present the MCCs for all three regimes separately, using the following formula for binary MCC,

$$MCC_b = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (47)$$

where  $TP$  is the number of true positives,  $TN$  is the number of true negatives,  $FP$  is the number of false positives, and  $FN$  is the number of false negatives.

The binary MCC was used to check that model was not making predictions—especially “bullish” and “bearish”—at random (which would be evidenced by an MCC value around zero). This is particularly important while testing each model’s performance on the 15% holdout set, as the combination of both a high ST and a high MCC would indicate that the model was indeed capable of predicting regime shifts ex ante accurately, and that the strategy based on these signals was reliable.

Apart from checking the ST on the holdout set, each model’s trading performance has additionally been compared to selected benchmarks. This process is described below.

**Model trading performance comparison.** Each Random Forest model’s cumulative trading performance (also referred to as cumulative geometric returns), calculated as in Equation 48 on the following page,

---

<sup>57</sup> The actual minimum value depends on the number and distribution of ground true labels. Refer to (Jurman, Riccadonna, & Furlanello, 2012) for more details.

$$Q(t) = \left[ \prod_{s=1}^t (1 + r_M(s)) \right] \times 100 \quad (48)$$

is compared to several chosen benchmarks out of sample, including those already listed in Table 10. Table 16 below summarizes the chosen benchmarks.

*Table 16: Benchmarks used for financial performance assessment in the three different asset classes, where the KAMA+MSR model is the regime detection (as opposed to prediction) model developed in Chapter 3. Note, all data series terminate on 29/04/2022.*

Asset Class	Benchmark	Start date
<b>Equities</b>	ACWI Index	06/03/2018
	Three-state HMM	06/03/2018
	KAMA+MSR model	06/03/2018
	All equity assets listed in Table 10	06/03/2018
<b>Commodities</b>	Bloomberg Commodity Index	19/12/2017
	Three-state HMM	19/12/2017
	KAMA+MSR model	19/12/2017
	All commodity assets listed in Table 10	19/12/2017
<b>FX</b>	Bloomberg DXY Index	20/04/2018
	Three-state HMM	20/04/2018
	KAMA+MSR model	20/04/2018
	All FX assets listed in Table 10	20/04/2018

The financial benchmarks, namely the ACWI Index, the Bloomberg Commodity Index, the Bloomberg DXY Index (also known as main or index benchmarks), and the assets from Table 10, are proxies of a long-only strategy which aims to replicate the cumulative performance in case of buying the underlying asset at the beginning of the timespan from Table 16 and selling it at the end. With regards to the main benchmarks, an annualised *information ratio* (IR) is computed to compare risk-adjusted excess returns. The annualised IR is calculated as

$$IR = \frac{\mu(r_M - r_B)}{\sigma(r_M - r_B)} \times \sqrt{252}, \quad (49)$$

where  $r_B$  are benchmark returns, and  $\sigma(\cdot)$  is a standard deviation of the excess returns  $(r_M - r_B)$  over all  $t = 1 \dots T$  trading days of the analysed period.

The Hidden Markov Model (HMM), discussed in Section 2.2.4, and the KAMA+MSR model from Chapter 3, serve as rather naïve strategies, in which the action is taken based either on the predicted (HMM) or detected (KAMA+MSR) regime shift. The mechanism by which the action is taken for these benchmarks works identically to the procedure described in the beginning of this section (back-testing strategy): whenever a regime shift is predicted or detected for an asset, the same asset is either bought, sold, shorted, or bought back. The trading performances of HMM and KAMA+MSR are calculated identically as for the models that are the main topic of this chapter, that is, using Equation 43 and finally Equation 48 for the cumulative performance.

## 4.3 Results and discussion

This section will present results of this chapter's main experiment, as well as discussing these results simultaneously, for clarification purposes.

### 4.3.1 Optimal hyperparameters

As described in the previous section, the selected hyperparameter tuning method was set to maximise the Sortino ratio based on the implemented trading strategy over PGTS validation splits. Table 17 on the following page lists the optimal Random Forest hyperparameters per each model validated.

Table 17: Optimal hyperparameters after tuning over PGTS validation splits.

Model	Hyperparameters	Optimal values
<b>Equity Random Forest model</b>	<i>n_estimators</i>	220
	<i>max_depth</i>	13
	<i>min_samples_split</i>	76
	<i>min_samples_leaf</i>	95
	<i>max_samples</i>	0.3649
	<i>min_weight_fraction_leaf</i>	0.0454
	<i>max_features</i>	0.2481
<b>Commodity Random Forest model</b>	<i>n_estimators</i>	280
	<i>max_depth</i>	3
	<i>min_samples_split</i>	18
	<i>min_samples_leaf</i>	95
	<i>max_samples</i>	0.1247
	<i>min_weight_fraction_leaf</i>	0.0213
	<i>max_features</i>	0.4001
<b>FX Random Forest model</b>	<i>n_estimators</i>	240
	<i>max_depth</i>	7
	<i>min_samples_split</i>	22
	<i>min_samples_leaf</i>	60
	<i>max_samples</i>	0.3603
	<i>min_weight_fraction_leaf</i>	0.0349
	<i>max_features</i>	0.3410

Table 17 demonstrates several similarities between each model's optimal hyperparameters. First, the number of trees (*n\_estimators*) is above 200 in each case (the allowed maximum being 300), evidencing that a complex model is needed to solve these problems. Second, the Equity and Commodity models have identical *min\_samples\_leaf*, while the Equity and FX model have almost the same *max\_samples*. Finally, the Commodity and FX models have similar values of *min\_samples\_split*, in contrast to the Equity model whose value for this hyperparameter differs significantly. (*min\_weight\_fraction\_leaf* was bounded between 0 and 0.05, thus the similar values for this parameter among the models is not remarkable.)

The next section focuses on feature importance for each model and presents graphs showing the most important features in terms of their predictive power for each regime.

### 4.3.2 Feature importance

Figures 6, 7, and 8 show the top 10 most important features out of those selected by the BorutaShap algorithm, in terms of their absolute impact on the predictions from the Equity, Commodity, and FX Random Forest models, respectively. The impact of each feature on prediction was computed using the Shapley values discussed in Section 4.1.2. It should be noted that the following graphs and tables present feature importances only for the 15% holdout set, as this is more relevant than the same statistics for the training set. In addition, Tables 18, 19, and 20 show the rankings of top 20 features divided into the ten features with the most positive and the ten features with the most negative impact on each model's predictions, for each asset class. Note that Table 20, which presents feature importances for the FX Random Forest model, lists only the 18 top features, as in this case there are only eight features having a positive impact on the model's predictions. (Tables 40, 41, 42 in Appendix B list all the features that each model uses for regime prediction.)

*Figure 6: Feature importances for the Equity Random Forest model.*

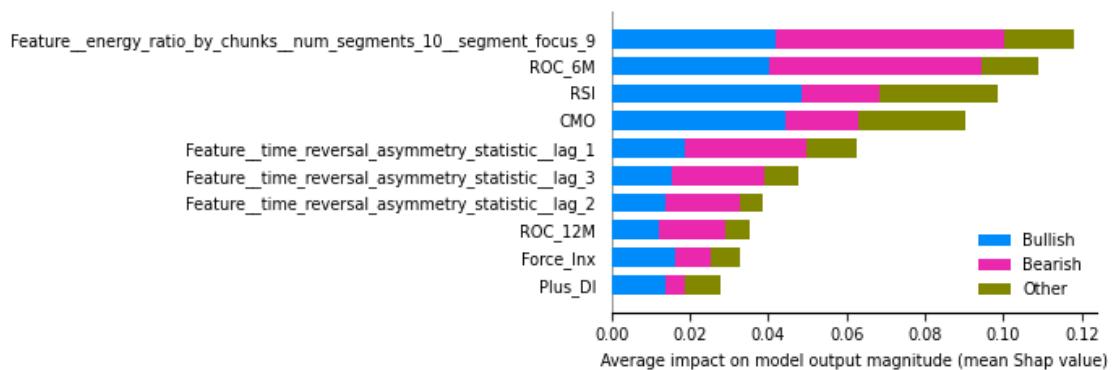


Figure 7: Feature importances for the Commodity Random Forest model.

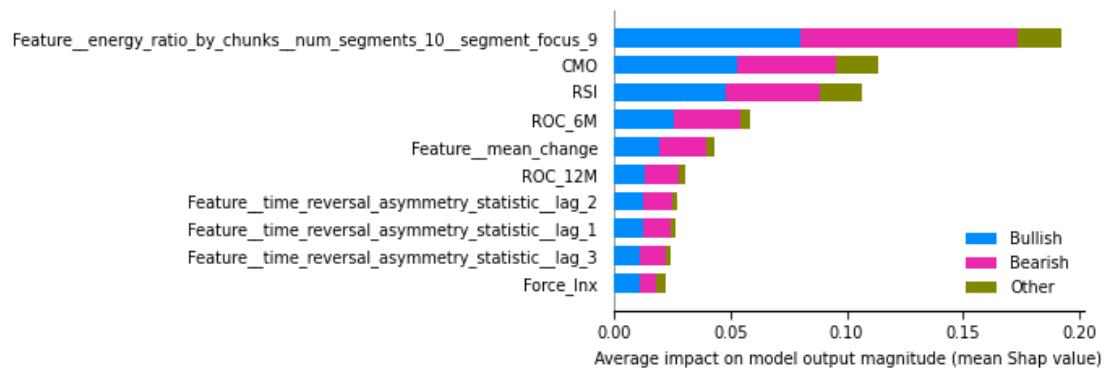


Figure 8: Feature importances for the FX Random Forest model.

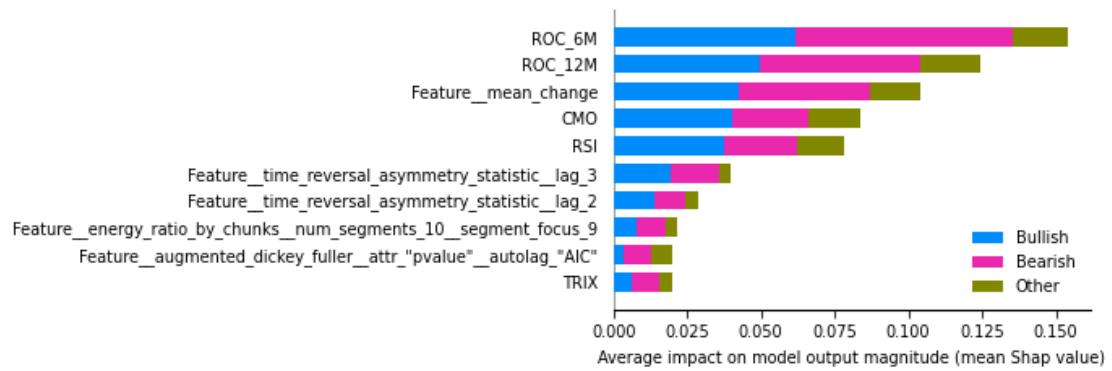


Table 18: Top 20 features with the 10 most positive and 10 most negative impact on the Equity Random Forest model's predictions.

"Bullish" regime		"Bearish" regime		"Other" regime	
Top 10 features with positive impact	Mean Shap value	Top 10 features with positive impact	Mean Shap value	Top 10 features with positive impact	Mean Shap value
Last_location_of_maximum	0.0005	ROC_6M	0.0172	Time_reversal_asymmetry_statistic_lag_1	0.0064
Minus_DI	0.0003	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_9	0.0167	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_9	0.0043
First_location_of_maximum	0.0002	Time_reversal_asymmetry_statistic_lag_1	0.0092	Time_reversal_asymmetry_statistic_lag_3	0.0042
Mean_Std_Month	0.0001	Time_reversal_asymmetry_statistic_lag_3	0.0064	Augmented_dickey_fuller_attr_"pvalue"_autolag_"AIC"	0.0041
Plus_DI	0.0000	Time_reversal_asymmetry_statistic_lag_2	0.0053	ROC_12M	0.0041
MFI	0.0000	ROC_12M	0.0047	Augmented_dickey_fuller_attr_"teststat"_autolag_"AIC"	0.0030
10Y bond yield	0.0000	Force_Inx	0.0045	Mean_change	0.0026
EMBI	0.0000	Mean_change	0.0029	Last_location_of_maximum	0.0015
US CPI	0.0000	CMO	0.0028	Time_reversal_asymmetry_statistic_lag_2	0.0013
Top 10 features with negative impact	Mean Shap value	Top 10 features with negative impact	Mean Shap value	Top 10 features with negative impact	Mean Shap value
Ulcer	-0.0029	Fourier_entropy_bins_100	0.0000	Awesome	-0.0002
Augmented_dickey_fuller_attr_"teststat"_autolag_"AIC"	-0.0043	Yield_Curve_2-3M	0.0000	Credit_ytw	-0.0003
Mean_change	-0.0055	Global_credit_ytw	0.0000	BB	-0.0005
Augmented_dickey_fuller_attr_"pvalue"_autolag_"AIC"	-0.0055	PE	0.0000	CCI	-0.0006
Time_reversal_asymmetry_statistic_lag_2	-0.0066	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_8	-0.0001	ROC_1M	-0.0006
ROC_12M	-0.0087	EMBI	-0.0001	Minus_DI	-0.0006
Time_reversal_asymmetry_statistic_lag_3	-0.0106	VIX	-0.0001	CMO	-0.0011
Time_reversal_asymmetry_statistic_lag_1	-0.0156	Cid_ce_normalize_True	-0.0002	Plus_DI	-0.0012
ROC_6M	-0.0172	First_location_of_maximum	-0.0012	RSI	-0.0012
Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_9	-0.0210	Last_location_of_maximum	-0.0020	Force_Inx	-0.0022

Table 19: Top 20 features with the 10 most positive and 10 most negative impact on the Commodity Random Forest model's predictions.

“Bullish” regime		“Bearish” regime		“Other” regime	
Top 10 features with positive impact	Mean Shap value	Top 10 features with positive impact	Mean Shap value	Top 10 features with positive impact	Mean Shap value
Ultimate	0.0001	Energy_ratio_by_chunks_n um_segments_10_segm t_focus_9	0.0220	Energy_ratio_by_chunks_n um_segments_10_segm t_focus_9	0.0017
First_location_of_maximum	0.0000	CMO	0.0107	RSI	0.0013
High	0.0000	RSI	0.0098	TRIX	0.0004
Low	0.0000	ROC_6M	0.0092	Index_mass_quantile_q_0. 7	0.0003
Close	0.0000	Mean_change	0.0074	Augmented_dickey_fuller_ attr_“teststat”_autolag_“AI C”	0.0003
Ar_coefficient_coeff_5_k_10	0.0000	Time_reversal_asymmetry_ statistic_lag_1	0.0046	Ar_coefficient_coeff_0_k_10	0.0003
Index_mass_quantile_q_0.9	0.0000	ROC_12M	0.0045	Last_location_of_maximum	0.0002
Variation_coefficient	0.0000	Time_reversal_asymmetry_ statistic_lag_2	0.0043	Lempel_ziv_complexity_b ins_10	0.0002
Fft_aggregated_aggtype_“variance”	0.0000	Time_reversal_asymmetry_ statistic_lag_3	0.0036	Augmented_dickey_fuller_ attr_“pvalue”_autolag_“AI C”	0.0002
Fft_aggregated_aggtype_“centroid”	0.0000	Force_Inx	0.0018	CCI	0.0002
Top 10 features with negative impact	Mean Shap value	Top 10 features with negative impact	Mean Shap value	Top 10 features with negative impact	Mean Shap value
TRIX	-0.0014	Mean_Std_Year	0.0000	Minus_DI	-0.0001
Time_reversal_asymmetry_statistic_lag_3	-0.0029	Count_below_mean	0.0000	CMO	-0.0001
Time_reversal_asymmetry_statistic_lag_2	-0.0037	BDI	0.0000	BB	-0.0001
Time_reversal_asymmetry_statistic_lag_1	-0.0041	Ar_coefficient_coeff_5_k_10	0.0000	ROC_12M	-0.0002
ROC_12M	-0.0043	Close	0.0000	Time_reversal_asymmetry_statistic_lag_1	-0.0005
Mean_change	-0.0064	Low	0.0000	Time_reversal_asymmetry_statistic_lag_2	-0.0006
ROC_6M	-0.0083	High	0.0000	Time_reversal_asymmetry_statistic_lag_3	-0.0007
CMO	-0.0106	Ultimate	-0.0001	Force_Inx	-0.0008
RSI	-0.0112	First_location_of_maximum	-0.0001	ROC_6M	-0.0009
Energy_ratio_by_chunks_n um_segments_10_segm t_focus_9	-0.0237	Last_location_of_maximum	-0.0002	Mean_change	-0.0010

Table 20: Top 18 features with the 8 most positive and 10 most negative impact on the FX Random Forest model's predictions.

"Bullish" regime		"Bearish" regime		"Other" regime	
Top 8 features with positive impact	Mean Shap value	Top 8 features with positive impact	Mean Shap value	Top 8 features with positive impact	Mean Shap value
Ar_coefficient_coeff_6_k_10	0.0018	ROC_6M	0.0266	ROC_12M	0.0060
BDI	0.0005	ROC_12M	0.0155	Mean_change	0.0052
Lempel_ziv_complexity_chunks_10	0.0002	Mean_change	0.0120	CMO	0.0036
CCI	0.0001	Time_reversal_asymmetry_statistic_lag_3	0.0032	RSI	0.0029
Last_location_of_maximum	0.0001	TRIX	0.0026	Augmented_dickey_fuller_attr_"pvalue"_autolag_"AIC"	0.0028
BB	0.0001	10Y bond yield	0.0018	TRIX	0.0017
High	0.0000	Energy_ratio_by_chunks_numb_segments_10_segments_focus_9	0.0016	ROC_6M	0.0010
Ar_coefficient_coeff_7_k_10	0.0000	Time_reversal_asymmetry_statistic_lag_2	0.0013	BB	0.0005
Top 10 features with positive impact		Top 10 features with positive impact	Mean Shap value	Top 10 features with positive impact	Mean Shap value
Last_location_of_minimum	-0.0007	Ultimate	-0.0001	Last_location_of_maximum	0.0001
Energy_ratio_by_chunks_numb_segments_10_segments_focus_9	-0.0009	BDI	-0.0001	Ar_coefficient_coeff_7_k_10	0.0000
RSI	-0.0012	Lempel_ziv_complexity_chunks_10	-0.0002	Lempel_ziv_complexity_chunks_10	0.0000
10Y bond yield	-0.0016	Last_location_of_maximum	-0.0002	Time_reversal_asymmetry_statistic_lag_1	-0.0002
Time_reversal_asymmetry_statistic_lag_3	-0.0019	High	-0.0003	10Y bond yield	-0.0003
Augmented_dickey_fuller_attr_"pvalue"_autolag_"AIC"	-0.0041	CCI	-0.0005	BDI	-0.0004
TRIX	-0.0043	BB	-0.0006	Ar_coefficient_coeff_6_k_10	-0.0006
Mean_change	-0.0171	Ar_coefficient_coeff_6_k_10	-0.0012	Energy_ratio_by_chunks_numb_segments_10_segments_focus_9	-0.0007
ROC_12M	-0.0215	RSI	-0.0017	Time_reversal_asymmetry_statistic_lag_2	-0.0009
ROC_6M	-0.0276	CMO	-0.0031	Time_reversal_asymmetry_statistic_lag_3	-0.0013

From Figures 6-8 it can be noted that all three models are driven by a similar set of features, although, notably, the FX Random Forest model, unlike the other models, is not mainly driven by the energy ratio by chunks<sup>58</sup> feature. The largest impact on all models' probabilities is from technical momentum variables, instead of macroeconomic variables, which is in line with the data frequency (daily); in the

<sup>58</sup> The variant of energy ratio by chunks seen in Figures 6-8 divides price data into ten different, non-shuffled chunks and calculates a sum of squares of prices in the ninth chunk over a sum of squares of the price data used to calculate this feature (see Table 11 for detailed explanation of this variable).

industry, it is commonly known that technical features will have a larger predictive power for higher frequency data, while macroeconomic features conversely tend to predict a dependent variable more accurately when the data is of lower frequency, such as monthly. This is because macroeconomic variables tend to change more slowly than technical, as well as usually lagging current market conditions.

Tables 18-20 confirm the contrarian behaviour of the models, particularly in the Commodities Random Forest model; when predicting the “bullish” regimes, absolute Shapley values are larger in the top 10 negative features, whilst for predicting the “bearish” regimes, absolute Shapley values are higher in the top 10 positive features. In addition, it is usually the case that the top 10 negative features in the “bullish” regime are mirrored in the top 10 positive features in the “bearish” regime. As noted above, the models are mainly driven by momentum variables. When in a “bullish” regime, a low positive price momentum (expressed by, for instance, rate of change) contributes the most to the predictions, which in turn indicates that when markets are exhausted after a price rally, there is a higher probability of a subsequent market downturn, and that thus the underlying asset should be shorted. On the contrary, in the “bearish” regime, once price momentum picks up, there is a higher probability for the price to rally further, thus the underlying asset should be bought. This can usually be seen near a market trough when the price rebounds, sometimes sharply, spurring an action to enter a long position within the asset.

The next section looks first at each Random Forest model’s performance with regards to classification and financial metrics, before comparing the models’ trading performance to selected benchmarks.

#### 4.3.3 Model evaluation

##### *The performance of the three Random Forest models*

Table 21 to follow lists out-of-sample cost-adjusted Sortino ratios (Equation 45) for each Random Forest model, as well as adjusted Sharpe ratios (Equation 22), and

cumulative geometric returns (Equation 48). In addition, Table 22 shows MCC scores for each model and regime label.

*Table 21: Random Forest model financial evaluation metrics. Note these metrics are cost-adjusted, and all data series terminate on 29/04/2022.*

Model	Sortino ratio (ann.)	Adjusted Sharpe ratio (ann.)	Cumulative geometric returns	Start date
<b>Equity Random Forest model</b>	24.99	1.44	284.16%	06/03/2018
<b>Commodity Random Forest model</b>	12.66	1.22	614.09%	19/12/2017
<b>FX Random Forest model</b>	4.72	0.29	58.77%	20-04-2018

*Table 22: MCC scores for each Random Forest model.*

Model	MCC "bullish"	MCC "bearish"	MCC avg. "bullish" and "bearish"	MCC "other"	MCC all labels (Equation 46)
<b>Equity Random Forest model</b>	0.4809	0.3768	0.4289	0.1455	0.3538
<b>Commodity Random Forest model</b>	0.4260	0.4191	0.4256	0.0632	0.3524
<b>FX Random Forest model</b>	0.5413	0.3295	0.4354	0.0552	0.3785

It is immediately apparent both that the binary MCCs for "bullish" and "bearish" classes are very high, in the context of financial prediction, while the binary MCCs for the "other" class are substantially lower. However, this latter is not a large problem as the "other" signal is used only to close a position, and hence a misclassification to "other" would represent only a loss due to the premature closing of a position, not a potentially much larger loss due to having taken the wrong position.

**Equity Random Forest model.** The Equity Random Forest model generated significant cumulative geometric returns over the holdout sample whilst it also achieved a very high Sortino ratio. The Sortino ratio includes only negative returns in its standard deviation calculations, while the adjusted Sharpe ratio uses all returns to compute volatility. A high Sortino ratio, but lower Sharpe ratio (though still above 1, which is welcomed in the industry), as here, indicates a high volatility of positive returns. The MCC scores, particularly for “bullish” and “bearish” labels, show clear evidence that the Equity model does not “guess” regimes but instead generates accurate predictions that the trading strategy acts upon.

**Commodity Random Forest model.** The Commodity model is the one that has the best trading performance; it generated over 600% during the test period, as can be seen in Table 21. However, its adjusted Sharpe and Sortino ratios were lower than those for the Equity model; this suggests a significantly higher volatility for commodity assets, which is in line with industry observations.

**FX Random Forest model.** The FX model generated the lowest cumulative returns (though its Sortino and Sharpe ratios were even so quite high), though it had the highest MCC scores, both when all labels were considered, as well as with regards to the average of the “bullish” and “bearish” labels only. This suggests that FX price movements are relatively small, so the potential for profit in this asset class is less. FX pairs of developed nations are not necessarily characterised by explosive performance; on the contrary, these assets tend to be mean-reverting over time.

The next section will compare each Random Forest model’s trading performance to the that of the selected benchmark models, as well as to index benchmarks from Table 10, in terms of cumulative performance and financial metrics.

*Trading performance compared to benchmark models*

Table 23 shows the financial performances of each benchmark and how they compare to the Random Forest models. After a brief description of the results, several figures to follow will show the trading performances in terms of cumulative geometric returns. Note that the test period over which the statistics in Table 23 were derived is the same as in Table 21.

*Table 23: Trading performance comparison between tested models. Note: the financial metrics are cost-adjusted.*

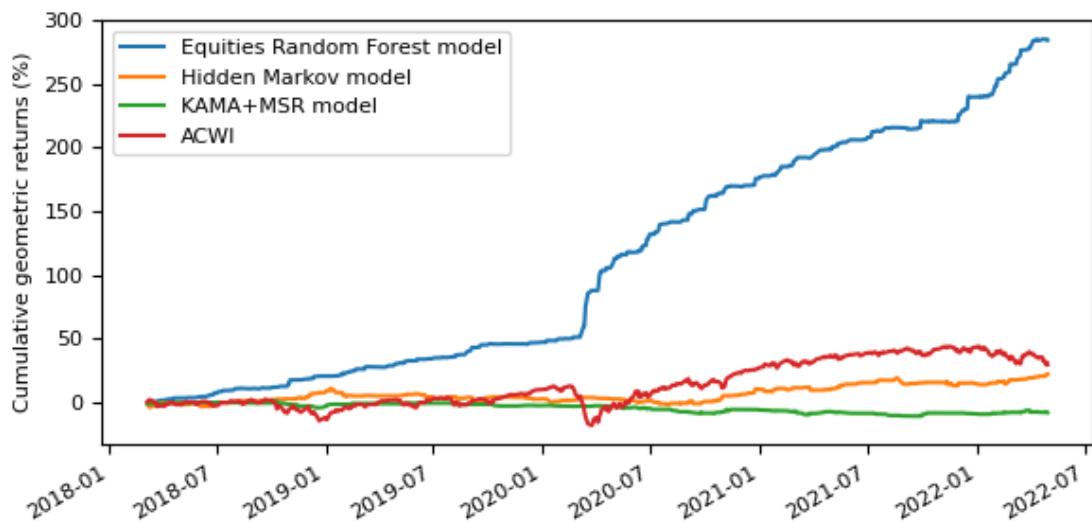
Asset class	Index benchmark	Model	Sortino ratio (ann.)	Adjusted Sharpe ratio (ann.)	Information ratio (ann.) vs. index benchmark
Equities	MSCI ACWI	Equity Random Forest model	<b>24.99</b>	<b>1.44</b>	<b>1.46</b>
		Hidden Markov model	0.32	0.005	-0.11
		KAMA+MSR	-0.29	-0.0004	-0.51
Commodities	Bloomberg Commodity Index	Commodity Random Forest model	<b>12.66</b>	<b>1.22</b>	<b>1.93</b>
		Hidden Markov model	-0.47	-0.002	-1.27
		KAMA+MSR	0.24	0.002	-0.42
Foreign Exchange	BBDXY Index	FX Random Forest model	<b>4.72</b>	<b>0.29</b>	<b>1.45</b>
		Hidden Markov model	-0.37	-0.0001	-0.88
		KAMA+MSR	0.17	0.1	-0.28

The Random Forest models have a far better out of sample performance over HMM and KAMA+MSR models. In addition, only the Random Forest models outperformed the index benchmarks, and by a significant margin, each Random

Forest model's annualised information ratio being substantially above 1, which is considered desirable in the industry.

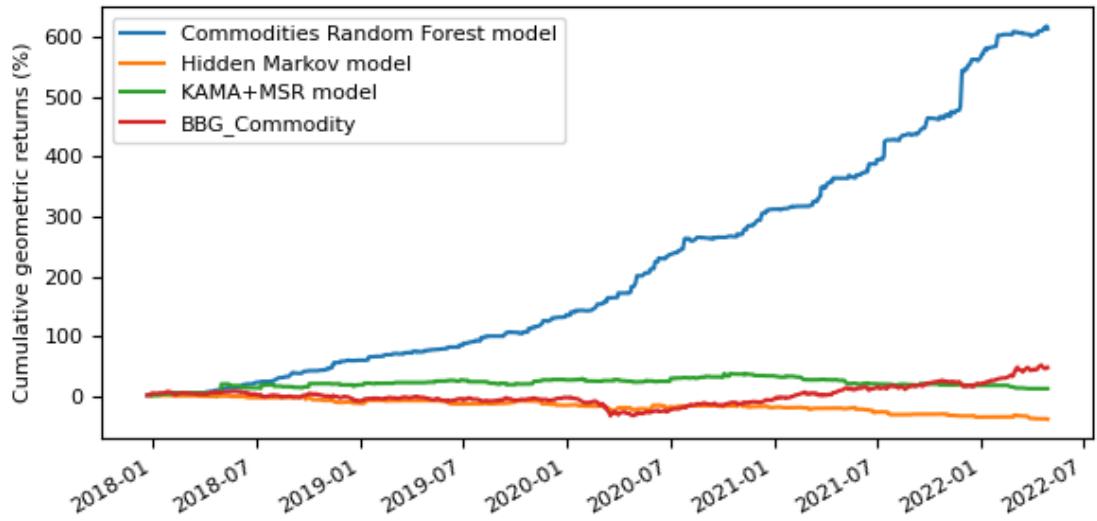
Figures 9, 10, and 11 show the cumulative geometric returns of each asset class model in Table 23, as well as those of its appropriate index benchmark. Note that the index benchmark does not present a geometric but rather a cumulative sum of returns, as it is simply bought and held over the same timespan as the models from Table 21. In addition, the index benchmark is cost-free.

*Figure 9: Cumulative geometric returns of the Equity, HMM, and KAMA+MSR models compared to a long-only position in MSCI ACWI (index benchmark). Note: the performance is cost-adjusted, apart from ACWI.*



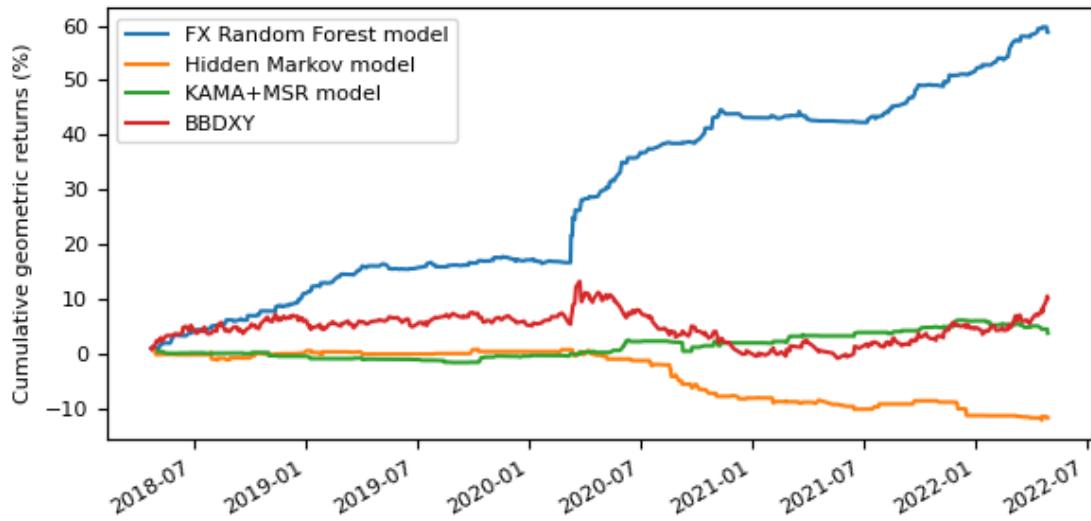
**Equity Random Forest model.** The Equity model (Figure 9) consistently outperformed all selected benchmarks. Its hedging abilities can be particularly noted; during both the Covid plunge of 2020 and the later 2022 downturn, the Equity model continued generating higher returns than the benchmarks, which shows it has been able to either short the markets in a timely fashion or avoid recommending a long position (by predicting the “other” regime during those times). The KAMA+MSR model, on the other hand, has consistently underperformed the index benchmark, while the hidden Markov model has only recently begun catching up with the market.

*Figure 10: Cumulative geometric returns of the Commodity, HMM, and KAMA+MSR models compared to a long-only position in the Bloomberg Commodity Index (the index benchmark). Note: the performance is cost-adjusted, apart from BBG\_Commodity.*



**Commodity Random Forest model.** Similarly to the Equity Random Forest model, the Commodity model (Figure 10) has consistently outperformed its benchmarks. The commodity market fall during the Covid plunge in 2020 did not negatively impact the model's performance, as its returns continued growing during that period. The only visible underperformance vs. the Bloomberg Commodity Index can be noticed in the beginning of 2022 where the benchmark index rose, while the model's returns failed to do so; however, the model's returns rebounded later in the year. Neither KAMA+MSR nor HMM outperformed the benchmark index throughout the whole period, although up until the middle of 2021 KAMA+MSR consistently outperformed the Bloomberg Commodity Index.

Figure 11: Cumulative geometric returns of the FX, HMM, and KAMA+MSR models compared to a long-only position in the BBDXY Index (the index benchmark). Note: the performance is cost-adjusted, apart from BBDXY.



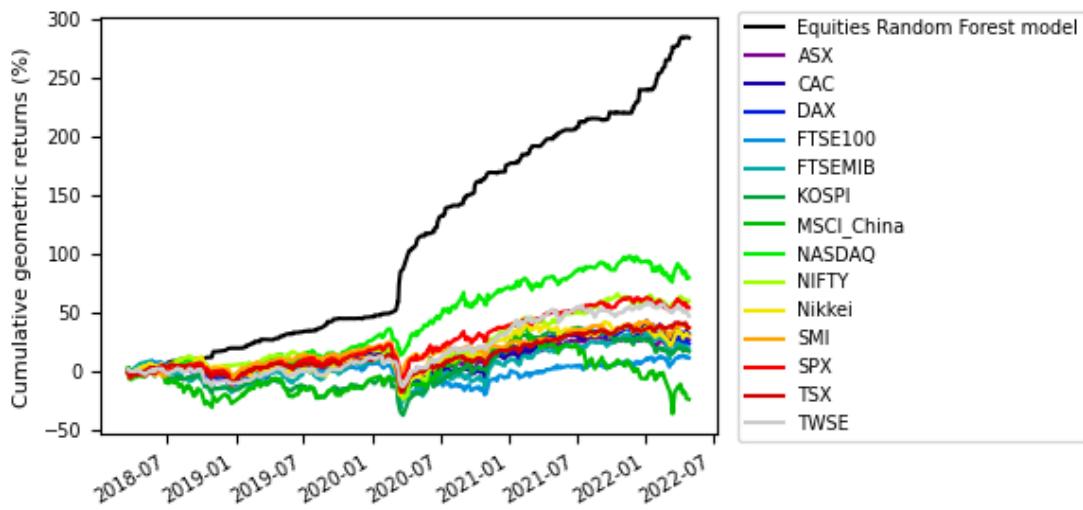
**FX Random Forest model.** As can be seen in Figure 11, for the first two years the FX model, even though it managed to outperform BBDXY, did not perform spectacularly. This changed only after the Covid meltdown in 2020, when the FX model began to be able to predict regimes in foreign exchange rates more accurately, apart from 2021, during which the model's trading performance started to slightly deteriorate. The HMM did worst of the models considered; the performance of the KAMA+MSR model from Chapter 3 began to improve in 2020, although in the end it underperformed the benchmark index.

The next section will compare the Random Forest models' performances to each component within each asset class (as listed in Table 10 of Section 4.2.1).

#### *Trading performance comparison vs. long-only positions in assets*

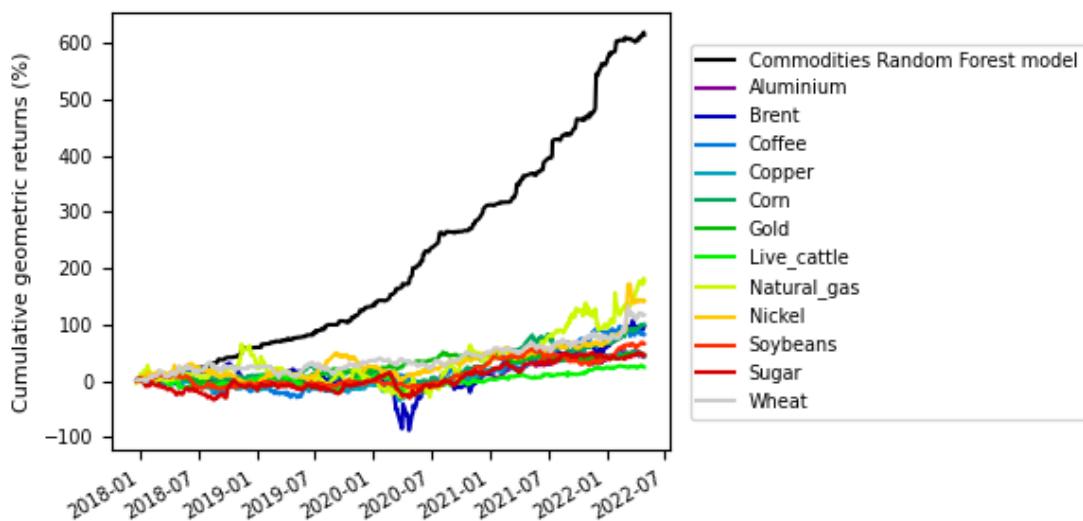
Figures 12, 13, and 14 show the trading performance of the Random Forest models relative to the long-only, cost-free position in each asset within its class.

*Figure 12: Performance of the Equity Random Forest model compared to that from holding a long-only position in each equity asset from Table 10.*



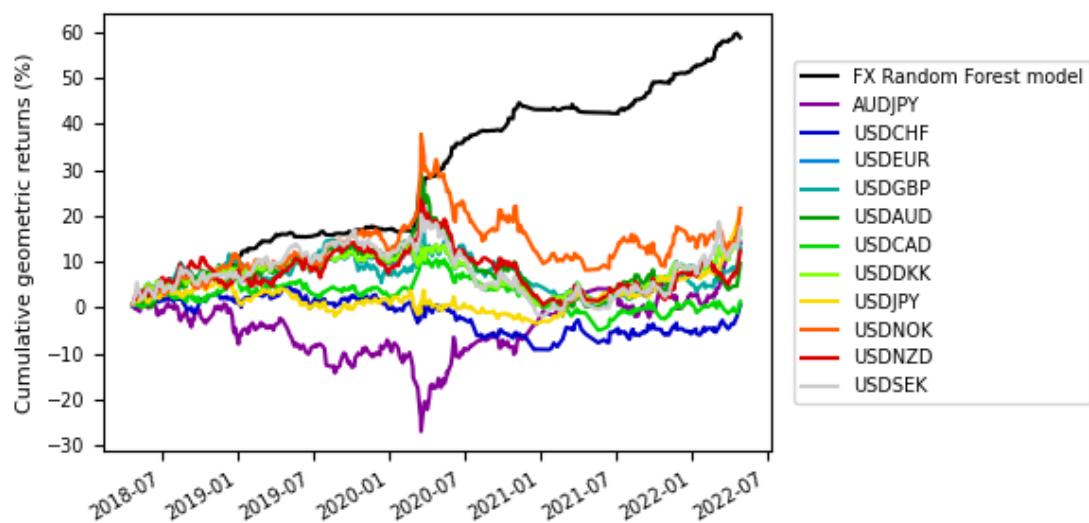
**Equity Random Forest model vs. component assets.** Clearly, from Figure 12, the Equity model has outperformed cost-free, long-only positions in each equity asset. Interestingly, when each equity asset began to fall at the beginning of 2022, the Equity model's performance improved; this indicates accurate timing of short positions during the market downturn of 2022.

*Figure 13: Performance of the Commodity Random Forest model compared to that from holding a long-only position in each commodity from Table 10.*



**Commodity Random Forest model vs. component assets.** The Commodity model, as can be seen in Figure 13, has also substantially outperformed every individual asset in its category, except for a brief period during winter 2018-2019 when the price of natural gas increased. In addition, the Random Forest model has managed to predict the rise in commodity prices since the beginning of 2022, though for a short period of time in early 2022 underperforming natural gas and nickel, whose prices soared while the Commodity model's performance marginally decreased, to rebound soon after.

Figure 14: Performance of the FX Random Forest model compared to that from holding a long-only position in each FX pair from Table 10.



**FX Random Forest model vs. component assets.** The FX model initially struggled to outperform some of the FX pairs, such as USDNZD, USDEUR, USDSEK, USDNOK, and USDAUD. That changed after the Covid recession in early 2020, when the model's performance significantly improved, while the performance of each FX pair, apart from AUDJPY, deteriorated. However, the model did not immediately capture the rally in the USD (and hence in individual FX pairs) toward the end of the test period, in April 2022.

### *Asset allocation based on the predictions of the Random Forest model*

To conclude, Figures 15 to 21 examine asset allocation within the 15% testing set, based on each Random Forest model's predictions. This further clarifies the contrarian behaviour of the models; when the "bearish" signal is generated, the asset should be bought, while after the "bullish" signal, the underlying asset should be shorted. The accumulated positions are shown against the index benchmarks from Table 10 in Section 4.2.1 on data (MSCI ACWI, Bloomberg Commodity Index, and Bloomberg DXY Index), which, even though they do not fully represent each underlying asset component's performance, still mimic the overall behaviour of a selected asset class during the holdout period.

*Figure 15: Equity assets recommended to be bought by the Equity model. Note: "accumulated bearish positions" means assets which received a "buying" signal from the model. The black line shows MSCI ACWI's – the equity index benchmark – performance.*

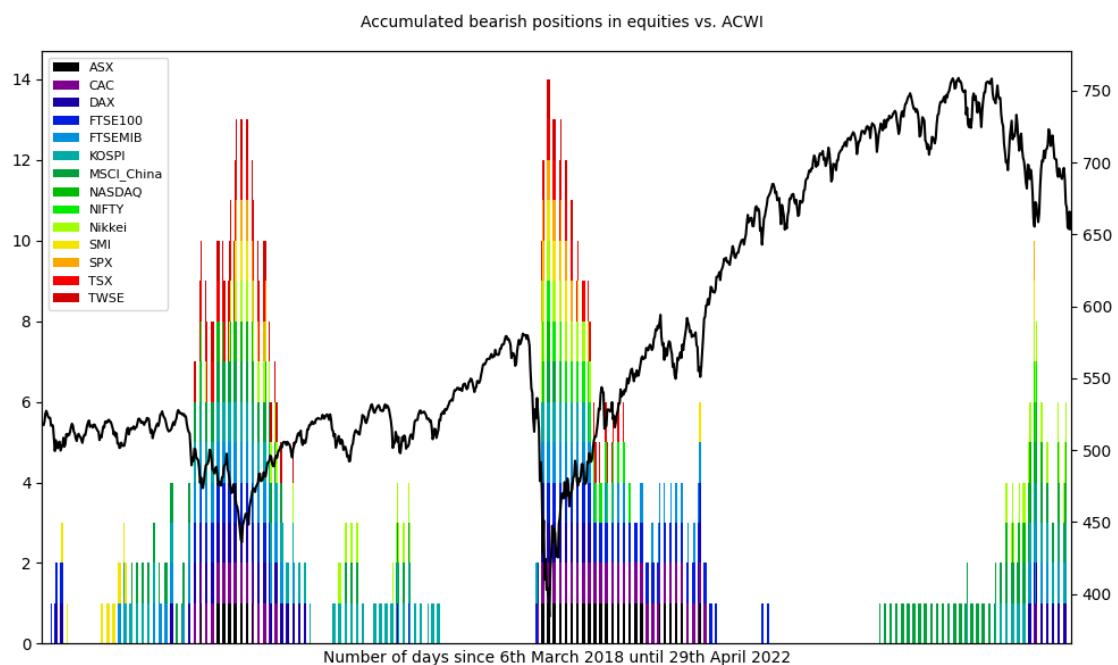
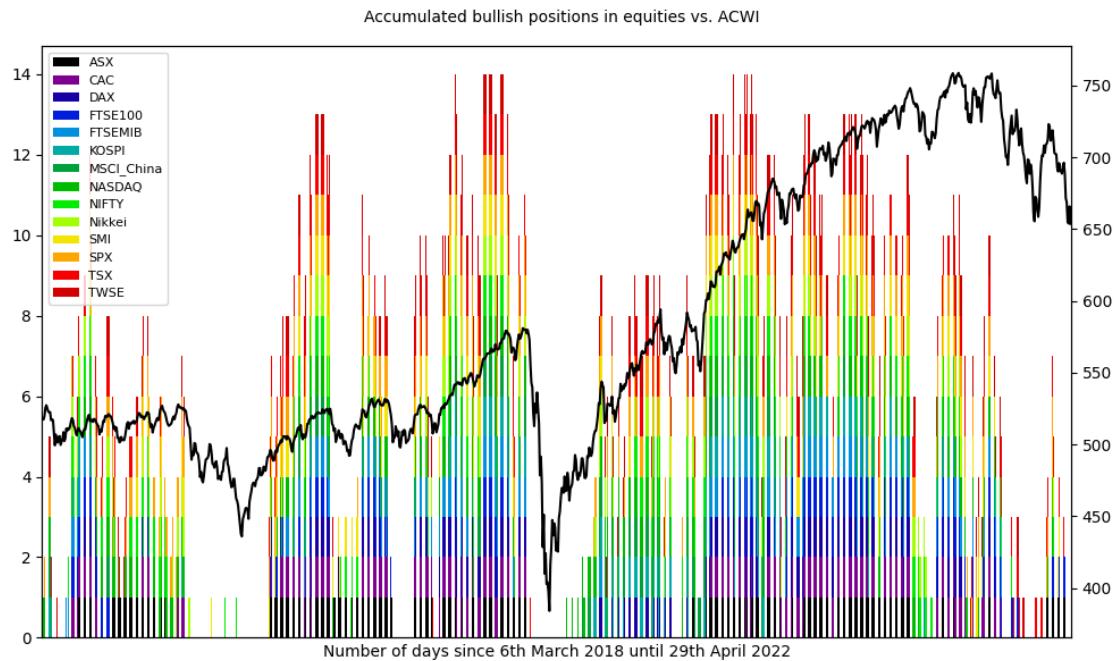


Figure 16: Equity assets recommended to be shorted by the Equity model. Note: "accumulated bullish positions" means assets which received a "shorting" signal from the model. The black line shows MSCI ACWI's – the equity index benchmark – performance.



**Equity Random Forest model.** Figures 15 and 16 clearly reveal the contrarian behaviour of the Equity model. Note that during the troughs of late 2018 and the Covid plunge in 2020, almost all the underlying equities were predicted to be bought. Simultaneously, right before each dip, such as in 2018, later in 2019, 2020, or beginning of 2022, multiple equity assets received a "shorting" signal.

Figure 17: Commodities recommended to be bought by the Commodity model. Note: "accumulated bearish positions" means assets which received a "buying" signal from the model. The black line shows the Bloomberg Commodity Index's – the commodity index benchmark – performance.

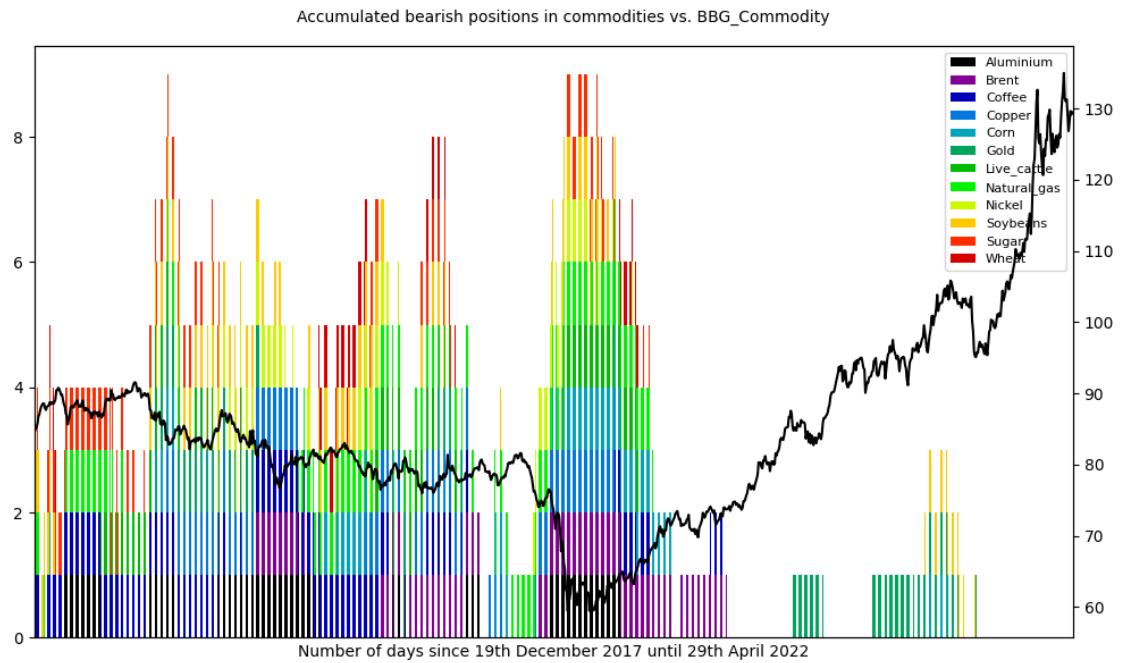
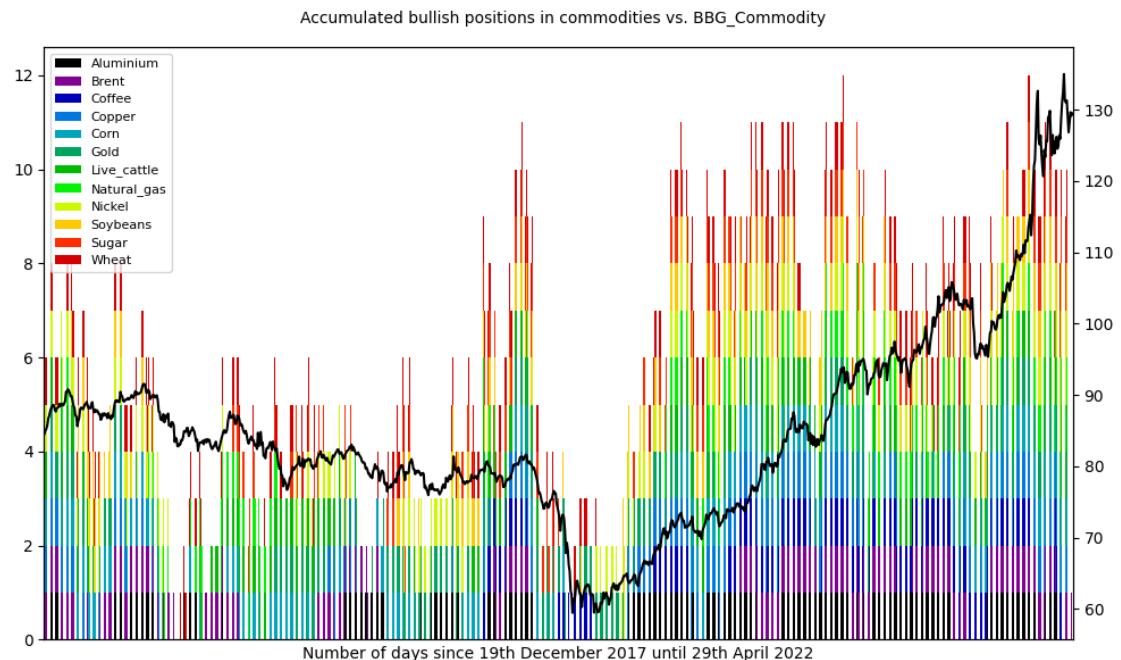


Figure 18: Commodities recommended to be shorted by the Commodity model. Note: "accumulated bullish positions" means assets which received a "shorting" signal from the model. The black line shows the Bloomberg Commodity Index's – the commodity index benchmark – performance.



**Commodity Random Forest model.** Similar asset allocation behaviour to the Equity Random Forest model can be seen in Figures 17 and 18 showing

commodities allocation based on the Commodity model's signals. Particularly noticeably, right before the Covid recession the model had recommended shorting all the commodity futures (Figure 18), while at the recession's trough the model sent a "buy-back" signal (Figure 17). Since the Covid plunge in 2020, the Commodity model's performance has been mainly derived from assuming short positions close to small dips; however, due to the optimal diversification of the Bloomberg Commodity Index (the black line in Figures 17 and 18), this may not be fully visible in the figures shown—in fact, the major driver of the benchmark index, Brent crude oil, has not been often shorted since the Covid recession, which is in line with the benchmark index performance.

*Figure 19: FX pairs recommended to be bought by the FX model. Note: "accumulated bearish positions" means FX pairs which received a "buying" signal from the model. The black line shows the Bloomberg DXY Index's – the FX index benchmark – performance.*

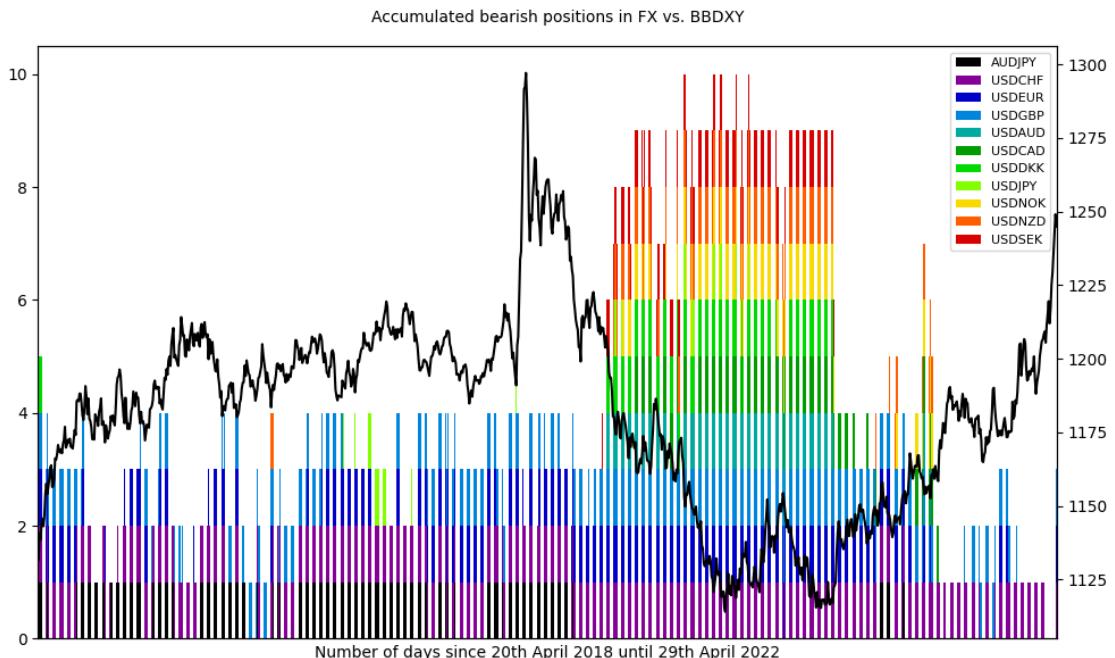
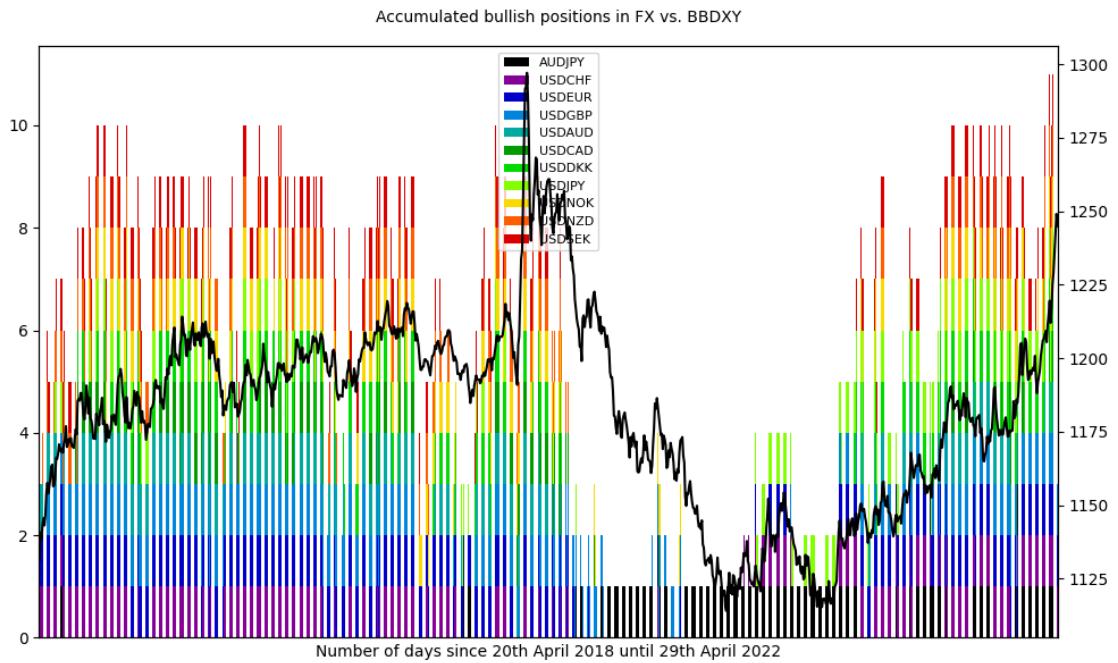


Figure 20: FX pairs recommended to be shorted by the FX model. Note: "accumulated bullish positions" means FX pairs which received a "shorting" signal from the model. The black line shows the Bloomberg DXY Index's – the FX index benchmark – performance.



**FX Random Forest model.** Finally, Figures 19 and 20 present the long and short positions, respectively, in G10 currencies. When the US dollar (represented by the Bloomberg DXY Index) was weakening, from the Covid recession in 2020 until the beginning of 2022, the FX model began recommending buying the quote currencies (Figure 19); on the other hand, right before and during the benchmark index peak in 2020, the FX model sent shorting signals for the quote currencies (Figure 20). Interestingly, the allocation of AUDJPY seems not to follow the general allocation pattern; the FX model recommended buying it at the peak and shorting it at the trough of the Bloomberg DXY Index. This could be related to the fact that AUDJPY usually serves as a contrarian indicator in the industry—by shorting JPY in AUDJPY investors bet on better times to come for the global economy, while buying JPY in AUDJPY is a signal for hedging against possible tumultuous markets in the future.

#### **4.3.4 To short, or not to short, that is the question**

The previous section described and discussed the results of the three Random Forest models, one for each of the asset classes considered, in detail. Clearly, the models, after learning to predict regimes that in Chapter 3 could only be detected after their onset, have been able to produce buying, selling, and shorting signals that turn out frequently to generate substantial profits, net of costs. But a question remains as to the source of these models' success: is it the unusual accuracy of the signals produced, or the type of positioning strategy then adopted, or a mix of both? In Figures 15-21 of the previous section, it can be clearly seen that, even though the models have been able to predict regime switches accurately, they have been more often producing shorting rather than buying signals. In addition, the short signals seem to be of lesser duration than the long signals. This suggests that the long positions may be produced on the basis of longer-term macroeconomic factors, while the short positions may be in contrast triggered by local drops, hence their frequency tends to be higher; this is a typical "scalping" behaviour, as it is known in the industry, which is characterised by smaller but more frequent gains generated over a short term. This can be particularly seen in case of the Equity and Commodity Random Forest models (Figures 16 and 18), where the majority of profits generated after the Covid recession have been attributable to short positions. Such frequent short positioning may be problematic in real time, as often shorting is not that simple as buying an asset, as well as being sometimes constrained by institutions providing brokerage services on grounds of financial risk or excluded from consideration by investors for ethical reasons. A solution to these various issues is to buy put options; however, such options incur additional costs, as premia must be paid for the right to sell short the underlying asset.

Eliminating the shorting label could possibly make the models more appealing to institutions which implement long-only strategies, although they might be considerably less financially successful if the shorting class was absent, as a large chunk of profitable trades would simply be gone. On the other hand, the supporting scoring metric, MCC (see Table 22 in the previous section) for the "bearish" label (noting that the models are contrarian), indicates that the models were predicting

these regimes better than random. It is thus possible that the models might be trained to assume long-only positions with success. However, such a binary model would not be optimal. Aside from adopting a short position, the shorting signal could be used in a different way, such as for pair trading, mixing asset classes, or creating an early-warning system. The first option, pair trading, is based on using the shorting signal of one asset to buy another asset whose returns are negatively correlated with the shorted asset's returns, though this is problematic in case of major market crashes, such as the Covid plunge, as often correlations between asset returns tend to increase during such events. The second option, mixing asset classes, could increase diversification; for instance, if the Equity Random Forest model sends shorting signals for multiple assets, selling them and buying bonds instead may serve as a solid protector against a market crash, as well as being profitable due to increasing bond prices. Finally, early-warning systems have a number of potential uses; for example, whenever short signals accumulate significantly, such as before the Covid plunge in 2020, institutions can use this information to launch scenarios that would protect portfolios, while investment banks could send a warning to their clients in a research note.

## 4.4 Conclusions

This chapter focused on predicting ex ante the detected regime categories from Chapter 3, using Random Forest. Three major asset classes were used for this experiment: equities, commodities, and foreign exchange, to increase the robustness of the experiment's conclusions. The results showed the success of the described methodology, as each of the Random Forest models, based on its generated signals, was able to outperform the selected benchmarks, as well as generate substantial risk-adjusted returns, net of costs. The Commodity Random Forest model had the highest cumulative geometric returns, while the Equity Random Forest model achieved the highest annualised Sortino and adjusted Sharpe ratios.

The major finding of this chapter is that the Random Forest models provide signals best interpreted as contrarian—it was discovered that the higher the

probability of a “bullish” class, the more probable it was that the asset price would actually fall, while the higher the probability of the “bearish” class, the more probable it was that the asset price would rise. This was clear in Figures 15-21 in the section on model evaluation (4.3.3), which show asset allocation based on the generated signals from each Random Forest model. In these figures, a large accumulation of “bullish” signals indicated a recommendation to short the underlying assets, such as immediately before the Covid plunge in 2020, whereas a substantial accumulation of “bearish” signals generated by the Random Forest models recommended conversely entering a long position within the underlying assets, such as during the trough of the Covid recession.

However, the high profits generated by acting upon signals from the Random Forest models come at a certain cost. As Figures 16, 18, and 20 show, a large chunk of profitable trades has been attributable to frequent short positions. Even though frequent shorting is not impossible, it could be problematical, for reasons described earlier. However, trying to mix asset classes (such as equities with bonds), or avoiding short positions by turning accumulated short signals into an early-warning system for research and trading, could overcome these concerns and make the Random Forest models highly useful in the industry.

## **5. Combining regime switching predictive framework with model predictive control for multi-period portfolio optimisation**

Chapter 4 used and enhanced the KAMA+MSR framework from Chapter 3 in order to generate labels to be predicted by a machine learning model. The set of three Random Forest models predicted ex-ante “bullish” and “bearish” regimes within equity, commodity, and FX asset classes, achieving not only high classification scores out of sample, but also high Sortino ratios, indicating solid financial performance when the models are applied to trading. However, one caveat associated with the work described in Chapter 4 was that the model’s excellent trading performance was associated with frequent shorting positions, which could be problematic for many institutions that are disallowed from doing using these. In addition, the model could be enhanced with more dynamic weight allocation (instead of using mean weights for each asset), as well as more complex cost model (Chapter 4 assumed fixed costs no matter the liquidity and volatility of the traded assets). Both the shorting issue and the introduction of more advanced asset allocation mechanisms can be solved by the portfolio optimisation algorithm presented in this chapter, which is focused on a long-only strategy, as well as being capable of dynamically shifting asset weights depending on the underlying asset return estimates in order to construct high Sortino portfolios under a realistic cost model.

This chapter will focus on applying a portfolio optimisation technique already mentioned in Chapter 2, the model predictive control (MPC), which can incorporate multiple factors, such as time-varying return estimates, transaction costs constraints, risk constraints, and the trade-off between short-term versus long-term asset holding, all over multiple horizons, owing to the concept of multi-period optimisation. It will be shown that the model predictive control is a promising wrapper for a regime-switching predictive framework, as it is able to use regime predictions as return estimates to establish realistic regime-robust asset allocation among different asset classes. To achieve the desired result, MPC will work with Chapter 4 data and models, namely equity and commodity models,

leaving FX out of the research scope due to its significant dissimilarity to equities and commodities.

Chapter 5 is organised as follows. First, relevant technical background and related work are presented. This is followed by a methodology section and a short discussion of the data used in this experiment. Finally, the chapter ends with a presentation and discussion of the results of this chapter.

## 5.1 Background and related work

### 5.1.1 Multi-period optimisation

Before delving into model predictive control, it is important to explain the concept of multi-period optimisation (MPO) that this chapter will use. While single-period optimisation (SPO) focuses on allocating asset weights based on a selected framework, such as mean-variance, for a single portfolio allocation over a specific horizon, i.e., next week, month, or a quarter, multi-period optimisation goes one step further and optimises portfolio based on a sequence of trades to be executed over multiple periods. To clarify, at each time period  $t$ , MPO uses available information, such as assets returns, risk, and, for example, trading costs, to optimise the assets weights for the planning horizon  $H$ , where  $t \in \{1, 2, \dots, H\}$ ; SPO can be thus seen as a subset of MPO where  $H = 1$ . Since SPO is focused on only one period ahead, it is more suitable for stationary mean-variance assumptions, whilst, as was already mentioned throughout this thesis, financial time series in practice are non-stationary with ever-changing mean-variance assumptions, which is better captured by MPO (Oprisor & Kwon, 2020). In addition, MPO addresses multiple factors that single-period optimisation fails to incorporate, such as the change of asset return dynamics, and the trade-off between short-term and long-term benefits. Also, as was indicated in Chapter 2, MPO has shown a lot of promise for portfolio optimisation, in terms of application to a regime-switching model output, as well as beating single-period optimisation techniques (such as the classic

portfolios described in Section 2.1.2). Finally, MPO takes the future into consideration (e.g., via return estimates) when planning to execute the trades, whilst single-period optimisation encompasses only backward-looking data (Li, Uysal, & Mulvey, 2022).

MPO is not a novel technique, as its roots can be found in the 1970s (Samuelson, 1975); however, only after the significant increase in computing power has it regained attention, as it has since then been possible to thoroughly test MPO in a multi-asset environment under many constraints, such as transaction costs, risk aversion, and holding costs (Boyd, et al., 2017). A number of algorithms have been proposed to solve the multi-period optimisation problem, such as the already mentioned model predictive control (e.g., (Herzog, Dondi, & Geering, 2007), (Boyd, et al., 2017)), particle swarm optimisation (Sun, Fang, Wu, Lai, & Xu, 2011), or, more recently, a genetic algorithm (Babazadeh & Esfahanipour, 2019); however, due to computational efficiency and a strong support in the literature (mentioned in Section 2.3.3), this chapter will focus solely on the MPC solution to solve the MPO problem under multiple constraints.

### 5.1.2 Model predictive control

Even though there are multiple variations of MPC, this chapter focuses solely on the mean-variance approach, a traditional method of selecting portfolio allocations based on a risk-return trade-off which has also been recently found to be robust in the dynamic, multi-period setting (van Staden, Dang, & Forsyth, 2021). Other MPC variations, based on risk parity or minimum variance, can be used to address the multi-period portfolio optimisation problem; however, they have not been investigated further here as it was considered that they were unnecessarily complex.

Assuming the optimised portfolio consists of  $n$  assets with an additional cash asset ( $n + 1$ ) tracked over a finite time horizon  $T$  and divided into discrete periods (e.g., days), such that  $t \in \{1, 2, \dots, T\}$ , let  $h_t \in \mathbb{R}^{n+1}$  denote the portfolio at the beginning of time  $t$ , where  $h_{t,i}$  is the dollar value of asset  $i$ . When  $h_{t,i} > 0$ , for  $i \in$

$\{1, 2, \dots, n\}$ , the portfolio is long-only, which is the portfolio of focus of this chapter, while  $h_{t,i} < 0$  implies a short position within an asset  $i$ ; in addition, if  $h_{t,n+1} = 0$ , where  $n + 1$  stands for the cash asset, the portfolio is fully invested at time  $t$ , implying no cash position whatsoever. The total dollar value of the portfolio  $h_t$  can be written as  $V_t = \mathbf{1}^T h_t$ . The values of  $h_t$  and  $V_t$  can be used to compute the weights  $w_t \in \mathbb{R}^{n+1}$  of each asset  $i$  within the portfolio  $h_t$ , such that  $w_t = \frac{h_t}{V_t}$ , where  $\mathbf{1}^T w_t = 1$ .

The construction of the MPC model here will follow that of (Boyd, et al., 2017). Assuming a sequence of allocations  $\{w_{t+1}, w_{t+2}, w_{t+3}, \dots, w_{t+H}\}$  executed at time  $t$  over a planning horizon  $H$ , where each allocation (set of portfolio weights) is an  $n+1$ -dimensional vector, the objective of MPC with a mean-variance term (namely, the maximisation of returns by the minimisation of risk and costs) can be reduced to the maximisation of the utility function

$$\sum_{\tau=t+1}^{t+H} \hat{\mathbf{r}}_\tau^T \mathbf{w}_\tau - \gamma^{sigma} \mathbf{w}_\tau^T \hat{K}_\tau \mathbf{w}_\tau - \gamma^{trade} \sum_{i=1}^n \widehat{TC}(\Delta w_{\tau,i}) - \gamma^{hold} \sum_{i=1}^n \widehat{HC}(\Delta w_{\tau,i}), \quad (50)$$

subject to

$$\tau = t + 1, \dots, t + H, w_\tau \geq 0, \mathbf{1}^T w_\tau = 1, \Delta w_{\tau,i} \triangleq w_{\tau,i} - w_{\tau-1,i},$$

where  $\hat{\mathbf{r}}_\tau$  and  $\hat{K}_\tau$  are returns and covariance matrix estimates (whose derivation will be discussed in Section 5.2, which deals with data and methodology),  $\gamma^{sigma}$  is the risk-aversion parameter,  $\gamma^{trade}$  is the trading penalty,  $\gamma^{hold}$  is the holding penalty,  $\widehat{TC}$  is the estimated transaction cost function (described shortly on the following page), and  $\widehat{HC}$  is the estimated holding cost function incorporating additional fees, such as borrowing fees for shorting assets (Boyd, et al., 2017). As this chapter aims to construct long-only portfolios the  $\widehat{HC}$ , related to short positions, will not be further pursued. The risk-aversion parameter controls the trade-off between risk and return; a low  $\gamma^{sigma}$  corresponds to a high appetite for risk, which can potentially generate high returns during the bullish regime, while high  $\gamma^{sigma}$  is typical of risk averse investors who can protect the portfolio during a bearish regime characterised by high volatility. The trading penalty controls for turnover. A high  $\gamma^{trade}$  leads to a lower turnover, which consumes less profits during rebalancing but exposes the portfolio to holding assets that may not be profitable in the longer term.

In contrast, a low  $\gamma^{trade}$  increases turnover, thus potentially the costs of trading, but allows the swift allocation of weights from unprofitable to profitable assets. The parameters  $\gamma^{sigma}$  and  $\gamma^{trade}$  need to be optimised which will be described in the methodology section later.

The estimation of the transaction cost function, i.e.,  $\widehat{TC}(\Delta w_{\tau,i})$ , will follow the solution recommended by (Boyd, et al., 2017), such that

$$\widehat{TC}(\Delta w_{t,i}) = \frac{b}{2} \times P_{t,i} \times |\Delta w_{t,i}| + \hat{\sigma}_{t,i} \frac{|\Delta w_{t,i}|^{\frac{3}{2}}}{\left(\frac{\hat{v}_{t,i}}{V_t}\right)^{\frac{1}{2}}} + c \times \Delta w_{t,i}, \quad (51)$$

where  $b$  is fundamentally the bid-ask spread, though it can also include additional costs (e.g., broker fees),  $P_{t,i}$  is the price of asset  $i$  at time  $t$ ,  $\hat{\sigma}_{t,i}$  is the estimated asset price volatility of asset  $i$  at time  $t$ ,  $\hat{v}_{t,i}$  is the estimated dollar volume traded for asset  $i$  at time  $t$ , and  $V_t$  is the total portfolio value at time  $t$ , and  $c$  is a parameter used to create asymmetry in the transaction cost function (when  $c > 0$ , the transaction costs of buying exceed those of selling, with the converse true when  $c < 0$ ). The estimation of the  $\hat{\sigma}_{t,i}$  and  $\hat{v}_{t,i}$  parameters, as well as the values of  $b$  and  $c$ , will be described in Section 5.2 on data and methodology.

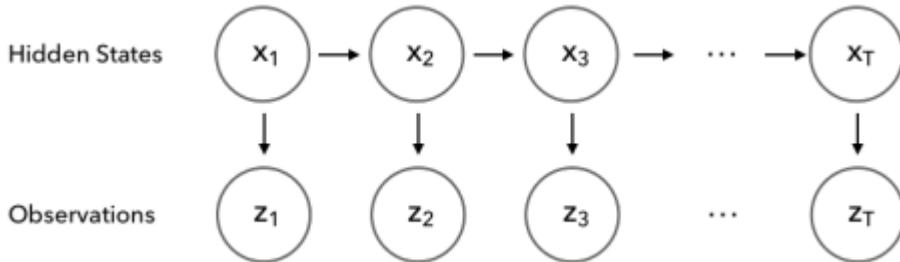
Finally, the utility function of Equation 50 assumes that the portfolio is self-funding, meaning it begins with a specific amount of money and no intermediate cash flows are registered; every profit is reinvested into the next period. In addition, the choice of the planning horizon  $H$ , which ultimately decides about the complexity of MPC with the mean-variance term (i.e., the number of allocation vectors  $w_{t+1}, w_{t+2}, \dots, w_{t+H}$  to be computed), is dependent on available return estimates  $\hat{r}_\tau$  and covariance matrix  $\hat{K}_\tau$ . As per Equation 50, if  $H > 1$ , then the utility function expects  $\hat{r}_{t+1}, \dots, \hat{r}_{t+H}$  and  $\hat{K}_{t+1}, \dots, \hat{K}_{t+H}$  at time  $t$ . The estimate of the covariance matrix can be simplified to a rolling covariance matrix, as suggested in (Boyd, et al., 2017), which this chapter will use and describe further in Section 5.2, which covers data and methodology; however, the return estimates require more a complex approach, and even though  $\hat{r}_{t+1}$  can be replicated from the predictions of the Chapter 4 models (as will be shown in Section 5.2), to tackle the problem of  $\hat{r}_{t+H}$  if  $H > 1$  this chapter will use the *Kalman filter* (KF), which is briefly introduced below.

### 5.1.3 Kalman filter

As previously mentioned in the literature review (Chapter 2), the Kalman filter, invented by Rudolf E. Kalman in the 1960s (Kalman, 1960), was initially used to track a moving target from noisy measurements of its position and to predict its future position. Today, the KF is used in various areas, such as location and navigation systems, control systems, computer graphics, target tracking, as well as finance. In financial time series, the KF is often used as a method to predict market betas (e.g., (Choudhry & Wu, 2008)), stock prices (e.g., (Martinelli & Rhoads, 2010)), or as a tool to denoise time series data (e.g., (Jansen, 2020), (Deepika & Bhat, 2021)).

Technically, the KF is a probabilistic model of a sequence of observations  $\{z_1, z_2, \dots, z_T\}$  and a corresponding sequence of hidden states  $\{x_1, x_2, \dots, x_T\}$ , as represented by Figure 21 below.

*Figure 21: The Kalman filter in a graphical representation. Source: (Jansen, 2020).*



The Kalman filter propagates the posterior distribution of the  $n$ -dimensional state variables  $x_t$ , given their measurements  $z_t$  over time. In the engineering framework from which the Kalman filter is derived,  $n$  is usually two (position and velocity); however, in this instance, the measurements will be one-dimensional, for instance, asset returns or prices, and it will be described later, in the methodology section, how the package was used in order to do this. To recover the hidden states  $x_t$  from a sequence of observations  $z_t$ , the KF iterates between the two following steps: prediction, which is an estimate of the current state, and measurement, which uses noisy observations  $z_t$  to update its estimate of the current state from the prediction step by averaging the information from both steps (prediction and measurement)

in a way that weights more certain estimates more highly. This can be translated to the basic logic of the KF: certain assumptions (explained shortly below) about a dynamic system, such as stock returns evolving over time, and a history of corresponding measurements, allow the estimation of a system's hidden state in a way that maximises the probability of the next measurements. The assumptions necessary to achieve the objective of estimating hidden states  $x_t$  are as follows:

- That the dynamic system behaves in a linear fashion.
- That the current hidden state vector  $x_t$  depends only on the most recent prior hidden state vector  $x_{t-1}$ .
- That measurements are subject to Gaussian, uncorrelated noise with constant covariance.

Mathematically, the KF framework can be expressed by the set of equations to follow, noting first the process below that defines the evolution of the hidden state from time  $t - 1$  to  $t$ ,

$$x_t = F_t x_{t-1} + B_t u_{t-1} + \varepsilon_{t-1}, \quad \varepsilon_{t-1} \sim N(0, Q),$$

where  $x_t$  is an  $n \times 1$  vector,  $F_t$  is an  $n \times n$  state transition matrix,  $B_t$  is an  $n \times k$  control-input matrix applied to a  $k \times 1$  control vector  $u_{t-1}$  (the term  $B_t u_{t-1}$  is not compulsory to include), and  $\varepsilon_{t-1}$  is an  $n \times 1$  process noise vector that is assumed to be zero-mean Gaussian with an  $n \times n$  covariance matrix  $Q$ . The process above is paired with the measurement model below that describes the relationship between the hidden state and the measurement at the current time step  $t$ ,

$$z_t = H_t x_t + \omega_t, \quad \omega_t \sim N(0, R),$$

where  $z_t$  is an  $m \times 1$  measurement vector,  $H_t$  is an  $m \times n$  measurement matrix (not to be confused with the horizon variable), and  $\omega_t$  is an  $m \times 1$  measurement noise vector that is assumed to be zero-mean Gaussian with an  $m \times m$  covariance matrix  $R$ . The objective of the Kalman filter is to estimate  $x_t$  at time  $t$ , given the initial estimate  $x_0$ , the  $n \times n$  initial hidden state error covariance matrix  $P_0$ , the sequence of measurements  $\{z_1, z_2, \dots, z_T\}$ , and the information of the dynamic system described by  $F_t, B_t, H_t, Q$  and  $R$ .

In order to achieve the objective described on the previous page, the Kalman filter is split into two procedures: prediction and update. The aim of the prediction stage is to estimate the current hidden state  $\mathbf{x}_t$  and the hidden state error covariance matrix  $P_t$ , while the update stage is focused on updating the current state estimate and the hidden state error covariance matrix by the use of the measurement vector  $\mathbf{z}_t$ . The prediction procedure involves the following set of equations:

$$\hat{\mathbf{x}}_t^- = F_t \hat{\mathbf{x}}_{t-1}^+ + B_t \mathbf{u}_{t-1}, \quad (52)$$

$$P_t^- = F_t P_{t-1}^+ F_t^T + Q, \quad (53)$$

where  $\hat{\mathbf{x}}_t$  is an estimate of the hidden state  $\mathbf{x}_t$  at time  $t$ ,  $P_t$  is the state error covariance matrix at time  $t$ , and the superscripts – and + denote predicted (prior) and updated (posterior) estimates, respectively. The update stage comprises of the following computations:

$$\tilde{\mathbf{z}}_t = \mathbf{z}_t - H_t \hat{\mathbf{x}}_t^-,$$

$$K_t = P_t^- H_t^T \times (R + H_t P_t^- H_t^T)^{-1}, \quad (54)$$

$$\hat{\mathbf{x}}_t^+ = \hat{\mathbf{x}}_t^- + K_t \tilde{\mathbf{z}}_t, \quad (55)$$

$$P_t^+ = (I - K_t H_t) \times P_t^-, \quad (56)$$

where  $\tilde{\mathbf{z}}_t$  is the measurement residual vector, that is, the difference between true measurements  $\mathbf{z}_t$  and estimated measurements  $H_t \hat{\mathbf{x}}_t^-$ ,  $H_t$  is an  $m \times n$  measurement matrix which maps the hidden state space to the measurement space,  $K_t$  is the Kalman gain<sup>59</sup>, an  $n \times m$  matrix that determines the relative weight of the prediction and the measurement in updating the hidden state estimate, and  $I$  is the identity matrix. The prediction and update procedures are repeated each time step  $t$  given the sequence of measurements  $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T\}$ .

The  $\hat{\mathbf{x}}_t^+$  vector of Equation 55 is effectively the estimation of the current hidden state vector  $\mathbf{x}_t$  given the correction  $K_t \tilde{\mathbf{z}}_t$  to the predicted prior estimate  $\hat{\mathbf{x}}_t^-$ . In other words, if the measurement vector  $\mathbf{z}_t$  contained stock returns, the  $\hat{\mathbf{x}}_t^+$  vector

<sup>59</sup> The Kalman gain can be interpreted as a weighting factor that balances the relative importance of the predicted state estimate and the measurement. It considers the uncertainty in both the predicted state estimate (represented by the covariance matrix  $P_t^-$ ) and the measurement noise (represented by the covariance matrix  $R$ ). The Kalman gain adjusts the contribution of each component in a way that minimizes the residual error of the estimated hidden state (Kalman, 1960).

would help estimate the underlying latent factors driving those returns. This is an important feature from the perspective of the predictability of the next returns which will be utilised in the following section. In addition, details of the method of application of the KF, as well as the values of  $F_t$ ,  $B_t$ ,  $H_t$ ,  $Q$ ,  $R$ ,  $\hat{x}_0$ , and  $P_0$ , will also be given in the following section.

## 5.2 Data and methodology

This section will cover data used for the experiment, the method of converting the classification predictions of the Chapter 4 models into estimated asset returns, as well as the implementation of the Kalman filter and the optimisation of the MPC parameters.

### 5.2.1 Data

Chapter 5 will re-use both equity and commodity assets and models from Chapter 4 in order to implement MPO; however, the data used for optimising and testing the model is limited to the holdout sample, as using data from the training or validation periods, with near-perfect predictions, would lead to overfitting. For the purposes of this chapter, the Chapter 4 holdout data will be split into two parts, one for optimising the model, and one for testing. It should be noted that the calculation of risk-related factors—as will be made clear in the methodology section—necessarily re-uses also the training sample listed in Table 10 of Chapter 4, in order to have enough data. However, this is not a problem as only returns are used for this purpose, not model predictions.

Table 24, on the following page, re-lists the assets used, with names and labels, as well as adds the US 3-month treasuries to serve as a cash proxy for the portfolio. The specific asset data used in each case comprises of daily returns, volatility of the returns, and trading volumes, where the latter two are necessary

inputs for the transaction cost function (Equation 51) described in the previous section.

Table 24: Historical asset data used for MPO.

Asset class	Asset	Asset label
<b>Equities</b>	S&P/ASX 200 ETF	<i>ASX</i>
	CAC40 ETF	<i>CAC</i>
	DAX ETF	<i>DAX</i>
	FTSE 100 ETF	<i>FTSE</i>
	FTSE MIB ETF	<i>FTSEMIB</i>
	KOSPI ETF	<i>KOSPI</i>
	MSCI China ETF	<i>MSCI_China</i>
	NASDAQ 100 ETF	<i>NASDAQ</i>
	NIFTY 50 ETF	<i>NIFTY</i>
	Nikkei 225 ETF	<i>Nikkei</i>
	Swiss Market ETF	<i>SMI</i>
	S&P 500 ETF	<i>SPX</i>
<b>Commodities</b>	S&P/TSX Composite ETF	<i>TSX</i>
	TWSE ETF	<i>TWSE</i>
	Aluminium Futures	<i>Aluminium</i>
	Brent Crude Oil Futures	<i>Brent</i>
	Coffee Futures	<i>Coffee</i>
	Copper Futures	<i>Copper</i>
	Corn Futures	<i>Corn</i>
	Gold Futures	<i>Gold</i>
	Live Cattle Futures	<i>Live_cattle</i>
	Natural Gas Futures	<i>Natural_gas</i>
	Nickel Futures	<i>Nickel</i>
<b>Cash</b>	Soybeans Futures	<i>Soybeans</i>
	Sugar Futures	<i>Sugar</i>
<b>Cash</b>	Wheat Futures	<i>Wheat</i>
	US 3-month Treasury Bill	<i>US3M</i>

MPO will incorporate both asset classes simultaneously; hence the data selected from the Chapter 4 holdout samples must for the purposes of this chapter begin at the same day. The period from 02/04/2018 to 29/04/2022 is used, being compatible with all assets. The experiment described here does not use the Chapter 4 benchmarks, namely MSCI ACWI for equities and Bloomberg Commodity Index for commodities; instead, the MPO performance will be benchmarked against 1/N and

buy-and-hold portfolios, which is a common practice in the industry, as well as being done in the works of (Boyd, et al., 2017) and (Li, Uysal, & Mulvey, 2022).

The following methodology section will show how the classification predictions of the models of Chapter 4 will be transformed by the use of a Kalman filter (Equations 52-56), the estimation of the covariance matrix for MPC, and finally the selection of the MPC parameters to be optimised (Equation 50), as well as estimating the transaction cost function given in Equation 51.

## 5.2.2 Methodology

### *Transforming classification predictions into return estimates*

As was shown in Section 5.1 (background and related work), MPO requires return estimates in order to dynamically allocate weights. The models of Chapter 4 predict the probability of entering or exiting a specific regime rather than raw asset returns. Thus, in order to transform probabilities into return estimates, this chapter will combine approaches from (Boyd, et al., 2017) and (Li, Uysal, & Mulvey, 2022); specifically, it will:

1. For each asset  $i$  calculate a lagged 10-day exponential weighted moving average<sup>60</sup> ( $EMA_{r_{t-1,i}}$ ) of actual daily returns (Boyd, et al., 2017).
2. Multiply the most recent value from step 1 by the probability of entering a new regime (Li, Uysal, & Mulvey, 2022), where:
  - a. If the probability indicates the “bullish” regime, i.e.,  $p_{t,i}^{bull}$ , the value from step 1 is multiplied by  $-p_{t,i}^{bull}$ .
  - b. If the probability indicates the “bearish” regime, i.e.,  $p_{t,i}^{bear}$ , the value from step 1 is multiplied by  $p_{t,i}^{bear}$ .
  - c. If the probability indicates the “other” regime, the value from step 1 is left “as is”, since the direction of estimated returns is unknown.

---

<sup>60</sup> Recall from Section 4.2.1 that the exponential weighted moving average (EMA) is calculated as  $EMA = \left[ C \times \left( \frac{2}{1+n} \right) \right] + EMA_p \times \left[ 1 - \left( \frac{2}{1+n} \right) \right]$ , where  $C$  is a closing price or last return of an asset,  $p$  is a prior close or return, and  $n$  is a moving time window, such as 10, for 10 days.

Recall that the models from Chapter 4 are contrarian, thus the "bullish" regime points at a possibility of selling / shorting an asset (hence  $-p_{t,i}^{bull}$ ), whilst the "bearish" regime indicates a buying opportunity (hence  $p_{t,i}^{bear}$ ). Thus, to summarise, the transformed asset return estimate can be expressed as

$$\hat{r}_{t+1,i} = \begin{cases} -p_{t,i}^{bull} \times EMA_{r_{t-1,i}} & \text{if prediction is "bullish"} \\ p_{t,i}^{bear} \times EMA_{r_{t-1,i}} & \text{if prediction is "bearish"} \\ EMA_{r_{t-1,i}} & \text{if prediction is "other".} \end{cases} \quad (57)$$

Since MPC allocates portfolio weights over a specific time horizon  $H$ , it requires return estimates at time  $t$  for future times  $t + 1, \dots, t + H$ ; however, the regime predictions of models from Chapter 4 are made only for time  $t + 1$ . In order to tackle this problem, a novel two-step solution has been implemented, in which an initial estimate for each required return within the horizon  $H$ , using available past time information, is then accuracy-boosted by the use of a Kalman filter:

1. Each estimated daily return within the horizon  $H$  is initially calculated as a lagged return from Equation 54. Specifically, the estimated return  $\hat{r}_\tau$ , for  $\tau = t + 1, \dots, t + H$  will be calculated using  $EMA_{r_{\tau-H,i}}$ .
2. The estimates  $\hat{r}_{\tau,i}$  are then improved by the use of a Kalman filter. Specifically, the accuracy-boosted return estimates  $\hat{r}_{\tau,i}^{KF}$  are calculated as the original estimate  $\hat{r}_{\tau,i}$  plus the corresponding lagged estimated hidden state  $\hat{x}_{\tau-1,i}^+$  from Equation 54 in Section 5.1.3, in which, within the generic Kalman filter package used, the expected "velocity" components of the required vectors are set to a constant value of 1, and the variables required here are assigned the role of "position". Update equations are hence not in essence vector-matrix equations, and the vector notation will be from this point suppressed, leading to the update rule

$$\hat{r}_{\tau,i}^{KF} = \hat{r}_{\tau,i} + \hat{x}_{\tau-1,i}^+. \quad (58)$$

Thus, for example, if  $H = 5$ , then the most recent value of  $\hat{r}_{\tau,i}^{KF}$  will estimate asset returns for  $t + 5$ ,  $\hat{r}_{\tau-1,i}^{KF}$  will estimate asset returns for  $t + 4$ , and so on. In this way, asset return estimates for all horizons  $H$  are made available at time  $t$ . The reason for using the lagged estimated hidden state  $\hat{x}_{\tau-1,i}^+$  stems from the issue of data availability at time  $\tau$ ; recall that  $\hat{x}_t^+$  from Equation 54 requires  $\hat{x}_t^-$ , which can be

calculated from  $\hat{x}_{t-1}^-$  (Equation 52), and  $K_t \tilde{z}_t$  that is a correction term. The problem with  $K_t \tilde{z}_t$  is that it requires values at the time  $t$  that will not be available at the time of the trade, which occurs at  $t$  based on estimates for  $\tau$ . Thus, the lagged values  $\hat{x}_{\tau-1,i}^+$  must be used in order to remain certain that the same methodology will work on a live dataset after the holdout set ends.

Note that the procedure described in this sub-section does not apply to cash, as this asset was not predicted in Chapter 4. Instead, US 3-month treasury bill yields are used as a cash proxy; these are annualised and then divided by 252 (number of trading days within a year) to obtain an average daily yield. These yields are subsequently lagged in a similar manner to  $\hat{r}_{\tau,i}^{KF}$ , so that the most current bond yield estimates cash returns for  $t + 5$ , while the first lag will estimate  $t + 4$ , and so on.

The next sub-section on the Kalman filter will briefly describe how the value of the estimated hidden state mean  $\hat{x}_{\tau-1,i}^+$  has been obtained.

#### *Applying the Kalman filter to boost the accuracy of estimated returns*

Recall from Section 5.1.3 that the Kalman filter (KF) requires several parameters to solve Equations 52-56, namely  $F_t$ ,  $B_t$ ,  $H_t$ ,  $Q$ ,  $R$ ,  $\hat{x}_0$ , and  $P_0$ . As the purpose of the KF is to boost the accuracy of estimated returns, due to the ability of  $\hat{x}_t^+$  to estimate hidden drivers of real returns, the value of  $\hat{x}_{0,i}$  is the first available lagged difference between estimated return  $\hat{r}_{\tau-H-1,i}$  and the real return  $r_{\tau-H-1,i}$ , such that

$$\hat{x}_{0,i} = \hat{r}_{\tau-H-1,i} - r_{\tau-H-1,i}. \quad (59)$$

Similarly to *EMA* expressions in Equation 57, the difference between estimated and real returns must be lagged, as the real returns will not be available at the time of the trade  $t$ . Going forward in time, the  $\hat{x}_{\tau-1,i}^+$  term can be treated as an estimation error between estimated and real returns that the Kalman filter subsequently aims to correct and potentially predict for the next timestep; this is the reason why  $\hat{x}_{\tau-1,i}^+$  serves as the accuracy booster for the estimated returns  $\hat{r}_{\tau,i}$  inside Equation 58.

The values of  $F_t$ ,  $B_t$ ,  $H_t$ ,  $R$ , and  $P_0$  are assumed default as used in the package that calculates the Kalman filter, FilterPy<sup>61</sup>. Specifically (noting again that the package expected variables with two dimensions, though here the "velocity" was set to 1),

$$F_t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, P_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, H_t = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{ and } R = 2,$$

and the external control term  $B_t u_{t-1}$  has been excluded from Equation 52 (also by default). The only non-default value is assumed by the parameter  $Q$  of Equation 53, representing process uncertainty (noise), such that,  $Q = \bar{\sigma}_T^2$ , where  $\bar{\sigma}_T^2$  is a mean variance of asset returns between a given asset's start date (see Table 10 in Chapter 4 for the start date for each asset used) until the beginning of the data period used for this chapter (06/03/2018).

The next sub-section will briefly describe the estimation of the covariance matrix, which is another critical component of the MPC method.

### *Estimation of the covariance matrix*

Apart from the estimated returns, MPC with the mean-variance term formula (Equation 50) requires an estimation of the covariance matrix  $\hat{K}_\tau$  in order to calculate the  $\gamma^{sigma}\hat{K}_\tau(w_\tau + s_\tau)$  term. This problem could in itself be a focus of extensive research, as there is no universal approach to the computation of  $\hat{K}_\tau$  (Boyd, et al., 2017). This chapter will thus use the solution offered by (Boyd, et al., 2017), namely that  $\hat{K}_\tau$  is a lagged rolling covariance matrix of asset returns over a period of 504 days (two trading years), where  $\tau \rightarrow \tau - H - 1$ . It will be assumed that the estimated risk of underlying assets can be explained by their current risk; the return estimates incorporate certain information about the future risk from the regime prediction perspective, i.e., if “bullish” then higher volatility should be expected (since it is a sell signal), while if “bearish” then the volatility should be subdued (as it is a buy signal).

---

<sup>61</sup> FilterPy is one of the most used Python packages for Kalman filter implementation. See <https://filterpy.readthedocs.io/en/latest> for description. Accessed on 21/02/2023.

The next subsection briefly describes the estimation of the transaction cost function  $\widehat{TC}(\Delta w_{\tau,i})$  (Equation 51), which the  $\gamma^{trade}$  component of MPC (Equation 50) heavily relies on.

#### *Estimation of the transaction cost function*

Recall from Equation 51 that the modelled transaction cost function  $\widehat{TC}(\Delta w_{\tau,i})$  requires the parameters  $b$  (a bid-ask spread and optional additional costs),  $\hat{\sigma}_{t,i}$  (the estimated price volatility for asset  $i$  at time  $t$ ),  $\hat{v}_{t,i}$  (the estimated dollar volume traded for asset  $i$  at time  $t$ ), and  $c$  (to create asymmetry in the transaction cost function). In the original work of (Boyd, et al., 2017), the parameter  $b$  was assumed to be only five basis points; however, following the high costs assumptions from Chapter 4,  $b$  will here be set to 20 basis points, as this would also include potential explicit costs, such as trading fees. The parameter  $c$  will assume its default value, namely 0. The parameter  $c$  is set to zero because this chapter does not encompass short-selling, thus there is no need to differentiate between transaction costs associated with long and short positions. The  $\hat{\sigma}_{t,i}$  and  $\hat{v}_{t,i}$  terms will be computed as in (Boyd, et al., 2017) using a lagged 10-day exponential weighted moving average. Thus, the estimated modelled transaction cost function  $\widehat{TC}(\Delta w_{\tau,i})$  is expressed as

$$\widehat{TC}(\Delta w_{t,i}) = \frac{b}{2} \times P_{t,i} \times |\Delta w_{t,i}| + EMA_{\sigma_{t-1,i}} \frac{|\Delta w_{t,i}|^{\frac{3}{2}}}{\left( \frac{EMA_{v_{t-1,i}}}{V_t} \right)^{\frac{1}{2}}}, \quad (60)$$

where  $P_{t,i}$  is an asset  $i$  price at time  $t$ ,  $EMA_{\sigma_{t-1}}$  is a lagged 10-day exponentially weighted moving standard deviation of asset returns,  $EMA_{V_{t-1}}$  is a lagged 10-day exponentially weighted moving average of the dollar volume traded for the asset, and  $V_t$  is the total portfolio value at time  $t$ .

The following sub-section will describe the optimisation of the parameters of MPC, as well as its implementation.

### *MPC parameter optimisation and the algorithm implementation*

Before implementing MPC of Equation 50, the values of  $H$  (investment horizon),  $\gamma^{\text{sigma}}$ , and  $\gamma^{\text{trade}}$  need to be established. Recall that  $\gamma^{\text{hold}}$  in Equation 50 is associated with short positions; thus in this chapter it will be assumed to be 0, as the portfolio allocation is restricted to long-only positions.

**Investment horizon.** The choice of the investment horizon  $H$  relies heavily on the accuracy of the risk-return estimates, since less accurate predictions may result in incorrect portfolio weights over a longer horizon, which may lead to unprofitable trades (see Table 24 and Equation 50 for the relationship between weights, investment horizon, and risk-return estimates). In addition, using a higher value for  $H$  is associated with higher computational complexity, longer calculation time, and finally increased probability of failure for MPC to converge over the investment horizon (Li, Uysal, & Mulvey, 2022). In the original work of (Boyd, et al., 2017)  $H$  was set to 2, whilst in (Li, Uysal, & Mulvey, 2022) values between 2 and 30 were studied, though in the same work it was shown that beyond  $H = 5$ , the portfolio performance based on the MPC results was gradually deteriorating. The initial experiments of this chapter assumed  $H$  to be 2, 3, 4, and 5; however, each subsequent value of  $H$  resulted in slower calculations, as well as an issue of running out of computational memory when  $H \geq 4$ . Thus, following (Boyd, et al., 2017) and (Li, Uysal, & Mulvey, 2022), this chapter will assume  $H$  to be 2, since based on the initial experiments of this chapter,  $H = 2$  outperformed the portfolio constructed upon the results of MPC when  $H = 3$ .

**Optimisation of  $\gamma^{\text{sigma}}$  and  $\gamma^{\text{trade}}$ .** The parameters  $\gamma^{\text{sigma}}$  and  $\gamma^{\text{trade}}$  are associated with the portfolio risk and turnover, respectively. They may be selected based on an investor's need in order to construct less risky and dynamic, though potentially less profitable, portfolios (or the converse), as was done in (Li, Uysal, & Mulvey, 2022); however, this chapter will attempt to find an optimal combination of  $\gamma^{\text{sigma}}$  and  $\gamma^{\text{trade}}$  in order to arrive at portfolio weights that result in a portfolio capable of significantly outperforming its benchmarks, namely buy-and-hold and 1/N portfolios. To achieve these optimal parameters, this chapter will use the

Optuna package used in Chapter 4; MPC will be run 200 times, using the training portion of the data, with different  $\gamma^{sigma}$  and  $\gamma^{trade}$  parameters, in order to arrive at the portfolio with the highest Sortino ratio (see Equation 45 from Chapter 4), the same scoring metric used in Chapter 4. Table 25, below, shows the parameter search space; note that the beginning and ending values of each parameter are based on suggestions from (Li, Uysal, & Mulvey, 2022).

*Table 25: Optimised parameters of MPC.*

Parameter	Type of data	Search space
$\gamma^{sigma}$	Float	{0.01,...,1000}
$\gamma^{trade}$	Float	{0.0001,...,25}

Specifically, the optimisation of the parameters from Table 26 will be performed over first two years of available data from Table 25, i.e., 02/04/2018 – 12/03/2020. The remaining data (periods between 27/03/2020 – 29/04/2022) will be used as a holdout sample to check the portfolio performance against its benchmarks.

**Implementation of MPC.** To execute the procedure outlined in Table 24, and thus compute portfolio weights for  $H = 2$  using risk-return estimates and different  $\gamma^{sigma}$  and  $\gamma^{trade}$ , this chapter will use the CVXPortfolio<sup>62</sup> package created by the authors of (Boyd, et al., 2017). Apart from the return and risk estimates ( $\hat{r}_\tau^{KF}$  and  $\hat{R}_\tau$ ), the estimated cost function ( $\widehat{TC}(\Delta w_{t,i})$ ),  $\gamma^{sigma}$  and  $\gamma^{trade}$ , CVXPortfolio requires the dates over which the portfolio construction takes place and the initial portfolio value. As mentioned in the previous paragraph on the optimisation of the  $\gamma^{sigma}$  and  $\gamma^{trade}$  parameters, the dates 02/04/2018 – 12/03/2020 will be used for optimisation, while 27/03/2020 – 29/04/2022 will be used for holdout testing. Between both set of dates there is a 15-day gap to avoid any leakage of the training data to the testing data; the Kalman filter necessary to calculate the return estimates from Equation 58 is additionally computed anew for the testing set in order to avoid

---

<sup>62</sup> See details on CVXPortfolio here: <https://github.com/cvxgrp/cvxportfolio>. Accessed on 21/02/2023.

using any information from the training set, even with the 15-day gap present. The starting portfolio value is \$26,000, equally distributed among assets (\$1,000 per each asset), except from the cash which initially is assumed to be 0. Finally, MPC has been restricted to assume the minimum weight of each asset to be 1% (thus making Equation 50 subject to  $w_{\tau,i} \geq 0.01$  instead of  $w_{\tau,i} \geq 0$ ); this is due to the issue of Chapter 3, where the optimisation algorithm moved the majority of the funds to cash in order to achieve a high Sharpe ratio, as the almost zero risk associated with cash could significantly enlarge the Sharpe ratio (or, as would be the case for this experiment, the Sortino ratio).

The CVXPortfolio package will optimise the portfolio weights for  $H = 2$  given the information described in the paragraph above and will execute the trades on a roll-forward basis. The cost-adjusted returns from these trades are stored in a vector array ready to be used for financial metric calculations when the procedure is completed. The next section will show and discuss the results of implementing MPC based on the methodology which has been described in this section.

## 5.3 Results and discussion

### 5.3.1 Optimal MPC parameters

Table 26 below shows the optimal MPC parameters after running the algorithm 200 times on the training dataset.

*Table 26: Optimal MPC parameters.*

Parameter	Optimal value
$\gamma^{sigma}$	0.1262
$\gamma^{trade}$	4.667

Recall from Section 5.1.2 on MPC that a high  $\gamma^{sigma}$  is associated with high risk aversion, while the higher the  $\gamma^{trade}$ , the lower the turnover of the portfolio. The optimal parameters in this chapter put more emphasis on lower turnover, though allow the portfolio to be riskier; that is, the algorithm can increase weights on assets with high volatility in order to potentially gain higher returns. The lower turnover can be linked to higher bid-ask spreads used to estimate the transaction cost function (Equation 60), while the lower risk aversion may stem from the accurate return estimates, particularly in commodities, which even though characterised by higher variance were the best performing asset class in Chapter 4, and whose predictions are a vital part of the return estimates of this chapter. In summary, more accurate return estimates allow one to assume more risk in a portfolio, since the potential profits may outweigh the higher risk. The next sub-section will investigate this risk-return performance on the holdout set.

### 5.3.2 MPC portfolio performance analysis

Table 27: *KAMA+RF+MPC+Kalman* portfolio performance vs. benchmarks, on the entire holdout set. Note the excess returns are calculated as portfolio cost-adjusted returns minus the risk-free returns of the cash component, namely US 3-month Treasury Bills.

Statistic	KAMA+RF+MPC+Kalman portfolio	Buy-and-hold portfolio	1/N portfolio
Mean excess returns	8.2%	0.23%	0.13%
Annualised mean excess returns	205.41%	65.43%	31.52%
Volatility	5.98%	2.64%	0.68%
Ann. volatility	94.93%	41.89%	10.74%
Ann. Sharpe ratio	2.16	1.56	2.93
Ann. Sortino ratio	5.11	2.67	3.9
Maximum drawdown	43.43%	36.17%	5.19%
Calmar ratio <sup>63</sup>	4.73	1.81	6.08
Ann. information ratio (vs. buy-and-hold portfolio)	2.11	-	-
Ann. information ratio (vs. 1/N portfolio)	1.91	-	-

Table 27 above shows the MPC portfolio performance over the entire holdout set and compares it to the selected benchmarks, namely buy-and-hold and 1/N portfolios. Note that the MPC portfolio will also be referred to as the *KAMA+RF+MPC+Kalman* portfolio to stress the various methods necessary to arrive at the result.

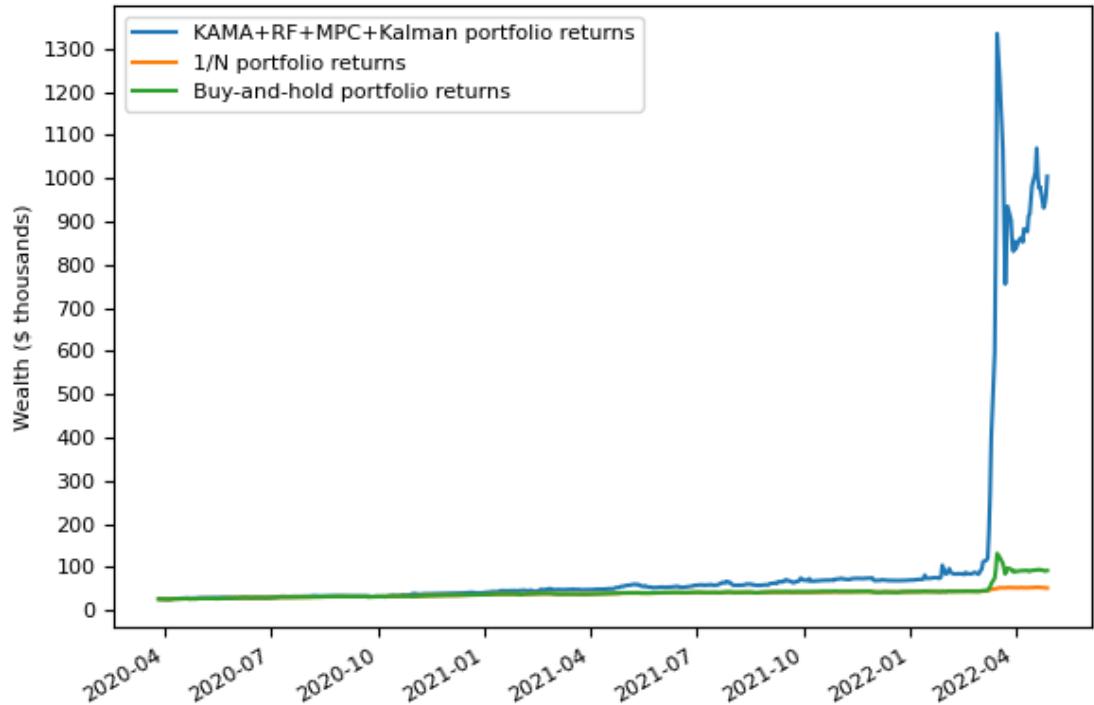
It can be immediately noticed that the MPC portfolio has significantly outperformed its benchmarks due to extraordinary returns at certain times, which outweighed the

---

<sup>63</sup> The Calmar ratio is a measure of risk-adjusted returns calculated as annualised return divided by maximum portfolio drawdown. In the industry, the Calmar ratio above 0.5 is already considered to be good, although the most desirable value is above 3.

increased volatility. Figures 22, 23, 24, and 25 below are helpful to explain the reason behind this.

Figure 22: KAMA+RF+MPC+Kalman portfolio performance vs. its benchmarks, over the entire holdout set. Note that the ticks on the left-hand side represent gained wealth in thousands of USD.



It can be seen in Figure 22 that up until March 2022 the MPC portfolio was steadily outperforming its benchmarks; however, its performance later suddenly soared, which can also be seen in the buy-and-hold portfolio (in the chart known as the 'holding portfolio'). This is due to the beginning of the Russo-Ukrainian War which caused the prices of wheat and nickel (shown in Figures 23 and 24) to soar.

Figure 23: Price of nickel since 2018. It can be noticed that at the beginning of the Russo-Ukrainian War (February 2022), the price nearly tripled over a short period of time.

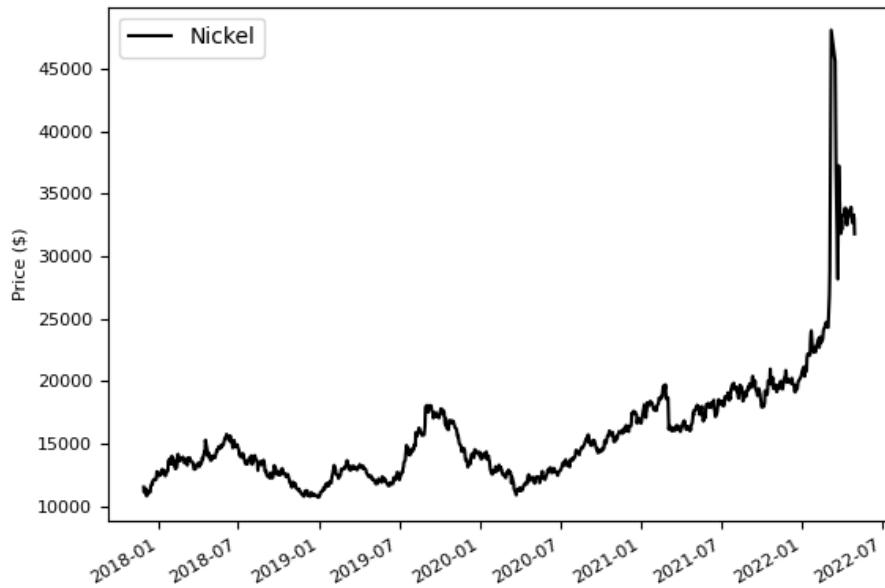


Figure 24: The price of wheat since 2018. It can be noticed that at the beginning of the Russo-Ukrainian War (February 2022), the price nearly tripled over a short period of time.

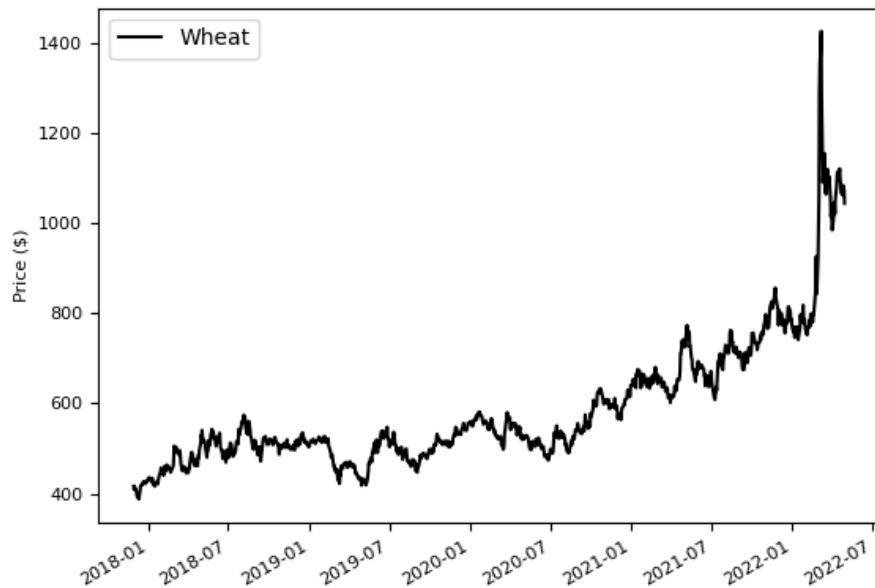
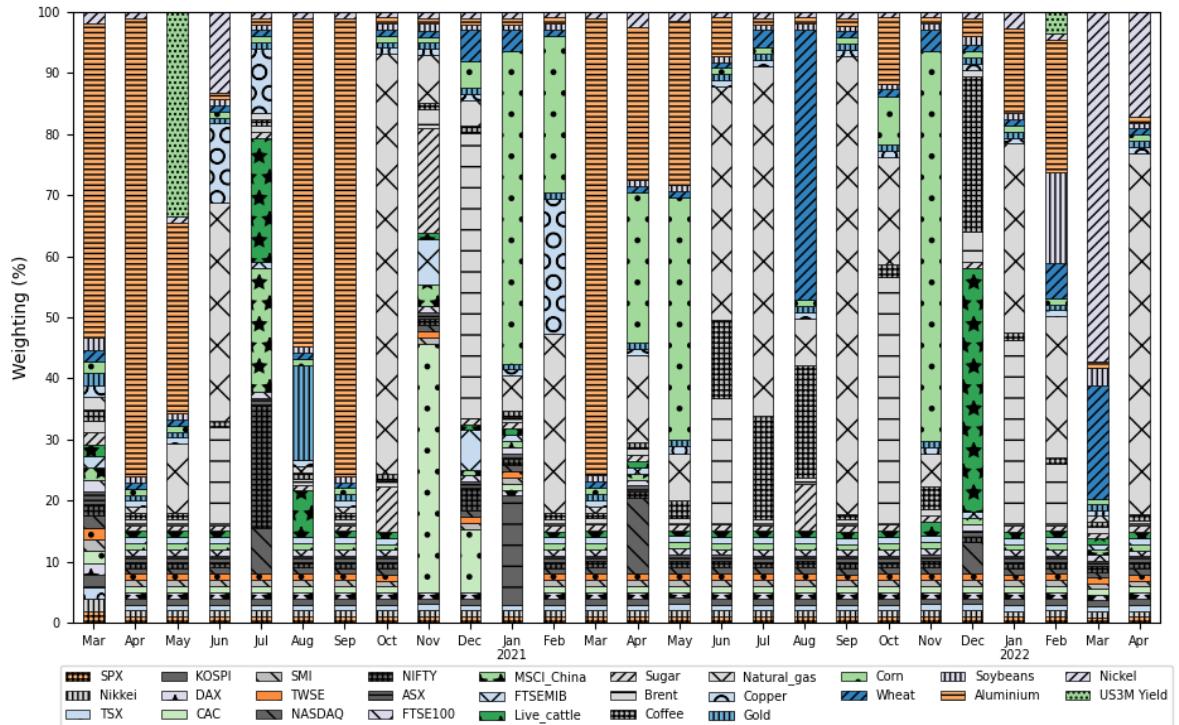


Figure 25 below reveals that in March 2022 the dominating asset was nickel, while the second most important was wheat, which explains the sudden increase in the portfolio performance visible in Figure 22.

*Figure 25: KAMA+RF+MPC+Kalman portfolio average monthly asset weights over the entire holdout set.*



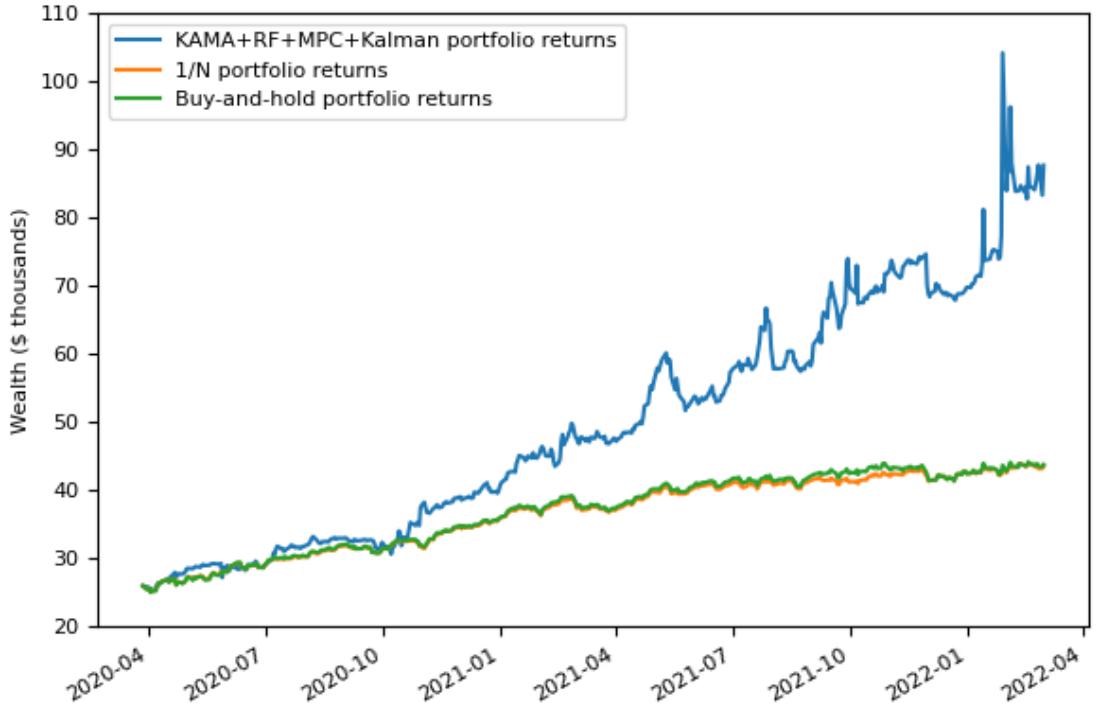
Though the KAMA+RF+MPC+Kalman portfolio performance shows promise, its extraordinary performance has been attributed to a singular event, which could even be treated as a “fluke” from the performance perspective. This is why Table 28 on the following page shows the MPC portfolio performance against its benchmarks up until 1<sup>st</sup> of March 2022 in order to remove the “war effect” from the results.

Table 28: KAMA+RF+MPC+Kalman portfolio performance vs. benchmarks, over the holdout set up until 1<sup>st</sup> of March 2022. Note the excess returns are calculated as portfolio cost-adjusted returns minus the risk-free returns of the cash component, namely US 3-month Treasury Bills.

Statistic	KAMA+RF+MPC+Kalman portfolio	Buy-and-hold portfolio	1/N portfolio
Mean excess returns	0.29%	0.11%	0.1%
Annualised mean excess returns	72.91%	26.95%	26.35%
Daily volatility	2.78%	0.71%	0.65%
Ann. volatility	44.09%	11.34%	10.25%
Ann. Sharpe ratio	1.65	2.38	2.57
Ann. Sortino ratio	2.32	3.03	3.24
Maximum drawdown	20.55%	5.97%	5.18%
Calmar ratio	3.55	4.52	5.09
Ann. information ratio (vs. buy-and-hold portfolio)	1.16	-	-
Ann. information ratio (vs. 1/N portfolio)	1.13	-	-

It can be seen from the table that the KAMA+RF+MPC+Kalman portfolio achieved the largest annualised excess returns also up until March 2022, i.e., before the "war effect", though it also had the highest volatility (due to the low  $\gamma^{\text{sigma}}$  parameter). Even though the benchmark portfolios achieved higher annualised Sharpe and Sortino ratios, the KAMA+RF+MPC+Kalman portfolio still managed to obtain annualised information ratios (IRs) above 1 versus both benchmarks; an IR  $> 1$  is highly desirable in the industry, as a high IR indicates a skilful manager who outperforms the benchmarks given the risk taken. As additional evidence, Figure 26 on the following page shows the MPC portfolio cumulative performance over the holdout set up until March 2022.

Figure 26: KAMA+RF+MPC+Kalman portfolio performance vs. its benchmarks, over the holdout set up until 1<sup>st</sup> of March 2022. Note that the ticks on the left-hand side represent gained wealth in thousands of USD.



Initially, the MPC portfolio underperformed its benchmarks, but after approximately five months it began outperforming the holding and 1/N portfolios. It is again evident that the KAMA+RF+MPC+Kalman portfolio is characterised by a higher volatility; however, obtained cost-adjusted returns outweigh the increased risk. In Figure 25, shown previously, which breaks down asset weights per each month, it can be seen that the portfolio is dominated by commodities, mainly natural gas, corn, wheat, Brent crude, nickel, and aluminium, which also explains the increased volatility of the MPC portfolio, due to the increased variance of commodity features relative to equities; however, the commodity domination is not surprising, as the commodity model in Chapter 4, whose predictions are used in this chapter, was shown to be the best performing.

Finally, Figure 27 on the following page shows the MPC portfolio performance without nickel and wheat, over the entire holdout set. The optimal  $\gamma^{sigma}$  and  $\gamma^{trade}$  parameters from Table 27 have been re-used to run this analysis,

although another optimisation over the training set, without these two commodities, could potentially improve the results.

*Figure 27: KAMA+RF+MPC+Kalman portfolio performance vs. its benchmarks, excluding nickel and wheat, over the entire holdout set. Note that the ticks on the left-hand side represent gained wealth in thousands of USD.*

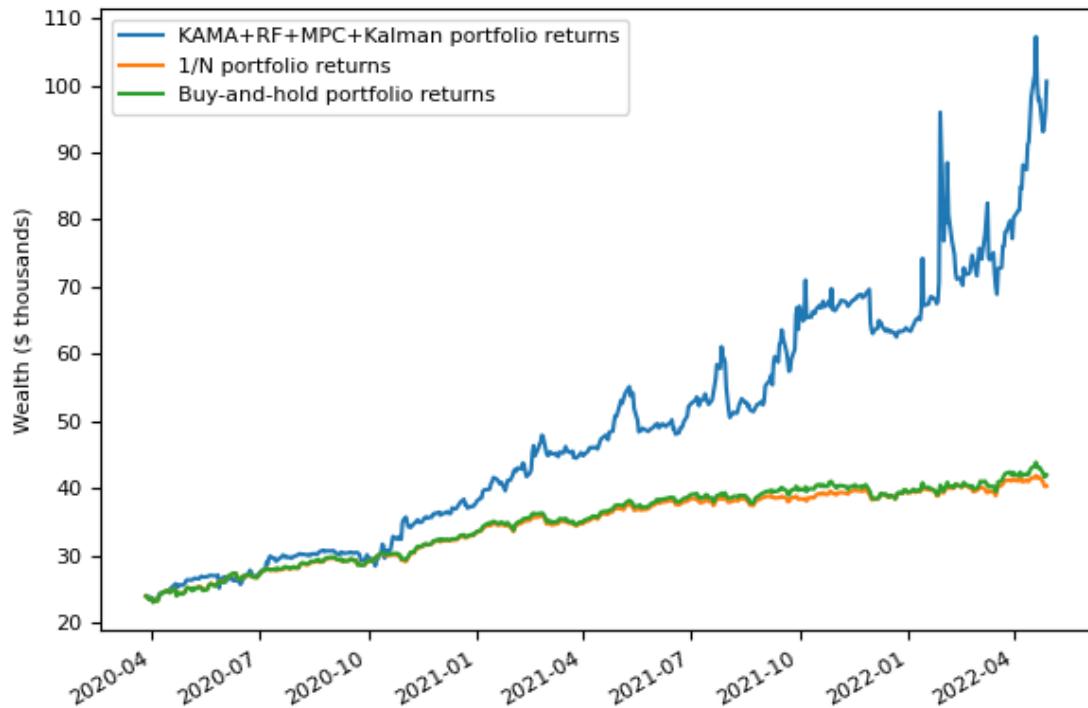


Table 29, on the following page, shows the performance of the KAMA+RF+MPC+Kalman portfolio without the nickel and wheat commodity components vs. its benchmarks, over the entire holdout set.

Without nickel and wheat, it can be seen from this table that the MPC portfolio performance is still excellent, as it achieved  $IR > 1$  over its benchmarks, as well as a high annualised Sharpe ratio of 1.73; even so, the volatility of the KAMA+RF+MPC+Kalman portfolio was high relative to the benchmarks, which caused the buy-and-hold and 1/N portfolios to achieve higher Sharpe ratios.

Table 29: KAMA+RF+MPC+Kalman portfolio performance vs. benchmarks (excluding wheat and nickel) on the entire holdout set. Note the excess returns are calculated as portfolio cost-adjusted returns minus the risk-free returns of the cash component, namely US 3-month Treasury Bills.

Statistic	KAMA+RF+MPC+Kalman portfolio (excl. nickel and wheat)	Buy-and-hold portfolio (excl. nickel and wheat)	1/N portfolio (excl. nickel and wheat)
Mean excess returns	0.3%	0.1%	0.1%
Annualised mean excess returns	74.34%	26.21%	24.21%
Daily volatility	2.71%	0.74%	0.66%
Ann. volatility	43.04%	11.79%	10.43%
Ann. Sharpe ratio	1.73	2.22	2.32
Ann. Sortino ratio	2.17	2.91	3
Maximum drawdown	28.21%	6.03%	4.27%
Calmar ratio	2.63	4.16	5.66
Ann. information ratio (vs. buy-and-hold portfolio)	1.22	-	-
Ann. information ratio (vs. 1/N portfolio)	1.23	-	-

### 5.3.3 Is the higher risk worth it?

Clearly, a limitation of the MPC portfolio is its elevated volatility relative to the benchmarks, even though MPC-obtained returns outweigh the risk, based on its reported Sharpe ratios. Technically, the benchmarks achieved higher Sharpe and Sortino ratios up until 1<sup>st</sup> of March 2022, as can be seen in Table 28, or over the entire holdout sample in the study without nickel and wheat (Table 29), which is preferable for an investor who seeks the highest-Sharpe/Sortino portfolios. Is the higher risk worthwhile to achieve superior returns? This depends on the investor's needs; if the same investor decided to put wealth into the benchmark portfolios,

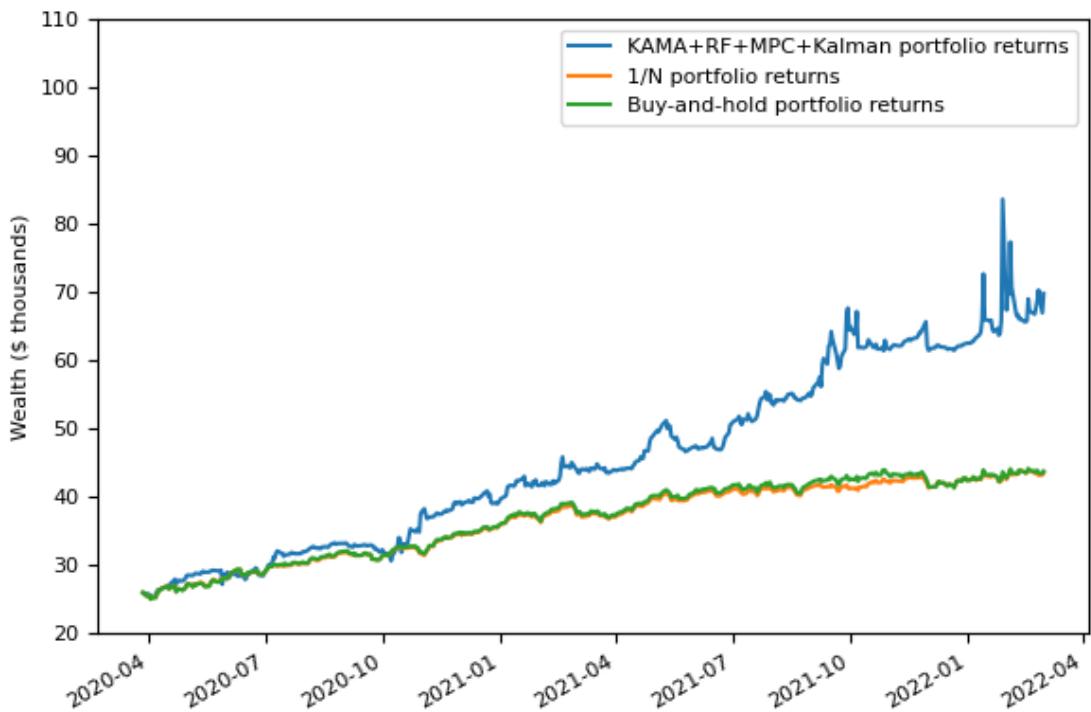
ultimately, they would be far less profitable than if they had put the funds into the KAMA+RF+MPC+Kalman portfolio. However, this investor might prefer to earn less for the sake of slow and steady growth, instead of more turbulent ups and downs. One way to reduce the volatility of the MPC portfolio would be to omit more volatile components, such as certain commodities or emerging market equities, noting that this can also significantly diminish returns. Another solution, which does not require any components to be omitted from the portfolio, might be to tweak the  $\gamma^{\text{sigma}}$  parameter in MPC to better suit a given investor's needs. Assuming the  $\gamma^{\text{trade}}$  parameter remains the same as in Table 26, Table 30 below shows the MPC portfolio performance with a more rigid  $\gamma^{\text{sigma}}$  parameter, which was here set to 50 (increased from the optimal value of approximately 0.13). Note that this analysis was performed on the holdout set (until 1<sup>st</sup> March 2022) without any training, just for the purpose of demonstrating the difference in the new portfolio performance against the original portfolio, whose statistics can be seen in Table 28.

*Table 30: KAMA+RF+MPC+Kalman portfolio performance up until 1<sup>st</sup> March 2022 with the  $\gamma^{\text{sigma}}$  parameter set to 50. Note these statistics are for a portfolio now again containing nickel and wheat.*

<b>Statistic</b>	<b>KAMA+RF+MPC+Kalman portfolio</b>
Mean excess returns	0.22%
Annualised mean excess returns	57.29%
Daily volatility	2.23%
Ann. volatility	35.57%
Ann. Sharpe ratio	1.61
Ann. Sortino ratio	1.85
Maximum drawdown	21.42%
Calmar ratio	2.67
Ann. information ratio (vs. buy-and-hold portfolio)	0.91
Ann. information ratio (vs. 1/N portfolio)	0.94

The annualised volatility dropped by nine percentage points relative to the MPC portfolio with the optimal  $\gamma^{sigma}$  parameter (Table 28), although the annualised returns shrank by almost 20 percentage points. The significantly higher  $\gamma^{sigma}$  indeed reduced risk but also had a substantially negative impact on the MPC portfolio returns. Figure 28 shows the performance of this modified, risk-reduced, portfolio, against the selected benchmarks.

Figure 28: KAMA+RF+MPC+Kalman portfolio performance vs. benchmarks up until 1<sup>st</sup> March 2022 with the  $\gamma^{sigma}$  parameter set to 50. Note this performance is for a portfolio now again containing nickel and wheat.

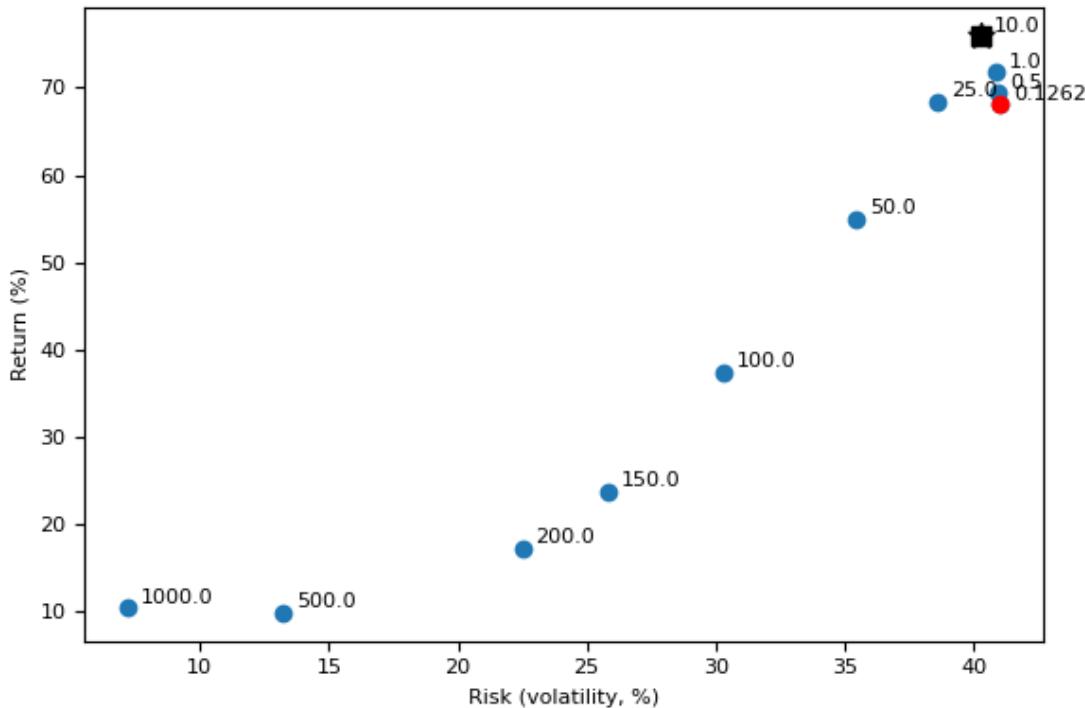


Even though the MPC portfolio outperformed its benchmarks on the cumulative returns basis, Table 30, on the previous page, showed that the achieved information ratios for the risk-modified version are below 1, thus making the modified KAMA+RF+MPC+Kalman portfolio less attractive for investors than the MPC portfolio with the optimal  $\gamma^{sigma}$  parameter (Table 28) which allowed for higher risk.

Finally, Figure 29, on the following page, shows the relationship between risk and returns of the MPC portfolio (up until 1<sup>st</sup> March 2022) with different  $\gamma^{sigma}$  parameters. It can be immediately seen that a larger  $\gamma^{sigma}$  often leads to a significant reduction in both risk and returns of the MPC portfolio. A risk-averse

investor can thus select a higher value for the  $\gamma^{\text{sigma}}$  and accept the lower rate of return in order to also obtain the lower risk; however, a visualisation such as Figure 29 could also help clarify for them that their profits might be significantly reduced along with the lower risk.

Figure 29: Annualised risk-return relationship for a range of values of the risk-aversion parameter  $\gamma^{\text{sigma}}$ , calculated for the test data (up until 1<sup>st</sup> March 2022), with the  $\gamma^{\text{sigma}}$  values shown beside each dot. The star indicates the  $\gamma^{\text{sigma}}$  with the highest Sortino ratio, and the square that with the highest Sharpe ratio (coincidentally, they are both at the 10.0 dot). The red dot indicates the position occupied by the KMRF+MPC portfolio with  $\gamma^{\text{sigma}} = 0.1262$ , optimised to deliver the best Sortino ratio during the preceding MPC validation period.



## 5.4 Conclusions

This chapter focused on applying model predictive control (MPC) to the predictions from Chapter 4 in order to achieve realistic regime-robust asset allocation, including multiple assets of different classes, over multiple horizons. MPC can incorporate factors such as time-varying return estimates, transaction costs constraints, risk constraints, and the trade-off between short-term versus long-term asset holding. By transforming the classification predictions from Chapter 4 into return estimates

by the use of a Kalman filter, and applying Optuna to optimise the MPC parameters, it was possible to obtain an optimal long-only portfolio capable of outperforming its benchmarks ( $1/N$  and buy-and-hold portfolios) with respect to the information ratio, as well as cumulative return, even after accounting for a more rigid cost model. MPO with model predictive control shows promise in building realistic portfolios by exploiting signals generated by, in this case, a machine learning model.

However, even though the MPC portfolio achieved superior returns, it could be immediately noticed that this was linked to higher risks. For risk-averse investors this could be problematic, as their investment strategies may simply not allow for higher levels of risk, even if this could lead to substantial profits. This, however, could be addressed either by changing the components of the portfolio, such that higher variance assets are dropped (e.g., certain commodities or emerging markets equities), or by tweaking the MPC's  $\gamma^{sigma}$  parameter to balance the risk-return relationship such that it better fits a risk-averse investor's needs.

## **6. Conclusions and future work**

In this final chapter, the work of this thesis is summarised and discussed, prior to outlining proposed extensions of the work in terms of detecting and predicting financial regimes in order to construct portfolios capable of withstanding turbulent times in the markets, as well as benefitting from price rallies.

### **6.1 Summary and discussion of the experiments undertaken**

The ultimate objective of this thesis was to propose and test a detection-prediction-optimisation regime-switching framework that would also be efficient in a real-world environment, and thus practical for asset managers. This problem was split into three research objectives, each becoming a focus of a single chapter, each chapter building upon preceding work:

1. Implementation of a novel regime detection framework.
2. Prediction of financial regimes ex-ante based on the detected regimes from 1.
3. Construction of realistic, regime-robust portfolios using the generated signals from 2.

In order to ensure the proposed methods were both novel and likely to succeed, Chapter 2 thoroughly reviewed the relevant literature and built foundations upon which the subsequent chapters were constructed.

The first objective listed above was the focus of Chapter 3, which blended a technical indicator, Kaufman's Adaptive Moving Average (KAMA), with a popular statistical method of regime detection, Markov-switching regression (MSR), in order to smoothly and accurately detect financial regimes. To test the effectiveness of this method, called the KAMA+MSR model, several other regime detection frameworks were used as benchmarks, notably Markov-switching regression. Even though the studied models were limited by the fact that regimes were being detected, not

predicted, and hence there would be an inevitable lag in the adjustment of trading positions, the KAMA+MSR model proved its superiority over its benchmarks, particularly Markov-switching regression used alone, in terms of stability and accuracy, which was reflected in its achievement of the best financial results, and thus could be used as a tool to generate concise labels to be subsequently fed into a predictive algorithm, the topic of the following chapter.

Prediction of the financial regimes ex-ante, the second objective of this thesis, was the focus of Chapter 4. Using the generated regime labels from Chapter 3, as well as appropriate feature engineering, feature selection, and the use of a Random Forest model as predictor, it was shown that financial regimes derived from those of Chapter 3 can be predicted efficiently, regardless of the asset class. The models showed solid out-of-sample accuracy, as well as achieving excellent financial results based on a long/short, cost-inclusive trading strategy exploiting the predicted signals. However, the main limitations of this chapter were that, in the real-world environment, many institutions are unable to short assets for ethical reasons; in addition, the implemented trading strategy, even though more advanced than an asset-cash exchange (as in Chapter 3), still did not fully reflect the real-world portfolio construction.

Both the contributions and limitations of Chapter 4 were addressed in Chapter 5, which focused on the third research objective. With the use of the model predictive control (MPC) algorithm, it was shown that the generated signals from Chapter 4 could be successfully exploited in portfolio construction, incorporating various real-world factors, such as risk, transaction, and liquidity constraints. The constructed portfolio was capable of outperforming its benchmarks, even though it did not allow shorting positions, which potentially could enhance its performance, as implied by the results of Chapter 4, yet were set aside in this chapter due to the aforementioned constraints on some financial institutions in regard to shorting.

The combined contributions of Chapters 3, 4, and 5 achieved the ultimate objective of this research, able to serve as key components of a quantitative trading system outlined in Section 2.1.3, stressing the practical aspect of this thesis.

## 6.2 Future work

A range of extensions to this work could be suggested, as described below.

1. **Incorporation of additional asset classes and financial instruments.** While the research presented in this thesis demonstrated the effectiveness of the proposed regime-switching framework across a range of asset classes, further investigation could be done to determine its suitability for other asset classes, such as bonds or cryptocurrencies, as well as financial instruments, such as options. This could lead to the development of a more comprehensive quantitative trading system that incorporated a wider range of assets and could mimic short trading—an ethical concern for multiple institutions—by buying bonds or put options. The latter, in particular, could effectively increase the cumulative return performance of the portfolios described in Chapter 5.
2. **Investigation of alternative prediction models.** While the Random Forest model used in Chapter 4 was effective in predicting financial regimes ex-ante, further investigation could be done to determine whether alternative machine learning models could provide improved performance or efficiency. For example, popular boosting algorithms, such as XGBoost or LightGBM, or neural network models (including recurrent neural networks), might offer advantages in capturing the complex relationships between financial data and regime changes that Random Forest might not have been able to capture.
3. **Incorporation of more advanced MPO optimisation techniques.** While multi-period optimisation (MPO) implemented via model predictive control (MPC) was used in Chapter 5 to construct realistic, regime-robust, long-only portfolios, there may be other optimisation techniques that could further improve performance; for example, genetic algorithms or particle swarm optimisation could be used instead of MPC to search for optimal portfolio weights.

In summary, such future work could lead to the development of more comprehensive and sophisticated quantitative trading systems capable of

generating more reliable predictions and constructing more effective, asset-diverse, portfolios able to take advantage of regime shifts while taking into account real-world constraining factors such as ethical considerations.

## References

- Adams, R. P., & MacKay, D. J. (2007). Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.
- Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering - A decade review. *Information Systems*, 16-38.
- Aistis, R., Vaidotas, L., & Edmundas, M. (2013). Moving Averages for Financial Data Smoothing. In S. Tomas, B. Rimantas, & B. Rita, *Information and Software Technologies. ICIST 2013. Communications in Computer and Information Science, vol 403* (pp. 34-45). Berlin: Springer.
- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2623–2631). ACM.
- Aldrich, J. H., & Nelson, F. D. (1984). *Linear probability, logit, and probit models*. Sage.
- Alessi, L., & Savona, R. (2021). Machine Learning for Financial Stability. In S. Consoli, D. R. Recupero, & M. Saisana, *Data Science for Economics and Finance* (pp. 65-87). Springer.
- Alizadeh, A. H., Nomikos, N. K., & Pouliasis, P. K. (2008). A Markov regime switching approach for hedging energy commodities. *Journal of Banking and Finance*, 1970–1983.
- Alizadeh, A., & Nomikos, N. (2004). A Markov regime switching approach for hedging stock indices. *Journal of Futures Markets*, 649-674.
- Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 175-185.
- Andrews, D. W., & Kim, J.-Y. (2003). End-of-sample cointegration breakdown tests. Available at SSRN 386081.

- Ang, A., & Bekaert, G. (2002). International asset allocation with regime shifts. *The Review of Financial Studies*, 1137-1187.
- Ang, A., & Bekaert, G. (2002). Regime switching in interest rates. *Journal of Business & Economic Statistics*, 163-182.
- Ang, A., & Bekaert, G. (2004). How Regimes Affect Asset Allocation. *Financial Analysts Journal*, 86-99.
- Ang, A., & Chen, J. (2002). Asymmetric correlations of equity portfolios. *Journal of Financial Economics*, 443-494.
- Ang, A., & Timmermann, A. (2012). Regime Changes and Financial Markets. *Annual Review of Financial Economics*, 313-337.
- Arthur, D., & Vassilvitskii, S. (2006). *k-means++: The Advantages of Careful Seeding*. Stanford.
- Assoe, K. G. (1998). Regime-switching in emerging stock market returns. *Multinational Finance Journal*, 101-132.
- Athey, S. (2019). The Impact of Machine Learning on Economics. In A. Agrawal, J. Gans, & A. Goldfarb, *The Economics of Artificial Intelligence: An Agenda* (pp. 507-547). Chicago: University of Chicago Press.
- Babazadeh, H., & Esfahanipour, A. (2019). A novel multi period mean-VaR portfolio optimization model considering practical constraints and transaction cost. *Journal of Computational and Applied Mathematics*, 313-342.
- Babu, M. S., Geethanjali, N., & Satyanarayana, B. (2012). Clustering approach to stock market prediction. *International Journal of Advanced Networking and Applications*, 1281-1291.
- Bae, G. I., Kim, W. C., & Mulvey, J. M. (2014). Dynamic asset allocation for varied financial markets under regime switching framework. *European Journal of Operational Research*, 450-458.
- Bagliano, F. C., & Favero, C. A. (1998). Measuring monetary policy with VAR models: An evaluation. *European Economic Review*, 1069-1112.

Bai, J., & Perron, P. (2003). Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 1-22.

Baillie, R. T., Bollerslev, T., & Mikkelsen, H. O. (1996). Fractionally integrated generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 3-30.

Ballings, Michel, V. d., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 7046-7056.

Bandt, C., & Pompe, B. (2002). Permutation entropy: a natural complexity measure for time series. *Physical review letters*, 174102.

Bansal, N., Connolly, R. A., & Stivers, C. (2010). Regime-switching in stock index and Treasury futures returns and measures of stock market stress. *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, 753-779.

Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, 1554-1563.

Bauwens, L., Lubrano, M., & Richard, J.-F. (2000). *Bayesian inference in dynamic econometric models*. New York: Oxford University Press.

Bauwens, L., Preminger, A., & Rombouts, J. (2010). Theory and inference for a Markov switching GARCH model. *Econometrics Journal*, 218-244.

Bekiros, S. D., & Paccagnini, A. (2016). Policy-Oriented Macroeconomic Forecasting with Hybrid DGSE and Time-Varying Parameter VAR Models. *Journal of Forecasting*, 613-632.

Berge, T. J. (2015). Predicting recessions with leading indicators: model averaging and selection over the business cycle. *Journal of Forecasting*, 455-471.

Bernard, H., & Gerlach, S. (1998). Does the term structure predict recessions? The international evidence. *International Journal of Finance & Economics*, 195-215.

Bernard, J.-T., Khalaf, L., Kichian, M., & McMahon, S. (2008). Forecasting commodity prices: GARCH, jumps, and mean reversion. *Journal of Forecasting*, 279-291.

- Bialkowski, J. (2004). Modelling returns on stock indices for western and central european stock exchanges - a Markov switching approach. *South Eastern Europe Journal of Economics*, 81-100.
- Billio, M., & Monfort, A. (1998). Switching state-space models Likelihood function, filtering and smoothing. *Journal of Statistical Planning and Inference*, 65-103.
- Billio, M., Casarin, R., & Sartore, D. (2007). Bayesian inference in dynamic models with latent factors. *WORKING PAPER-DEPARTMENT OF ECONOMICS, CÀ FOSCARI. UNIVERSITY OF VENICE*, 1-30.
- Billio, M., Casarin, R., Ravazzolo, F., & van Dijk, H. K. (2012). Combination schemes for turning point predictions. *The Quarterly Review of Economics and Finance*, 402-412.
- Bilokon, P., Jacquier, A., & McIndoe, C. (2021). *Market regime classification with signatures*. arXiv preprint arXiv:2107.00066.
- Bini, B. S., & Mathew, T. (2016). Clustering and regression techniques for stock prediction. *Procedia Technology*, 1248-1255.
- Birchenhall, C. R., Jessen, H., Osborn, D. R., & Simpson, P. (1999). Predicting US business-cycle regimes. *Journal of Business & Economic Statistics*, 313-323.
- Birchenhall, C., Osborn, D., & Sensier, M. (2001). Predicting UK business cycle regimes. *Scottish Journal of Political Economy*, 179-195.
- Black, F., & Litterman, R. (1992). Global Portfolio Optimization. *Financial Analysts Journal*, 28-43.
- Boldin, M. D. (1996). A check on the robustness of Hamilton's Markov switching model approach to the economic analysis of the business cycle. *Studies in Nonlinear Dynamics and Econometrics*, 35-46.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 307-327.
- Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*. (pp. 144-152). COLT.

Boyd, S., Busseti, E., Diamond, S., Kahn, R. N., Koh, K., Nystrup, P., & Speth, J. (2017). Multi-period trading via convex optimization. *Foundations and Trends® in Optimization*, 1-76.

Bracegirdle, C., & Barber, D. (2011). Switch-reset models: Exact and approximate inference. *Proceedings of The Fourteenth International Conference on Artificial Intelligence and Statistics*, 190-198.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 123-140.

Breiman, L. (2001). Random forests. *Machine learning*, 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Taylor & Francis.

Brock, W., Lakonishok, J., & LeBaron, B. (1992). Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. *The Journal of Finance*, 1731-1764.

Brodsky, E., & Darkhovsky, B. S. (2013). Nonparametric methods in change point problems. Vol. 243. *Springer Science & Business Media*.

Bulla, J. (2011). Hidden Markov models with t components. Increased persistence and other aspects. *Quantitative Finance*, 459-475.

Bulla, J., Mergner, S., Bulla, I., Sesboüé, A., & Chesneau, C. (2011). Markov-switching asset allocation: Do profitable strategies exist? *Journal of Asset Management*, 310-321.

Candanedo, L. M., Feldheim, V., & Deramaix, D. (2017). Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, 81-97.

Caporin, M., & McAleer, M. (2013). Ten things you should know about the dynamic conditional correlation representation. *Econometrics*, 115-126.

Carmona, P., Climent, F., & Momparler, A. (2019). Predicting failure in the US banking sector: An extreme gradient boosting approach. *International Review of Economics & Finance*, 304-323.

Carpinteyro, M., Venegas-Martínez, F., & Aali-Bujari, A. (2021). Modeling Precious Metal Returns through Fractional Jump-Diffusion Processes Combined with Markov Regime-Switching Stochastic Volatility. *Mathematics*, 407-434.

Cboe Exchange, Inc. (2021, 08 20). *Cboe*. Retrieved from Cboe Volatility Index: <https://cdn.cboe.com/resources/vix/vixwhite.pdf>

Chang, P.-C., Wang, D.-d., & Zhou, C.-l. (2012). A novel model by evolving partially connected neural network for stock price trend forecasting. *Expert Systems with Applications*, 611-620.

Chan-Lau, J. A. (2017). *Lasso regressions and forecasting models in applied stress testing*. International Monetary Fund.

Chappell, D. R. (2019). Regime heteroskedasticity in Bitcoin: A comparison of Markov switching models. *Journal of Applied Economic Sciences (JAES)*, 621-641.

Chauvet, M., & Potter, S. (2005). Forecasting recessions using the yield curve. *Journal of forecasting*, 77-103.

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

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. (pp. 785-794). ACM.

Chen, Y.-C., Rogoff, K. S., & Rossi, B. (2010). Can exchange rates forecast commodity prices? *The Quarterly Journal of Economics*, 1145-1194.

Cheung, W. (2010). The Black–Litterman model explained. *Journal of Asset Management*, 229-243.

Cheung, Y.-W., & Lai, K. S. (1995). Lag order and critical values of the augmented Dickey–Fuller test. *Journal of Business & Economic Statistics*, 277-280.

Choi, K., & Hammoudeh, S. (2010). Volatility behavior of oil, industrial commodity and stock markets in a regime-switching environment. *Energy Policy*, 4388-4399.

- Choudhry, T., & Wu, H. (2008). Forecasting ability of GARCH vs Kalman filter method: evidence from daily UK time-varying beta. *Journal of Forecasting*, 670-689.
- Chow, G. C. (1960). Tests of equality between sets of coefficients in two linear regressions. *Econometrica*, 591–605.
- Christ, M., Kempa-Liehr, A. W., & Feindt, M. (2016). *Distributed and parallel time series feature extraction for industrial big data applications*. arXiv preprint arXiv:1610.07717.
- Christoffel, K., Coenen, G., & Warne, A. (2010). Forecasting with DSGE models. *ECB Working Paper, No. 1185, European Central Bank*.
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*. arXiv preprint arXiv:1412.3555 .
- Consolo, A., Favero, C. A., & Paccagnini, A. (2009). On the statistical identification of DSGE models. *Journal of Econometrics*, 99-115.
- Dao, T.-L. (2014). Momentum Strategies with L1 Filter. *Journal of Investment Strategies*, 1-26.
- Das, S., & Roy, S. S. (2021). Predicting regime switching in BRICS currency volatility: a Markov switching autoregressive approach. *DECISION*, 1-16.
- Davig, T., & Hall, A. S. (2019). Recession forecasting using Bayesian classification. *International Journal of Forecasting*, 848-867.
- Davis, R. A., Lee, T. C., & Rodriguez-Yam, G. A. (2006). Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, 223-239.
- Dbouk, W., & Jamali, I. (2018). Predicting daily oil prices: Linear and non-linear models. *Research in International Business and Finance*, 149-165.
- De, S. K., & Mukherjee, S. S. (2022). Exact tests for offline changepoint detection in multichannel binary and count data with application to networks. *Journal of Statistical Computation and Simulation*, 3659-3678.

- Deepika, N., & Bhat, M. N. (2021). An efficient stock market prediction method based on Kalman Filter. *Journal of The Institution of Engineers (India): Series B*, 629-644.
- Del Negro, M., & Schorfheide, F. (2013). DSGE model-based forecasting. *Handbook of economic forecasting*, 57-140.
- DeMiguel, V., Garlappi, L., & Raman, U. (2009). Optimal Versus Naive Diversification: How Inefficient is the 1/N Portfolio Strategy? *The Review of Financial Studies*, 1915-1953.
- Deng, H., & Runger, G. (2013). Gene selection with guided regularized random forest. *Pattern Recognition*, 3483-3489.
- Dey, S., Kumar, Y., Saha, S., & Basak, S. (2016). *Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting*. PESIT South Campus.
- Di Persio, L., & Honchar, O. (2017). Recurrent neural networks approach to the financial forecast of Google assets. *International journal of Mathematics and Computers in simulation*, 7-13.
- Dias, A., & Embrechts, P. (2002). Change-point analysis for dependence structures in finance and insurance. *Novos Rumos em Estatística (Ed. C. Carvalho, F. Brilhante and F. Rosado)*, Sociedade Portuguesa de Estatística, Lisbon, 69-86.
- Dias, J. G., Vermunt, J. K., & Ramos, S. (2015). Clustering financial time series: New insights from an extended hidden Markov model. *European Journal of Operational Research*, 852-864.
- Djemo, C. R., Eita, J. H., & Mwamba, J. W. (2021). Predicting Foreign Exchange Rate Movements: An Application of the Ensemble Method. *Review of Development Finance*, 58-70.
- Döpke, J., Fritzsche, U., & Pierdzioch, C. (2017). Predicting recessions with boosted regression trees. *International Journal of Forecasting*, 745-759.
- Du, P., Kibbe, W. A., & Lin, S. M. (2006). Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 2059-2065.

Dueker, M., & Neely, C. J. (2007). Can Markov switching models predict excess foreign exchange returns? *Journal of Banking & Finance*, 279-296.

Eddy, S. R. (1996). Hidden markov models. *Current opinion in structural biology*, 361-365.

Edelen, R., Evans, R., & Kadlec, G. (2013). Shedding Light on "Invisible" Costs: Trading Costs and Mutual Fund Performance. *Financial Analysts Journal*, 34-44.

El Mokhtari, K., Higdon, B. P., & Başar, A. (2019). Interpreting financial time series with SHAP values. *CASCON '19: Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*. (pp. 166-172). ACM.

Elliott, R. J., & Osakwe, C. (2018). New Filters for the Calibration of Regime Switching Beta Dynamics. *Communications on Stochastic Analysis*, 7.

Ellis, C. A., & Parbery, S. A. (2005). Is smarter better? A comparison of adaptive, and simple moving average trading strategies. *Research in International Business and Finance*, 399-411.

Engel, C., & Hamilton, J. D. (1990). Long swings in the dollar: Are they in the data and do markets know it? *American Economic Review*, 689-713.

Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 987-1007.

Engle, R. F. (2002). Dynamic Conditional Correlation: A Simple Class of Multivariate Generalized Autoregressive Conditional Heteroskedasticity Models. *Journal of Business & Economic Statistics*, 339-350.

Estrella, A., & Hardouvelis, G. A. (1991). The term structure as a predictor of real economic activity. *The Journal of Finance*, 555-576.

Estrella, A., & Mishkin, F. S. (1998). Predicting US recessions: Financial variables as leading indicators. *Review of Economics and Statistics*, 45-61.

Estrella, A., Rodrigues, A. P., & Schich, S. (2003). How stable is the predictive power of the yield curve? Evidence from Germany and the United States. *Review of Economics and Statistics*, 629-644.

- Fama, E. F., & French, K. R. (1989). Business conditions and expected returns on stocks and bonds. *Journal of financial economics*, 23-49.
- Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 3133-3181.
- Filardo, A. J. (1999). How reliable are recession prediction models? *Economic Review - Federal Reserve Bank of Kansas City*, 35-56.
- Fiszeder, P., & Fałdziński, M. (2019). Improving forecasts with the co-range dynamic conditional correlation model. *Journal of Economic Dynamics and Control*, 103736.
- Fossati, S. (2016). Dating US business cycles with macro factors. *Studies in Nonlinear Dynamics & Econometrics*, 529-547.
- Frankel, J. A. (2008). The effect of monetary policy on real commodity prices. *Asset prices and monetary policy*, 291.
- Franses, P. H., & Van Dijk, D. (1996). Forecasting stock market volatility using (non-linear) Garch models. *Journal of Forecasting*, 229-235.
- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 119-139.
- Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 337-407.
- Friewald, N., Wagner, C., & Zechner, J. (2014). The cross-section of credit risk premia and equity returns. *The Journal of Finance*, 2419-2469.
- Frühwirth-Schnatter, S. (2001). Fully Bayesian analysis of switching Gaussian state space models. *Annals of the Institute of Statistical Mathematics*, 31-49.
- Garcia, R., & Perron, P. (1996). An analysis of the real interest rate under regime shifts. *Review of Economics and Statistics*, 111-125.

Gauvain, J.-L., & Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE transactions on speech and audio processing* (pp. 291-298). IEEE.

Geurkink, Y., Boone, J., Verstockt, S., & Bourgois, J. G. (2021). Machine learning-based identification of the strongest predictive variables of winning and losing in Belgian professional soccer. *Applied Sciences*, 2378.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 3-42.

Gilchrist, S., & Zakrajšek, E. (2012). Credit spreads and business cycle fluctuations. *American economic review*, 1692-1720.

Giordani, P., & Kohn, R. (2008). Efficient Bayesian inference for multiple change-point and mixture innovation models. *Journal of Business & Economic Statistics*, 66-77.

Giudici, P., & Hashish, I. A. (2020). A hidden Markov model to detect regime changes in cryptoasset markets. *Quality and Reliability Engineering International*, 2057-2065.

Giusto, A., & Piger, J. (2017). Identifying business cycle turning points in real time with vector quantization. *International Journal of Forecasting*, 174-184.

Gogas, P., Papadimitriou, T., Matthaiou, M., & Chrysanthidou, E. (2015). Yield Curve and Recession Forecasting in a Machine Learning Framework. *Computational Economics*, 635-645.

Gokcan, S. (2000). Forecasting volatility of emerging stock markets: linear versus non-linear GARCH models. *Journal of Forecasting*, 499-504.

Goldfeld, S. M., & Quandt, R. E. (1973). A Markov model for switching regressions. *Journal of Econometrics*, 3-15.

Graflund, A., & Nilsson, B. (2003). Dynamic portfolio selection: The relevance of switching regimes and investment horizon. *European Financial Management*, 179-200.

- Grondman, I., Busoniu, L., Lopes, G. A., & Babuska, R. (2012). A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* (pp. 1291-1307). IEEE.
- Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 2223-2273.
- Guidolin, M. (2011). Markov switching models in empirical finance. In D. M. Drukker, *Missing data methods: Time-series methods and applications*. (pp. 1-86). Emerald Group Publishing Limited.
- Guidolin, M., & Timmermann, A. (2005). Economic implications of bull and bear regimes in UK stock and bond returns. *Economic Journal*, 111-143.
- Guidolin, M., & Timmermann, A. (2007). Asset allocation under multivariate regime switching. *Journal of Economic Dynamics and Control*, 3503-3544.
- Gunasekharage, A., & Power, D. M. (2001). The profitability of moving average trading rules in South Asian stock markets. *Emerging Markets Review*, 17-33.
- Gupta, A., & Dhingra, B. (2012). Stock market prediction using hidden markov models. *2012 Students Conference on Engineering and Systems*. (pp. 1-4). IEEE.
- Habibi, R. (2022). Bayesian online change point detection in finance. *Financial Internet Quarterly*, 27-33.
- Hallac, D., Vare, S., Boyd, S., & Leskovec, J. (2017). Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 215-223).
- Hamilton, J. D. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica*, 357-384.
- Han, L., Wan, L., & Xu, Y. (2020). Can the Baltic Dry Index predict foreign exchange rates? *Finance Research Letters*, 101157.
- Hansen, P. R., & Lunde, A. (2005). A forecast comparison of volatility models: does anything beat a GARCH (1, 1)? *Journal of Applied Econometrics*, 873-889.

Hassan, R., & Nath, B. (2005). Stock market forecasting using hidden Markov model: a new approach. *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)* (pp. 192-196). IEEE.

Hau, H., & Rey, H. (2006). Exchange rates, equity prices, and capital flows. *The Review of Financial Studies*, 273-317.

Hauptmann, J., Hoppenkamps, A., Min, A., Ramsauer, F., & Zagst, R. (2014). Forecasting market turbulence using regime-switching models. *Financial Markets and Portfolio Management*, 139-164.

Herzog, F., Dondi, G., & Geering, H. P. (2007). Stochastic model predictive control and portfolio optimization. *International Journal of Theoretical and Applied Finance*, 203-233.

Hess, M. K. (2006). Timing and diversification: A state-dependent asset allocation approach. *The European Journal of Finance*, 189-204.

Hinkley, D. V. (1970). Inference about the change-point in a sequence of random variables. *Biometrika*, 1-17.

Hoang, D., & Wiegert, K. (2023). Machine learning methods in finance: Recent applications and prospects. *European Financial Management*, 1657-1701.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 1735-1780.

Hosking, J. R. (1981). Fractional differencing. *Biometrika*, 165-175.

Hossain, M. A., Karim, R., Thulasiram, R., Bruce, N. D., & Wang, Y. (2018). Hybrid deep learning model for stock price prediction. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1837-1844). IEEE.

Hsu, P.-H., & Kuan, C.-M. (2005). Reexamining the profitability of technical analysis with data snooping checks. *Journal of Financial Econometrics*, 606-628.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 489-501.

- Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers and Operations Research*, 2513-2522.
- Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 73-101.
- Iglesias, F., & Kastner, W. (2013). Analysis of similarity measures in times series clustering for the discovery of building energy patterns. *Energies*, 579-597.
- Imandoust, S. B., & Bolandraftar, M. (2013). Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications*, 605-610.
- Israelsen, C. L. (2005). A refinement to the Sharpe ratio and information ratio. *Journal of Asset Management*, 423-427.
- Jabeur, S. B., Mefteh-Wali, S., & Viviani, J.-L. (2021). Forecasting gold price with the XGBoost algorithm and SHAP interaction values. *Annals of Operations Research*, 1-21.
- Jansen, S. (2020). *Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python, 2nd Edition*. Packt Publishing.
- Jeanne, O., & Masson, P. (2000). Currency crises, sunspots and Markov-switching regimes. *Journal of International Economics*, 327-350.
- Jedelyn, C., Adolf, J., Tuerlinckx, F., Kuppens, P., & Ceulemans, E. (2018). Detecting long-lived autodependency changes in a multivariate system via change point detection and regime switching models. *Scientific Reports*, 15637.
- Jurman, G., Riccadonna, S., & Furlanello, C. (2012). A comparison of MCC and CEN error measures in multi-class prediction. *PLoS ONE*, e41882.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 35-45.
- Kasahara, H., & Shimotsu, K. (2018). Testing the number of regimes in Markov regime switching models. *arXiv preprint arXiv:1801.06862*.

Katechos, G. (2011). On the relationship between exchange rates and equity returns: A new approach. *Journal of International Financial Markets, Institutions and Money*, 550-559.

Kato, K. (2010). Solving L1 Regularization Problems With Piecewise Linear Losses. *Journal of Computational and Graphical Statistics*, 1024-1040.

Kaufman, P. J. (1995). *Smarter Trading: Improving Performance in Changing Markets*. New York: McGraw-Hill.

Kauppi, H., & Saikkonen, P. (2008). Predicting US recessions with dynamic binary response models. *The Review of Economics and Statistics*, 777-791.

Kawahara, Y., Yairi, T., & Machida, K. (2007). Change-point detection in time-series data based on subspace identification. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 559-564.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., . . . Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 3146-3154.

Khaidem, L., Saha, S., & Dey, S. R. (2016). *Predicting the direction of stock market prices using random forest*. arXiv preprint arXiv:1605.00003.

Kim, B. Y., & Han, H. (2022). Multi-Step-Ahead Forecasting of the CBOE Volatility Index in a Data-Rich Environment: Application of Random Forest with Boruta Algorithm.Kim, Byung Yeon, and Heejoon Han. *Korean Economic Review* , 541-569.

Kim, C.-J. (1993). Unobserved-Component Time Series Models With Markov-Switching Heteroscedasticity: Changes in Regime and the Link Between Inflation Rates and Inflation Uncertainty. *Journal of Business and Economic Statistics*, 341-349.

Kim, C.-J., & Nelson, C. R. (1999). *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*. MIT Press Books.

Kim, C.-J., Nelson, C. R., & Startz, R. (1998). Testing for mean reversion in heteroskedastic data based on Gibbs-sampling-augmented randomization. *Journal of Empirical Finance*, 131-154.

- Kim, C.-J., Piger, J., & Startz, R. (2003). Estimation of Markov Regime-Switching Regression Models with Endogenous Switching. *Journal of Econometrics*, 263-273.
- Kim, E.-c., Jeong, H.-w., & Lee, N.-y. (2019). Global Asset Allocation Strategy Using a Hidden Markov Model. *Journal of Risk and Financial Management*, 168-183.
- Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 307-319.
- Kim, S.-J., Koh, K., Boyd, S., & Gorinevsky, D. (2009).  $\ell_1$  Trend Filtering. *SIAM review*, 339-360.
- Kohonen, T. (2001). Learning vector quantization. In T. Kohonen, *Self-organizing maps* (pp. 245-261). Berlin, Heidelberg: Springer.
- Koski, A. (1996). Modelling ECG signals with hidden Markov models. *Artificial intelligence in medicine*, 453-471.
- Kristjanpoller, W., & Michell, K. (2018). A stock market risk forecasting model through integration of switching regime, ANFIS and GARCH techniques. *Applied soft computing*, 106-116.
- Kritzman, M., Page, S., & Turkington, D. (2012). Regime shifts: Implications for dynamic strategies (corrected). *Financial Analysts Journal*, 22-39.
- Krolzig, H.-M. (1997). *Markov-Switching Vector Autoregressions: Modelling, Statistical Inference, and Application to Business Cycle Analysis*. New York: Springer.
- Kumar, I., Dogra, K., Utreja, C., & Yadav, P. (2018). A comparative study of supervised machine learning algorithms for stock market trend prediction. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 1003-1007). IEEE.
- Kumar, M., Patel, N. R., & Woo, J. (2002). Clustering seasonality patterns in the presence of errors. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 557-563).
- Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the Boruta package. *Journal of statistical software*, 1-13.

Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, 159-178.

Lautier, D. (2002). *The Kalman filter in finance: An application to term structure models of commodity prices and a comparison between the simple and the extended filters*. Paris Dauphine University.

Lee, H.-T., & Yoder, J. (2007). A bivariate Markov regime switching GARCH approach to estimate time varying minimum variance hedge ratios. *Applied Economics*, 1253-1265.

Lee, H.-T., & Yoder, J. (2007). Optimal hedging with a regime-switching time-varying correlation GARCH model. *Journal of Futures Markets*, 495-516.

Lee, J. W. (2001). Stock price prediction using reinforcement learning. *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570)* (pp. 690-695). IEEE.

Lee, Y., Ow, L. T., & Ling, D. N. (2014). Hidden markov models for forex trends prediction. *2014 International Conference on Information Science & Applications (ICISA)* (pp. 1-4). IEEE.

Legey, L. F., Silva, E. A., & Silva, E. G. (2010). Forecasting oil price trends using wavelets and hidden Markov models. *Energy Economics*, 1507-1519.

Lempel, A., & Ziv, J. (1976). On the complexity of finite sequences. *IEEE Transactions on information theory* (pp. 75-81). IEEE.

Leutner, B. F., Reineking, B., Müller, J., Bachmann, M., Beierkuhnlein, C., Dech, S., & Wegmann, M. (2012). Modelling forest  $\alpha$ -diversity and floristic composition - On the added value of LiDAR plus hyperspectral remote sensing. *Remote Sensing*, 2818-2845.

Li, H., Dagli, C. H., & Enke, D. (2007). Short-term stock market timing prediction under reinforcement learning schemes. *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning* (pp. 233-240). IEEE.

- Li, X., Uysal, A. S., & Mulvey, J. M. (2022). Multi-period portfolio optimization using model predictive control with mean-variance and risk parity frameworks. *European Journal of Operational Research*, 1158-1176.
- Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., ... Deng, X. (2016). Empirical analysis: stock market prediction via extreme learning machine. *Neural Computing and Applications*, 67-78.
- Liang, Y., Thavaneswaran, A., Yu, N., Hoque, M. E., & Thulasiram, R. K. (2020). Dynamic data science applications in optimal profit algorithmic trading. *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 1314-1319.
- Lin, L., Fang, W., Xiaolong, X., & Shisheng, Z. (2017). Random forests-based extreme learning machine ensemble for multi-regime time series prediction. *Expert Systems with Applications*, 164-176.
- Lin, Y., Guo, H., & Hu, J. (2013). An SVM-based Approach for Stock Market Trend Prediction. *The 2013 international joint conference on neural networks (IJCNN)* (pp. 1-7). IEEE.
- Liu, D., & Zhang, L. (2010). China stock market regimes prediction with artificial neural network and markov regime switching. *Proceedings of the world congress on engineering* (pp. 378-383). London: WCE.
- Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance*, 1705-1770.
- Lombardi, M. J., & Ravazzolo, F. (2016). On the correlation between commodity and equity returns: Implications for portfolio allocation. *Journal of Commodity Markets*, 45-57.
- Low, R. K., Faff, R., & Aas, K. (2016). Enhancing mean-variance portfolio selection by modeling distributional asymmetries. *Journal of Economics and Business*, 49-72.
- López de Prado, M. (2016). Building Diversified Portfolios that Outperform Out of Sample. *The Journal of Portfolio Management*, 59-69.

López de Prado, M. (2018). *Advances in Financial Machine Learning*. New Jersey: Wiley.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 4765-4774.

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2018). *Consistent individualized feature attribution for tree ensembles*. arXiv preprint arXiv:1802.03888.

Man, X., & Chan, E. P. (2021). The best way to select features? Comparing mda, lime, and shap. *he Journal of Financial Data Science*, 127-139.

Marcucci, J. (2005). Forecasting Stock Market Volatility with Regime-Switching GARCH Models. *Studies in Nonlinear Dynamics & Econometrics*.

Maringer, D., & Ramtohul, T. (2012). Regime-switching recurrent reinforcement learning for investment decision making. *Computational Management Science*, 89-107.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 77-91.

Martinelli, R., & Rhoads, N. (2010). Predicting market data using the Kalman filter. *Technical Analysis of Stocks and Commodities*, 46-71.

McNally, S., Roche, J., & Caton, S. (2018). Predicting the price of bitcoin using machine learning. *26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing* (pp. 339-343). IEEE.

Mhatre, G., Almeida, O., Mhatre, D., & Joshi, P. (2008). Credit card fraud detection using hidden Markov model. *EEE Transactions on dependable and secure computing* (pp. 37-48). IEEE.

Mikaeil, A. M., Bin, G., Xuemei, B., & Zhijun, W. (2014). Hidden Markov and Markov Switching Model for Primary User Channel State Prediction in Cognitive Radio. *The 2014 2nd international conference on systems and informatics (ICSAI 2014)* (pp. 854-859). IEEE.

Milosevic, N. (2016). *Equity forecast: Predicting long term stock price movement using machine learning*. arXiv preprint arXiv:1603.00751.

- Ming, C.-L. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, 10896-10904.
- Mitra, S. K., & Rohit, A. (2020). Momentum Trading with the  $\ell_1$ -Filter: Are the Markets Efficient? *International Review of Finance*, 827-856.
- Moore, T., & Wang, P. (2007). Volatility in stock returns for new EU member states: Markov regime switching model. *International Review of Financial Analysis*, 282-292.
- Mullainathan, S., & Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives*, 87-106.
- Murphy, K. P. (1998). Switching kalman filters. *Compaq Cambridge Research Lab Tech. Report*.
- Musilek, P., & Gupta, M. M. (2000). Neural networks and fuzzy systems. In M. M. Gupta, *Soft Computing and Intelligent Systems: Theory and Applications* (pp. 137-160). San Diego: Academic Press.
- Naik, N., & Mohan, B. R. (2019). Optimal feature selection of technical indicator and stock prediction using machine learning technique. *ICETCE 2019: Emerging Technologies in Computer Engineering: Microservices in Big Data Analytics* (pp. 261-268). Springer.
- Nazário, R. T., Lima e Silva, J., Sobreiro, V. A., & Kimura, H. (2017). A literature review of technical analysis on stock markets. *The Quarterly Review of Economics and Finance*, 115-126.
- Neely, C. J., & Dueker, M. (2007). Can Markov switching models predict excess foreign exchange returns? *Journal of Banking & Finance*, 279-296.
- Neely, C. J., Rapach, D. E., Tu, J., & Zhou, G. (2014). Forecasting the Equity Risk Premium: The Role of Technical Indicators. *Management Science*, 1772-1791.
- Neftiçi, S. N. (1982). Optimal prediction of cyclical downturns. *Journal of Economic Dynamics and Control*, 225-241.

Nelson, C. R., Piger, J., & Zivot, E. (2001). Markov regime switching and unit-root tests. *Journal of Business & Economic Statistics*, 404-415.

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the Econometric Society*, 347-370.

Netzer, O., Lattin, J. M., & Srinivasan, V. (2008). A hidden Markov model of customer relationship dynamics. *Marketing science*, 185-204.

Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proceedings of the twenty-first international conference on Machine learning*, 78.

Ng, S. (2014). Boosting recessions. *Canadian Journal of Economics/Revue canadienne d'économique*, 1-34.

Nguyen, N. (2014). *Probabilistic methods in estimation and prediction of financial models*. The Florida State University.

Nguyen, N. (2018). Hidden Markov Model for Stock Trading. *International Journal of Financial Studies*, 36-53.

Nguyen, N., & Nguyen, D. (2015). Hidden Markov Model for Stock Selection. *Risks*, 455-473.

Niemira, M. P., & Klein, P. A. (1994). *Forecasting financial and economic cycles*. John Wiley & Sons.

Niu, Y. S., Hao, N., & Zhang, H. (2016). Multiple change-point detection: a selective overview. *Statistical Science*, 611-623.

Nystrup, P., Boyd, S., Madsen, H., & Lindström, E. (2019). Multi-period portfolio selection with drawdown control. *Annals of Operations Research*, 245-271.

Nystrup, P., Madsen, H., & Lindström, E. (2018). Dynamic portfolio optimization across hidden market regimes. *Quantitative Finance*, 83-95.

Oates, T., Firoiu, L., & Cohen, P. R. (1999). Clustering Time Series with Hidden Markov Models and Dynamic Time Warping. *Workshop on neural, symbolic and*

*reinforcement learning methods for sequence learning* (pp. 17-21). Stockholm: IJCAI-99.

Oprisor, R., & Kwon, R. (2020). Multi-period portfolio optimization with investor views under regime switching. *Journal of Risk and Financial Management*, 3.

Ormerod, P., Rosewell, B., & Phelps, P. (2013). Inflation/unemployment regimes and the instability of the Phillips curve. *Applied Economics*, 1519-1531.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 100-115.

Paoletta, M. S., Polak, P., & Walker, P. S. (2019). Regime switching dynamic correlations for asymmetric and fat-tailed conditional returns. *Journal of Econometrics*, 493-515.

Park, C.-H., & Irwin, S. H. (2007). What do we know about the profitability of technical analysis? *Journal of Economic Surveys*, 786-826.

Pasricha, G. K. (2006). *Kalman filter and its economic applications*.

Pavlov, V., & Hurn, S. (2012). Testing the profitability of moving-average rules as a portfolio selection strategy. *Pacific-Basin Finance Journal*, 825-842.

Pelletier, D. (2006). Regime switching for dynamic correlations. *Journal of Econometrics*, 445-473.

Piger, J. (2020). Turning points and classification. *Advanced Studies in Theoretical and Applied Econometrics*, 585-624.

Piryonesi, S. M., & El-Diraby, T. E. (2020). Role of data analytics in infrastructure asset management: Overcoming data size and quality problems. *Journal of Transportation Engineering, Part B: Pavements*.

Plakandaras, V., Cunado, J., Gupta, R., & Wohar, M. E. (2017). Do Leading Indicators Forecast U.S. Recessions? A Nonlinear Re-Evaluation Using Historical Data. *International Finance*, 289-316.

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, e1301.

Procacci, P. F., & Aste, T. (2019). Forecasting market states. *Quantitative Finance*, 1491-1498.

Qi, M. (2001). Predicting US recessions with leading indicators via neural network models. *International journal of forecasting*, 383-401.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77.2 (pp. 257-286). IEEE.

Rao, S., & Hong, J. (2010). *Analysis of hidden markov models and support vector machines in financial applications*. University of California at Berkeley.

Ratto, A. P., Merello, S., Oneto, L., Ma, Y., Malandri, L., & Cambria, E. (2018). Ensemble of Technical Analysis and Machine Learning for Market Trend Prediction. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 2090-2096). IEEE.

Ren, L., Wei, Y., Cui, J., & Du, Y. (2017). A sliding window-based multi-stage clustering and probabilistic forecasting approach for large multivariate time series data. *Journal of Statistical Computation and Simulation*, 2494-2508.

Reus, L., & Mulvey, J. M. (2016). Dynamic allocations for currency futures under switching regimes signals. *European Journal of Operational Research*, 85-93.

Richman, J. S., & Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 2039-2049.

Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, (pp. 41-46).

Rom, B. M., & Ferguson, K. W. (1994). Post-modern portfolio theory comes of age. *Journal of Investing*, 11-17.

Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock prices using LSTM. *International Journal of Science and Research (IJSR)*, 1754-1756.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 386–408.

- Rosset, S., & Zhu, J. (2007). Piecewise linear regularized solution paths. *The Annals of Statistics*, 1012-1030.
- Rotta, P. N., & Valls Pereira, P. L. (2016). Analysis of contagion from the dynamic conditional correlation model with Markov Regime switching. *Applied Economics*, 2367-2382.
- Roubaud, D., & Mohamed, A. (2018). Oil prices, exchange rates and stock markets under uncertainty and regime-switching. *Finance Research Letters*, 28-33.
- Ruiz-Franco, L., Jiménez-Gómez, M., & Lambis-Alandete, E. (2018). Trading Strategy on the Future Mini S & P 500. *International Journal of Applied Engineering Research*, 11018-11024.
- Salhi, K., Deaconua, M., Lejaya, A., Champagnat, N., & Navet, N. (2016). Regime switching model for financial data: empirical risk analysis. *Physica A: Statistical Mechanics and its Applications*, 148-157.
- Samé, A., Chamroukhi, F., Govaert, G., & Akni, P. (2011). Model-based clustering and segmentation of time series with changes in regime. *Advances in Data Analysis and Classification*, 301-321.
- Samuelson, P. A. (1975). Lifetime portfolio selection by dynamic stochastic programming. *Stochastic optimization models in finance*, 517-524.
- Schaller, H., & Van Norden, S. (1997). Regime switching in stock market returns. *Applied Financial Economics*, 177-191.
- Shapley, L. S. (1953). A value for n-person games. *Contributions to the theory of games.*, 307-317.
- Sheu, H.-J., & Lee, H.-T. (2014). Optimal Futures Hedging Under Multichain Markov Regime Switching. *Journal of Futures Markets*, 173-202.
- Sheu, H.-J., Lee, H.-T., & Lai, Y.-S. (2013). *A Markov regime switching GARCH model with realized measures of volatility for optimal futures hedging*. Retrieved from European Financial Management Association 2013 annual meeting paper: [https://www.efmaefm.org/0EFMAMEETINGS/EFMA%20ANNUAL%20MEETINGS/2013-Reading/papers/EFMA2013\\_0146\\_fullpaper.pdf](https://www.efmaefm.org/0EFMAMEETINGS/EFMA%20ANNUAL%20MEETINGS/2013-Reading/papers/EFMA2013_0146_fullpaper.pdf)

Shynkevich, A. (2012). Performance of technical analysis in growth and small cap segments of the US equity market. *Journal of Banking and Finance*, 193-208.

Sims, C. A., & Zha, T. (2006). Were there regime switches in U.S. monetary policy? *American Economic Review*, 54-81.

Škrinjarić, T., & Šego, B. (2016). Asset allocation and regime switching on Croatian financial market. *Croatian Operational Research Review*, 201-215.

Smets, F., & Wouters, R. (2007). Shocks and frictions in US business cycles: A Bayesian DSGE approach. *American economic review*, 586-606.

Srivastava, S., & Bhattacharyya, R. (2018, 03 20). *Evaluating the Building Blocks of a Dynamically Adaptive Systematic Trading Strategy*. Retrieved from SSRN: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3144169](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3144169)

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 1-21.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 647-665.

Sun, J., Fang, W., Wu, X., Lai, C.-H., & Xu, W. (2011). Solving the multi-stage portfolio optimization problem with a novel particle swarm optimization. *Expert Systems with Applications*, 6727-6735.

Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning*. Cambridge: MIT press.

Ta, V.-D., Liu, C.-M., & Addis, D. (2018). Prediction and portfolio optimization in quantitative trading using machine learning techniques. *Proceedings of the Ninth International Symposium on Information and Communication Technology* (pp. 98-105). Association for Computing Machinery.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 267-288.

- Timmermann, A. (2000). Moments of Markov switching models. *Journal of Econometrics*, 75-111.
- Tong, H., & Lim, K. S. (1980). Threshold autoregression, limit cycles and cyclical data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 245-268.
- Tsay, R. S. (1989). Testing and Modeling Threshold Autoregressive Processes. *Journal of the American Statistical Association*, 231-240.
- Tsay, R. S. (2010). *Analysis of Financial Time Series*. New Jersey: Wiley.
- Uysal, A. S., & Mulvey, J. M. (2021). A Machine Learning Approach in Regime-Switching Risk Parity Portfolio. *The Journal of Financial Data Science*, 87-108.
- van Staden, P. M., Dang, D.-M., & Forsyth, P. A. (2021). The surprising robustness of dynamic mean-variance portfolio optimization to model misspecification errors. *European Journal of Operational Research*, 774-792.
- Vishwakarma, K. P. (1994). Recognizing business cycle turning points by means of a neural network. *Computational Economics*, 175-185.
- Vo, L. H. (2014). Application of Kalman Filter on modelling interest rates. *Journal of Management Science*, 1-15.
- Vrontos, S. D., Galakis, J., & Vrontos, I. D. (2021). Modeling and predicting U.S. recessions using machine learning techniques. *International Journal of Forecasting*, 647-671.
- Wainer, J. (2016). *Comparison of 14 different families of classification algorithms on 115 binary datasets*. arXiv preprint arXiv:1606.00930.
- Walasek, R., & Gajda, J. (2021). Fractional differentiation and its use in machine learning. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, 270-277.
- Wang, C., Xiao, Z., Zhao, F.-S., Ni, D., & Li, L. (2019). Research on Economic Recession Prediction Model From the Multiple Behavioral Features Perspective. *IEEE Access*, (pp. 83363-83371).

Wang, J., Athanasopoulos, G., Hyndman, R. J., & Wang, S. (2016). Crude oil price forecasting based on internet concern using an extreme learning machine. *International Journal of Forecasting*, 665-677.

Wang, M., Lin, Y.-H., & Mikhelson, I. (2020). Regime-Switching Factor Investing with Hidden Markov Models. *Journal of Risk and Financial Management*, 311-326.

Wang, X., Zhang, H. H., & Wu, Y. (2019). Multiclass probability estimation with support vector machines. *Journal of Computational and Graphical Statistics*, 586-595.

Wang, Z., Li, K., Xia, S. Q., & Liu, H. (2021). *Economic Recession Prediction Using Deep Neural Network*. arXiv preprint arXiv:2107.10980.

Ward, F. (2017). Spotting the Danger Zone - Forecasting Financial Crises with Classification Tree Ensembles and Many Predictors. *Journal of Applied Econometrics*, 359-378.

Weigend, A. S., & Shi, S. (2000). Predicting daily probability distributions of S&P500 returns. *Journal of Forecasting*, 375-392.

Welch, I., & Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, 1455-1508.

Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics* (pp. 70-73). IEEE.

Wen, Q., Gao, J., Song, X., Sun, L., & Tan, J. (2019). *RobustTrend: A Huber loss with a combined first and second order difference regularization for time series trend filtering*. arXiv preprint arXiv:1906.03751.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE 78.10* (pp. 1550-1560). IEEE.

West, K. D., & Cho, D. (1995). The predictive ability of several models of exchange rate volatility. *Journal of Econometrics*, 367-391.

Whaley, R. E. (2009). Understanding the VIX. *The Journal of Portfolio Management*, 98-105.

- Wilms, I., Rombout, J., & Croux, C. (2021). Multivariate volatility forecasts for stock market indices. *International Journal of Forecasting*, 484-499.
- Wyckoff, R. D. (1937). *The Richard D. Wyckoff Method of Trading in Stocks*. New York: Wyckoff Associates, Inc.
- Xiaomao, X., Xudong, Z., & Yuanfang, W. (2019). A comparison of feature selection methodology for solving classification problems in finance. *Journal of Physics: Conference Series*, 012026.
- Yang, J., Xu, Y., & Chen, C. S. (1994). Hidden Markov model approach to skill learning and its application to telerobotics. *IEEE transactions on robotics and automation* (pp. 621-631). IEEE.
- Yazdani, A. (2020). Machine Learning Prediction of Recessions: An Imbalanced Classification Approach. *The Journal of Financial Data Science*, 21-32.
- Yentes, J. M., Hunt, N., Schmid, K. K., McGrath, D., & Stergiou, N. (2013). The appropriate use of approximate entropy and sample entropy with short data sets. *Annals of biomedical engineering*, 349-365.
- Yin, L., & Yang, Q. (2016). Predicting the oil prices: Do technical indicators help? *Energy Economics*, 338-350.
- Zakoian, J.-M. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics and Control*, 931-955.
- Zhu, Y., & Zhou, G. (2009). Technical analysis: an asset allocation perspective on the use of moving averages. *Journal of Financial Economics*, 519-544.
- Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 301-320.

# Appendix

## Appendix A

Table 31: Phase 1 (in-sample): annualised returns winning score ratio for equities

Equities	Proposed model	WQ mode 1	Marko v 2S	Marko v 3S	Marko v KNS	Proposed model 2S	WQ mode 12S	Marko v 3S 2S	Marko v KNS 2S
<b>MSCI USA</b>	0.	0.	0.	0.	0.	0.053	0.	0.	0.
<b>NASDAQ</b>	0.	0.	0.221	0.	0.	0.	0.	0.	0.
<b>MSCI EM</b>	0.	0.	0.	0.	0.	0.145	0.	0.	0.
<b>Stoxx 600</b>	0.	0.	0.	0.	0.	0.191	0.	0.	0.
<b>DAX</b>	0.	0.	0.088	0.	0.	0.	0.	0.	0.
<b>CAC40</b>	0.	0.	0.	0.	0.	0.121	0.	0.	0.
<b>FTSE100</b>	0.	0.	0.194	0.	0.	0.	0.	0.	0.
<b>Eurostoxx</b>	0.	0.	0.	0.	0.	0.111	0.	0.	0.
<b>HSI</b>	0.	0.	0.	0.	0.	0.024	0.	0.	0.
<b>Shenzhen</b>	0.	0.	0.	0.	0.	0.085	0.	0.	0.
<b>Nikkei</b>	0.	0.	0.	0.	0.	0.	0.	0.072	0.
<b>MSCI USA Cons Discr</b>	0.	0.	0.11	0.	0.	0.	0.	0.	0.
<b>MSCI USA Cons Staples</b>	0.	0.	0.03	0.	0.	0.	0.	0.	0.
<b>MSCI USA Energy</b>	0.	0.	0.081	0.	0.	0.	0.	0.	0.
<b>MSCI USA Financials</b>	0.	0.	0.031	0.	0.	0.	0.	0.	0.
<b>MSCI USA Industrials</b>	0.	0.	0.133	0.	0.	0.	0.	0.	0.
<b>MSCI USA Healthcare</b>	0.	0.	0.	0.	0.	0.166	0.	0.	0.
<b>MSCI USA Info Tech</b>	0.	0.	0.302	0.	0.	0.	0.	0.	0.
<b>MSCI USA Materials</b>	0.	0.	0.	0.	0.	0.006	0.	0.	0.
<b>MSCI USA Telcos</b>	0.	0.	0.234	0.	0.	0.	0.	0.	0.
<b>MSCI USA Utilities</b>	0.	0.	0.115	0.	0.	0.	0.	0.	0.
<b>TSX</b>	0.	0.	0.03	0.	0.	0.	0.	0.	0.
<b>ASX</b>	0.	0.	0.	0.	0.	0.077	0.	0.	0.
<b>FTSE MIB</b>	0.	0.	0.	0.	0.	0.256	0.	0.	0.
<b>AVERAGE</b>	0.	0.	<b>0.065</b>	0.	0.	0.051	0.	0.003	0.

Table 32: Phase 1 (in-sample): annualised returns winning score ratio for commodities

Commodities	Proposed model	WQ model	Markov 2S	Markov 3S	Markov KNS	Proposed model 2S	WQ model 2S	Markov 3S 2S	Markov KNS 2S
<b>Gold</b>	0.	0.	0.	0.	0.	0.15	0.	0.	0.
<b>Brent Crude</b>	0.	0.	0.304	0.	0.	0.	0.	0.	0.
<b>Copper</b>	0.	0.	0.	0.	0.	0.226	0.	0.	0.
<b>Natural Gas</b>	0.	0.	0.	0.	0.	0.	0.066	0.	0.
<b>Aluminium</b>	0.	0.	0.	0.	0.	0.394	0.	0.	0.
<b>Wheat</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.232

<b>Corn</b>	0.	0.	0.	0.	0.	0.076	0.	0.	0.
<b>Nickel</b>	0.	0.	0.	0.	0.	0.479	0.	0.	0.
<b>Live Cattle</b>	0.	0.	0.	0.	0.	0.	0.105	0.	0.
<b>Coffee</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.13
<b>Soybeans</b>	0.	0.	0.	0.	0.	0.	0.	0.032	0.
<b>Sugar</b>	0.	0.	0.	0.489	0.	0.	0.	0.	0.
<b>AVERAGE</b>	0.	0.	0.025	0.041	0.	<b>0.11</b>	0.014	0.003	0.03

Table 33: Phase 1 (in-sample): annualised returns winning score ratio for FX

FX	Proposed model	WQ model	Markov 2S	Markov 3S	Markov KNS	Proposed model 2S	WQ model 2S	Markov 3S 2S	Markov KNS 2S
<b>DXY</b>	0.	0.	0.053	0.	0.	0.	0.	0.	0.
<b>GBPUSD</b>	0.	0.	0.	0.	0.	0.	0.028	0.	0.
<b>EURUSD</b>	0.	0.	0.	0.	0.	0.	0.163	0.	0.
<b>USDJPY</b>	0.	0.	0.	0.	0.	0.237	0.	0.	0.
<b>AUDJPY</b>	0.	0.	0.	0.	0.	0.101	0.	0.	0.
<b>CHFUSD</b>	0.	0.	0.	0.	0.346	0.	0.	0.	0.
<b>USDCAD</b>	0.	0.	0.	0.	0.	0.138	0.	0.	0.
<b>USDAUD</b>	0.	0.	0.	0.	0.	0.536	0.	0.	0.
<b>USDNZD</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.03
<b>USDNOK</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.012
<b>USDSEK</b>	0.	0.	0.	0.	0.	0.	0.	0.071	0.
<b>USDDKK</b>	0.	0.	0.	0.	0.	0.	0.	0.135	0.
<b>EURRUB</b>	0.	0.	0.	0.	0.	0.16	0.	0.	0.
<b>AVERAGE</b>	0.	0.	0.004	0.	0.027	<b>0.09</b>	0.015	0.016	0.003

Table 34: Phase 1 (in-sample): annualised returns winning score ratio for fixed income ETFs

Fixed Income	Proposed model	WQ mode 1	Marko v 2S	Marko v 3S	Marko v KNS	Proposed model 2S	WQ mode 12S	Marko v 3S 2S	Marko v KNS 2S
<b>Long Bonds ETF</b>	0.	0.	0.154	0.	0.	0.	0.	0.	0.
<b>Short Bonds ETF</b>	0.	0.	0.089	0.	0.	0.	0.	0.	0.
<b>Ultra-Long Bonds ETF</b>	0.	0.	0.	0.	0.	0.	0.	0.474	0.
<b>TIPS Bonds ETF</b>	0.	0.	0.172	0.	0.	0.	0.	0.	0.
<b>EM Bonds ETF</b>	0.	0.	0.	0.	0.	0.327	0.	0.	0.
<b>HY Credit ETF</b>	0.	0.	0.342	0.	0.	0.	0.	0.	0.
<b>IG Credit ETF</b>	0.	0.	0.157	0.	0.	0.	0.	0.	0.
<b>AVERAGE</b>	0.	0.	<b>0.131</b>	0.	0.	0.047	0.	0.068	0.

Table 35: Phase 1 (in-sample): adjusted Sharpe ratio winning score ratio for equities

Equities	Proposed model	WQ mode 1	Marko v 2S	Marko v 3S	Marko v KNS	Proposed model 2S	WQ mode 12S	Marko v 3S 2S	Marko v KNS 2S
<b>MSCI USA</b>	0.	0.	0.	0.	0.	0.136	0.	0.	0.
<b>NASDAQ</b>	0.	0.	0.	0.	0.	0.135	0.	0.	0.
<b>MSCI EM</b>	0.	0.	0.	0.	0.	0.	0.066	0.	0.
<b>Stoxx 600</b>	0.	0.	0.	0.	0.	0.087	0.	0.	0.
<b>DAX</b>	0.	0.	0.	0.	0.	0.028	0.	0.	0.

CAC40	0.186	0.	0.	0.	0.	0.	0.	0.	0.	0.
FTSE100	0.	0.	0.	0.	0.	0.	0.029	0.	0.	0.
Eurostoxx	0.061	0.	0.	0.	0.	0.	0.	0.	0.	0.
HSI	0.	0.	0.	0.	0.	0.241	0.	0.	0.	0.
Shenzhen	0.	0.	0.	0.	0.	0.	0.	0.002	0.	0.
Nikkei	0.	0.	0.	0.	0.	0.	0.	0.378	0.	0.
MSCI USA Cons Discr	0.	0.	0.	0.	0.	0.209	0.	0.	0.	0.
MSCI USA Cons Staples	0.	0.	0.	0.	0.	0.124	0.	0.	0.	0.
MSCI USA Energy	0.	0.	0.	0.	0.	0.	0.057	0.	0.	0.
MSCI USA Financials	0.	0.	0.	0.	0.	0.	0.	0.027	0.	0.
MSCI USA Industrials	0.	0.	0.	0.	0.	0.055	0.	0.	0.	0.
MSCI USA Healthcare	0.	0.	0.	0.	0.	0.258	0.	0.	0.	0.
MSCI USA Info Tech	0.	0.	0.	0.	0.	0.144	0.	0.	0.	0.
MSCI USA Materials	0.	0.	0.	0.	0.	0.034	0.	0.	0.	0.
MSCI USA Telcos	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.051
MSCI USA Utilities	0.	0.	0.019	0.	0.	0.	0.	0.	0.	0.
TSX	0.	0.	0.	0.	0.	0.192	0.	0.	0.	0.
ASX	0.	0.	0.	0.	0.	0.037	0.	0.	0.	0.
FTSE MIB	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.003
AVERAGE	0.01	0.	0.001	0.	0.	<b>0.07</b>	0.006	0.017	0.002	

Table 36: Phase 1 (in-sample): adjusted Sharpe ratio winning score ratio for commodities

Commodities	Proposed model	WQ model	Markov 2S	Markov 3S	Markov KNS	Proposed model 2S	WQ model 2S	Markov 3S 2S	Markov KNS 2S
Gold	0.	0.	0.	0.	0.	0.274	0.	0.	0.
Brent Crude	0.	0.	0.	0.	0.	0.	0.224	0.	0.
Copper	0.	0.	0.	0.	0.	0.	0.135	0.	0.
Natural Gas	0.	0.	0.	0.	0.	0.	0.135	0.	0.
Aluminium	0.	0.	0.327	0.	0.	0.	0.	0.	0.
Wheat	0.	0.	0.	0.	0.	0.	0.	0.377	0.
Corn	0.	0.	0.	0.	0.	0.	0.	0.576	0.
Nickel	0.	0.	0.	0.	0.	0.09	0.	0.	0.
Live Cattle	0.	0.	0.	0.	0.	0.	0.058	0.	0.
Coffee	0.	0.	0.	0.	0.	0.	0.	0.	0.207
Soybeans	0.	0.	0.	0.	0.	0.	0.	0.	0.281
Sugar	0.	0.	0.	0.	0.	0.	0.	0.225	0.
AVERAGE	0.	0.	0.027	0.	0.	0.03	0.046	<b>0.098</b>	0.041

Table 37: Phase 1 (in-sample): adjusted Sharpe ratio winning score ratio for FX

FX	Proposed model	WQ model	Markov 2S	Markov 3S	Markov KNS	Proposed model 2S	WQ model 2S	Markov 3S 2S	Markov KNS 2S
DXY	0.	0.	0.057	0.	0.	0.	0.	0.	0.
GBPUSD	0.	0.	0.	0.	0.	0.	0.275	0.	0.
EURUSD	0.	0.	0.	0.	0.	0.	0.	0.087	0.
USDJPY	0.	0.	0.	0.	0.	0.296	0.	0.	0.
AUDJPY	0.	0.	0.	0.	0.	0.092	0.	0.	0.

<b>CHFUSD</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.019	0.
<b>USDCAD</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.106
<b>USDAUD</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.311
<b>USDNZD</b>	0.	0.	0.	0.	0.013	0.	0.	0.	0.	0.
<b>USDNOK</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.018	0.
<b>USDSEK</b>	0.	0.	0.171	0.	0.	0.	0.	0.	0.	0.
<b>USDDKK</b>	0.112	0.	0.	0.	0.	0.	0.	0.	0.	0.
<b>EURRUB</b>	0.	0.	0.233	0.	0.	0.	0.	0.	0.	0.
<b>AVERAGE</b>	0.009	0.	<b>0.036</b>	0.	0.001	0.03	0.021	0.01	0.032	

Table 38: Phase 1 (in-sample): adjusted Sharpe ratio winning score ratio for fixed income ETFs

Fixed Income	Proposed model	WQ mode 1	Marko v 2S	Marko v 3S	Marko v KNS	Proposed model 2S	WQ mode 12S	Marko v 3S 2S	Marko v KNS 2S
<b>Long Bonds ETF</b>	0.	0.	0.	0.082	0.	0.	0.	0.	0.
<b>Short Bonds ETF</b>	0.	0.	0.	0.	0.	0.089	0.	0.	0.
<b>Ultra-Long Bonds ETF</b>	0.	0.	0.	0.	0.	0.	0.	0.266	0.
<b>TIPS Bonds ETF</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.04
<b>EM Bonds ETF</b>	0.	0.	0.313	0.	0.	0.	0.	0.	0.
<b>HY Credit ETF</b>	0.	0.	0.192	0.	0.	0.	0.	0.	0.
<b>IG Credit ETF</b>	0.	0.	0.	0.	0.	0.	0.	0.	0.011
<b>AVERAGE</b>	0.	0.	<b>0.072</b>	0.012	0.	0.013	0.	0.038	0.007

Table 39: Phase 2 (out-of-sample): annualised returns winning score ratio for equities

Equities	Markov 2S	Proposed model 2S	Markov 3S 2S
<b>MSCI USA</b>	0.	0.137	0.
<b>NASDAQ</b>	0.	0.288	0.
<b>MSCI EM</b>	0.	0.235	0.
<b>Stoxx 600</b>	0.331	0.	0.
<b>DAX</b>	0.149	0.	0.
<b>CAC40</b>	0.	0.202	0.
<b>FTSE100</b>	0.	0.021	0.
<b>Eurostoxx</b>	0.	0.077	0.
<b>HSI</b>	0.	0.171	0.
<b>Shenzhen</b>	0.	0.374	0.
<b>Nikkei</b>	0.	0.116	0.
<b>MSCI USA Cons Discr</b>	0.127	0.	0.
<b>MSCI USA Cons Staples</b>	0.132	0.	0.
<b>MSCI USA Energy</b>	0.	0.	0.609
<b>MSCI USA Financials</b>	0.219	0.	0.
<b>MSCI USA Industrials</b>	0.283	0.	0.
<b>MSCI USA Healthcare</b>	0.	0.	0.098
<b>MSCI USA Info Tech</b>	0.012	0.	0.
<b>MSCI USA Materials</b>	0.037	0.	0.
<b>MSCI USA Telcos</b>	0.	0.064	0.
<b>MSCI USA Utilities</b>	0.	0.033	0.
<b>TSX</b>	0.351	0.	0.
<b>ASX</b>	0.224	0.	0.

<b>FTSE MIB</b>	0.	0.587	0.
<b>AVERAGE</b>	0.078	<b>0.096</b>	0.029

Table 40: Phase 2 (out-of-sample): annualised returns winning score ratio for commodities

Commodities	Markov 2S	Proposed model 2S	Markov 3S 2S
<b>Gold</b>	0.	0.85	0.
<b>Brent Crude</b>	0.	0.	0.11
<b>Copper</b>	0.01	0.	0.
<b>Natural Gas</b>	0.	0.355	0.
<b>Aluminium</b>	0.	0.277	0.
<b>Wheat</b>	0.	0.	0.929
<b>Corn</b>	0.	0.454	0.
<b>Nickel</b>	0.	0.405	0.
<b>Live Cattle</b>	0.	0.	0.141
<b>Coffee</b>	0.19	0.	0.
<b>Soybeans</b>	0.	0.	0.243
<b>Sugar</b>	0.	0.875	0.
<b>AVERAGE</b>	0.017	<b>0.268</b>	0.119

Table 41: Phase 2 (out-of-sample): annualised returns winning score ratio for FX

FX	Markov 2S	Proposed model 2S	Markov 3S 2S
<b>DXY</b>	0.958	0.	0.
<b>GBPUSD</b>	0.443	0.	0.
<b>EURUSD</b>	0.	0.443	0.
<b>USDJPY</b>	0.883	0.	0.
<b>AUDJPY</b>	0.	0.195	0.
<b>CHFUSD</b>	0.	0.	0.265
<b>USDCAD</b>	0.163	0.	0.
<b>USDAUD</b>	0.	0.484	0.
<b>USDNZD</b>	0.401	0.	0.
<b>USDNOK</b>	0.	0.096	0.
<b>USDSEK</b>	0.	0.218	0.
<b>USDDKK</b>	0.	0.251	0.
<b>EURRUB</b>	0.	0.845	0.
<b>AVERAGE</b>	0.219	<b>0.195</b>	0.02

Table 42: Phase 2 (out-of-sample): annualised returns winning score ratio for fixed income ETFs

Fixed Income	Markov 2S	Proposed model 2S	Markov 3S 2S
<b>Long Bonds ETF</b>	0.3	0.	0.
<b>Short Bonds ETF</b>	0.	0.24	0.
<b>Ultra-Long Bonds ETF</b>	0.	0.639	0.
<b>TIPS Bonds ETF</b>	0.486	0.	0.
<b>EM Bonds ETF</b>	0.651	0.	0.
<b>HY Credit ETF</b>	0.341	0.	0.
<b>IG Credit ETF</b>	0.	0.367	0.
<b>AVERAGE</b>	<b>0.254</b>	0.178	0.

Table 43: Phase 2 (out-of-sample): adjusted Sharpe ratio winning score ratio for equities

Equities	Markov 2S	Proposed model 2S	Markov 3S 2S
MSCI USA	0.	0.22	0.
NASDAQ	0.	0.513	0.
MSCI EM	0.	0.515	0.
Stoxx 600	0.	0.505	0.
DAX	0.	0.485	0.
CAC40	0.	0.645	0.
FTSE100	0.	0.674	0.
Eurostoxx	0.	0.829	0.
HSI	0.	0.397	0.
Shenzhen	0.	0.556	0.
Nikkei	0.	0.218	0.
MSCI USA Cons Discr	0.	0.297	0.
MSCI USA Cons Staples	0.135	0.	0.
MSCI USA Energy	0.	0.	0.524
MSCI USA Financials	0.144	0.	0.
MSCI USA Industrials	0.	0.026	0.
MSCI USA Healthcare	0.	0.	0.085
MSCI USA Info Tech	0.	0.522	0.
MSCI USA Materials	0.	0.244	0.
MSCI USA Telcos	0.	0.77	0.
MSCI USA Utilities	0.508	0.	0.
TSX	0.	0.22	0.
ASX	0.	0.353	0.
FTSE MIB	0.	0.	0.23
AVERAGE	0.033	<b>0.333</b>	0.035

Table 44: Phase 2 (out-of-sample): adjusted Sharpe ratio winning score ratio for commodities

Commodities	Markov 2S	Proposed model 2S	Markov 3S 2S
Gold	0.	0.178	0.
Brent Crude	0.	0.	0.187
Copper	0.836	0.	0.
Natural Gas	0.	0.584	0.
Aluminium	0.	0.	0.19
Wheat	0.	0.	0.693
Corn	0.497	0.	0.
Nickel	0.812	0.	0.
Live Cattle	0.	0.	0.923
Coffee	0.866	0.	0.
Soybeans	0.	0.	0.828
Sugar	0.	0.853	0.
AVERAGE	<b>0.251</b>	0.135	0.235

Table 45: Phase 2 (out-of-sample): adjusted Sharpe ratio winning score ratio for FX

FX	Markov 2S	Proposed model 2S	Markov 3S 2S
DXY	0.	0.	0.698
GBPUSD	0.892	0.	0.

<b>EURUSD</b>	0.	0.	0.169
<b>USDJPY</b>	0.297	0.	0.
<b>AUDJPY</b>	0.	0.872	0.
<b>CHFUSD</b>	0.	0.	0.469
<b>USDCAD</b>	0.	0.	0.868
<b>USDAUD</b>	0.	0.414	0.
<b>USDNZD</b>	0.	0.554	0.
<b>USDNOK</b>	0.	0.	0.701
<b>USDSEK</b>	0.	0.	0.819
<b>USDDKK</b>	0.	0.122	0.
<b>EURRUB</b>	0.	0.	0.087
<b>AVERAGE</b>	0.092	0.151	<b>0.293</b>

Table 46: Phase 2 (out-of-sample): adjusted Sharpe ratio winning score ratio for fixed income ETFs

<b>Fixed Income</b>	<b>Markov 2S</b>	<b>Proposed model 2S</b>	<b>Markov 3S 2S</b>
<b>Long Bonds ETF</b>	0.	0.208	0.
<b>Short Bonds ETF</b>	0.	0.644	0.
<b>Ultra-Long Bonds ETF</b>	0.343	0.	0.
<b>TIPS Bonds ETF</b>	0.	0.148	0.
<b>EM Bonds ETF</b>	0.084	0.	0.
<b>HY Credit ETF</b>	0.	0.561	0.
<b>IG Credit ETF</b>	0.	0.	0.08
<b>AVERAGE</b>	0.061	<b>0.223</b>	0.011

## Appendix B

Table 47: Feature importances of the Equity Random Forest model.

“Bullish” regime		“Bearish” regime		“Other” regime	
Features	Mean Shap value	Features	Mean Shap value	Features	Mean Shap value
Last_location_of_maximum	0.0005	ROC_6M	0.0172	Time_reversal_asymmetry_statistic_lag_1	0.0064
Minus_DI	0.0003	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_9	0.0167	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_9	0.0043
First_location_of_maximum	0.0002	Time_reversal_asymmetry_statistic_lag_1	0.0092	Time_reversal_asymmetry_statistic_lag_3	0.0042
Mean_Std_Month	0.0001	Time_reversal_asymmetry_statistic_lag_3	0.0064	Augmented_dickey_fuller_attr_“pvalue”_autolag_“AI C”	0.0041
Plus_DI	0.0000	Time_reversal_asymmetry_statistic_lag_2	0.0053	ROC_12M	0.0041
MFI	0.0000	ROC_12M	0.0047	Augmented_dickey_fuller_attr_“teststat”_autolag_“AI C”	0.0030
10Y bond yield	0.0000	Force_Inx	0.0045	Mean_change	0.0026
EMBI	0.0000	Mean_change	0.0029	Last_location_of_maximum	0.0015
US CPI	0.0000	CMO	0.0028	Time_reversal_asymmetry_statistic_lag_2	0.0013
Ar_coefficient_coeff_1_k_10	0.0000	ROC_1M	0.0026	Ulcer	0.0011
Cid_ce_normalize_True	0.0000	RSI	0.0025	First_location_of_maximum	0.0011
Yield Curve 2-3M	0.0000	Index_mass_quantile_q_0_7	0.0021	Ar_coefficient_coeff_0_k_10	0.0009
Fx_rate	0.0000	Index_mass_quantile_q_0_8	0.0020	ATR	0.0007
Autocorrelation_lag_8	0.0000	ATR	0.0019	Index_mass_quantile_q_0_8	0.0006
Longest_strike_above_mean	0.0000	Ulcer	0.0018	Index_mass_quantile_q_0_7	0.0005
Fourier_entropy_bins_3	0.0000	Awesome	0.0015	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_0	0.0005
Agg_autocorrelation_f_agg_“var”_maxlag_40	0.0000	Augmented_dickey_fuller_attr_“pvalue”_autolag_“AI C”	0.0014	TRIX	0.0003
CRB Spot	0.0000	Augmented_dickey_fuller_attr_“teststat”_autolag_“AI C”	0.0013	Kyle_L	0.0002
Mean_Std_Year	0.0000	Plus_DI	0.0012	Cid_ce_normalize_True	0.0002
Cid_ce_normalize_False	0.0000	CCI	0.0009	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_1	0.0001
Ar_coefficient_coeff_3_k_10	0.0000	Last_location_of_minimum	0.0006	VIX	0.0001
PE	0.0000	BB	0.0006	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_8	0.0001
Lempel_ziv_complexity_bins_3	0.0000	Kyle_L	0.0005	Fft_aggregated_aggrtype_“c entroid”	0.0001
Ar_coefficient_coeff_5_k_10	0.0000	TRIX	0.0005	MOVE	0.0001
Lempel_ziv_complexity_bins_10	0.0000	Credit_ytw	0.0004	Autocorrelation_lag_7	0.0001
Spkt_welch_density_coeff_5	0.0000	Minus_DI	0.0004	ROC_6M	0.0001

Skewness	0.0000	First_location_of_minimum	0.0003	EMBI	0.0001
Fourier_entropy_bins_10	0.0000	Max_Rets_1M	0.0001	Agg_autocorrelation_f_agg_"median"_maxlag_40	0.0000
Ar_coefficient_coeff_9_k_10	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_0	0.0001	Ar_coefficient_coeff_8_k_10	0.0000
Permutation_entropy_dim ension_4_tau_1	0.0000	MFI	0.0001	Global_credit_ytw	0.0000
Spkt_welch_density_coeff_8	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_1	0.0001	PE	0.0000
High	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_5	0.0001	Mean_abs_change	0.0000
Close	0.0000	Ar_coefficient_coeff_0_k_10	0.0001	First_location_of_minimum	0.0000
Lempel_ziv_complexity_bis_5	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_4	0.0001	Autocorrelation_lag_4	0.0000
ADXR	0.0000	10Y bond yield	0.0000	Ultimate	0.0000
Permutation_entropy_dim ension_5_tau_1	0.0000	Fft_aggregated_aggtpe_"skew"	0.0000	LIBOR-OIS Spread	0.0000
Ar_coefficient_coeff_10_k_10	0.0000	Lempel_ziv_complexity_bis_3	0.0000	Fourier_entropy_bins_100	0.0000
BDI	0.0000	MOVE	0.0000	Euro CPI	0.0000
Count_below_mean	0.0000	Skewness	0.0000	Agg_autocorrelation_f_agg_"mean"_maxlag_40	0.0000
Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_3	0.0000	US10Y yield	0.0000	Ar_coefficient_coeff_2_k_10	0.0000
Count_above_mean	0.0000	Ar_coefficient_coeff_8_k_10	0.0000	Variation_coefficient	0.0000
Partial_autocorrelation_la g_6	0.0000	Lempel_ziv_complexity_bis_10	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_2	0.0000
Partial_autocorrelation_la g_9	0.0000	Variation_coefficient	0.0000	TED Spread	0.0000
Partial_autocorrelation_la g_3	0.0000	LIBOR-OIS Spread	0.0000	Yield Curve 2-3M	0.0000
OBV	0.0000	Fft_aggregated_aggtpe_"kurtosis"	0.0000	Spkt_welch_density_coeff_5	0.0000
Ar_coefficient_coeff_6_k_10	0.0000	Lempel_ziv_complexity_bis_5	0.0000	Partial_autocorrelation_la g_1	0.0000
Permutation_entropy_dim ension_6_tau_1	0.0000	Euro CPI	0.0000	CRB Spot	0.0000
Ar_coefficient_coeff_7_k_10	0.0000	Mean_abs_change	0.0000	DY	0.0000
Partial_autocorrelation_la g_2	0.0000	STC	0.0000	Autocorrelation_lag_8	0.0000
Permutation_entropy_dim ension_3_tau_1	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_6	0.0000	Autocorrelation_lag_6	0.0000
Augmented_dickey_fuller_attr_"usedlag"_autolag_"AIC"	0.0000	Ar_coefficient_coeff_9_k_10	0.0000	MACD	0.0000
Binned_entropy_max_bins_10	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_7	0.0000	Autocorrelation_lag_3	0.0000
Fourier_entropy_bins_5	0.0000	Fft_aggregated_aggtpe_"variance"	0.0000	Fft_aggregated_aggtpe_"variance"	0.0000
PS	0.0000	Permutation_entropy_dim ension_4_tau_1	0.0000	Fourier_entropy_bins_3	0.0000
Kurtosis	0.0000	Mean_Std_Month	0.0000	Agg_autocorrelation_f_agg_"var"_maxlag_40	0.0000
PB	0.0000	Volume	0.0000	Autocorrelation_lag_1	0.0000
Lempel_ziv_complexity_bis_2	0.0000	DY	0.0000	Fft_aggregated_aggtpe_"kurtosis"	0.0000
Ar_coefficient_coeff_4_k_10	0.0000	Spkt_welch_density_coeff_2	0.0000	Spkt_welch_density_coeff_2	0.0000

Permutation_entropy_dim_en_sion_7_tau_1	0.0000	Number_cwt_peaks_n_5	0.0000	Low	0.0000
Longest_strike_below_mean	0.0000	Spkt_welch_density_coeff_8	0.0000	Autocorrelation_lag_9	0.0000
Autocorrelation_lag_2	0.0000	Mean_Std_Year	0.0000	ADXR	0.0000
Number_cwt_peaks_n_1	0.0000	Ar_coefficient_coeff_6_k_10	0.0000	High	0.0000
Autocorrelation_lag_9	0.0000	Ar_coefficient_coeff_7_k_10	0.0000	Fourier_entropy_bins_10	0.0000
DXY	0.0000	Ar_coefficient_coeff_10_k_10	0.0000	DXY	0.0000
Energy_ratio_by_chunks_n_um_segments_10_segm_en_t_focus_6	0.0000	Permutation_entropy_dim_en_sion_3_tau_1	0.0000	Close	0.0000
Ar_coefficient_coeff_2_k_10	0.0000	Kurtosis	0.0000	Autocorrelation_lag_2	0.0000
Autocorrelation_lag_6	0.0000	Permutation_entropy_dim_en_sion_6_tau_1	0.0000	Volume	0.0000
Number_cwt_peaks_n_5	0.0000	Augmented_dickey_fuller_attr_"usedlag"_autolag_"AIC"	0.0000	Longest_strike_below_mean	0.0000
Autocorrelation_lag_3	0.0000	Partial_autocorrelation_la_g_3	0.0000	Number_cwt_peaks_n_5	0.0000
Partial_autocorrelation_la_g_1	0.0000	Partial_autocorrelation_la_g_6	0.0000	STC	0.0000
Fourier_entropy_bins_100	0.0000	Fourier_entropy_bins_5	0.0000	Number_cwt_peaks_n_1	0.0000
Energy_ratio_by_chunks_n_um_segments_10_segm_en_t_focus_7	0.0000	Partial_autocorrelation_la_g_9	0.0000	Energy_ratio_by_chunks_n_um_segments_10_segm_en_t_focus_7	0.0000
Low	0.0000	Count_above_mean	0.0000	Permutation_entropy_dim_en_sion_5_tau_1	0.0000
VIX	0.0000	Count_below_mean	0.0000	Cid_ce_normalize_False	0.0000
Volume	0.0000	Partial_autocorrelation_la_g_2	0.0000	Count_above_mean	0.0000
Autocorrelation_lag_1	0.0000	Permutation_entropy_dim_en_sion_7_tau_1	0.0000	Partial_autocorrelation_la_g_2	0.0000
US10Y yield	0.0000	Binned_entropy_max_bins_10	0.0000	Partial_autocorrelation_la_g_9	0.0000
MACD	0.0000	PB	0.0000	Partial_autocorrelation_la_g_6	0.0000
Global_credit_ytw	0.0000	Lempel_ziv_complexity_bis_2	0.0000	Binned_entropy_max_bins_10	0.0000
Energy_ratio_by_chunks_n_um_segments_10_segm_en_t_focus_4	0.0000	OBV	0.0000	Augmented_dickey_fuller_attr_"usedlag"_autolag_"AIC"	0.0000
Spkt_welch_density_coeff_2	0.0000	Ar_coefficient_coeff_4_k_10	0.0000	Count_below_mean	0.0000
STC	0.0000	BDI	0.0000	Partial_autocorrelation_la_g_3	0.0000
TED Spread	0.0000	Energy_ratio_by_chunks_n_um_segments_10_segm_en_t_focus_3	0.0000	Ar_coefficient_coeff_4_k_10	0.0000
Ultimate	0.0000	PS	0.0000	Lempel_ziv_complexity_bis_2	0.0000
Fft_aggregated_aggtpe_"variance"	0.0000	Number_cwt_peaks_n_1	0.0000	Fourier_entropy_bins_5	0.0000
DY	0.0000	Ar_coefficient_coeff_3_k_10	0.0000	Permutation_entropy_dim_en_sion_7_tau_1	0.0000
Energy_ratio_by_chunks_n_um_segments_10_segm_en_t_focus_2	0.0000	Ar_coefficient_coeff_5_k_10	0.0000	Ar_coefficient_coeff_6_k_10	0.0000
Agg_autocorrelation_f_agg_mean_maxlag_40	0.0000	Permutation_entropy_dim_en_sion_5_tau_1	0.0000	Kurtosis	0.0000
Fft_aggregated_aggtpe_"kurtosis"	0.0000	Fft_aggregated_aggtpe_c_entroid	0.0000	Energy_ratio_by_chunks_n_um_segments_10_segm_en_t_focus_3	0.0000
Fft_aggregated_aggtpe_"skew"	0.0000	Longest_strike_below_mean	0.0000	Ar_coefficient_coeff_10_k_10	0.0000
Autocorrelation_lag_4	0.0000	Longest_strike_above_mean	0.0000	PS	0.0000

Agg_autocorrelation_f_agg "median" _maxlag_40	0.0000	Autocorrelation_lag_2	0.0000	Permutation_entropy_dimen sion_3_tau_1	0.0000
Energy_ratio_by_chunks_n um_segments_10_segm ent_focus_8	0.0000	Low	0.0000	Ar_coefficient_coeff_7_k_10	0.0000
Credit_ytw	0.0000	Cid_ce_normalize_False	0.0000	Permutation_entropy_dimen sion_6_tau_1	0.0000
Autocorrelation_lag_7	0.0000	Close	0.0000	OBV	0.0000
Variation_coefficient	0.0000	MACD	0.0000	BDI	0.0000
Euro CPI	0.0000	DXY	0.0000	PB	0.0000
LIBOR-OIS Spread	0.0000	Autocorrelation_lag_1	0.0000	Spkt_welch_density_coeff_8	0.0000
Mean_abs_change	-0.0001	ADXR	0.0000	Ar_coefficient_coeff_5_k_10	0.0000
Ar_coefficient_coeff_8_k_10	-0.0001	Fourier_entropy_bins_10	0.0000	Ar_coefficient_coeff_3_k_10	0.0000
BB	-0.0001	High	0.0000	Mean_Std_Year	0.0000
Energy_ratio_by_chunks_n um_segments_10_segm ent_focus_5	-0.0001	Fx_rate	0.0000	Permutation_entropy_dimen sion_4_tau_1	0.0000
Max_Rets_1M	-0.0001	Energy_ratio_by_chunks_n um_segments_10_segm ent_focus_2	0.0000	Longest_strike_above_mea n	0.0000
Fft_aggregated_aggtpe_c entroid"	-0.0001	Autocorrelation_lag_9	0.0000	Ar_coefficient_coeff_9_k_10	0.0000
MOVE	-0.0001	Autocorrelation_lag_3	0.0000	Fx_rate	0.0000
Energy_ratio_by_chunks_n um_segments_10_segm ent_focus_1	-0.0002	TED Spread	0.0000	US CPI	0.0000
First_location_of_minimum	-0.0003	Agg_autocorrelation_f_agg "mean" _maxlag_40	0.0000	Energy_ratio_by_chunks_n um_segments_10_segm ent_focus_6	0.0000
CCI	-0.0003	Autocorrelation_lag_6	0.0000	Ar_coefficient_coeff_1_k_10	0.0000
Last_location_of_minimum	-0.0005	Autocorrelation_lag_4	0.0000	Fft_aggregated_aggtpe_s kew"	0.0000
Energy_ratio_by_chunks_n um_segments_10_segm ent_focus_0	-0.0006	Partial_autocorrelation_la g_1	0.0000	Energy_ratio_by_chunks_n um_segments_10_segm ent_focus_5	0.0000
Kyle_L	-0.0007	Ar_coefficient_coeff_1_k_10	0.0000	Lempel_ziv_complexity_b ins_5	0.0000
TRIX	-0.0008	Agg_autocorrelation_f_agg "var" _maxlag_40	0.0000	US10Y yield	0.0000
Ar_coefficient_coeff_0_k_10	-0.0009	Fourier_entropy_bins_3	0.0000	Lempel_ziv_complexity_b ins_10	0.0000
Awesome	-0.0012	Agg_autocorrelation_f_agg "median" _maxlag_40	0.0000	Skewness	0.0000
RSI	-0.0013	Autocorrelation_lag_7	0.0000	Last_location_of_minimum	0.0000
CMO	-0.0017	Spkt_welch_density_coeff_5	0.0000	Max_Rets_1M	0.0000
ROC_1M	-0.0020	US CPI	0.0000	Lempel_ziv_complexity_b ins_3	0.0000
Force_Inx	-0.0023	CRB Spot	0.0000	Energy_ratio_by_chunks_n um_segments_10_segm ent_focus_4	-0.0001
Index_mass_quantile_q_0. 7	-0.0025	Ultimate	0.0000	Mean_Std_Month	-0.0001
Index_mass_quantile_q_0. 8	-0.0025	Autocorrelation_lag_8	0.0000	10Y bond yield	-0.0001
ATR	-0.0026	Ar_coefficient_coeff_2_k_10	0.0000	MFI	-0.0001
Ulcer	-0.0029	Fourier_entropy_bins_100	0.0000	Awesome	-0.0002
Augmented_dickey_fuller_ attr_"teststat" _autolag_AIC	-0.0043	Yield Curve 2-3M	0.0000	Credit_ytw	-0.0003
Mean_change	-0.0055	Global_credit_ytw	0.0000	BB	-0.0005

Augmented_dickey_fuller_attr_ "pvalue" _autolag_ "AI C"	-0.0055	PE	0.0000	CCI	-0.0006
Time_reversal_asymmetry_statistic_lag_2	-0.0066	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_8	-0.0001	ROC_1M	-0.0006
ROC_12M	-0.0087	EMBI	-0.0001	Minus_DI	-0.0006
Time_reversal_asymmetry_statistic_lag_3	-0.0106	VIX	-0.0001	CMO	-0.0011
Time_reversal_asymmetry_statistic_lag_1	-0.0156	Cid_ce_normalize_True	-0.0002	Plus_DI	-0.0012
ROC_6M	-0.0172	First_location_of_maximum	-0.0012	RSI	-0.0012
Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_9	-0.0210	Last_location_of_maximum	-0.0020	Force_Inx	-0.0022

Table 48: Feature importances of the Commodity Random Forest model.

“Bullish” regime		“Bearish” regime		“Other” regime	
Features	Mean Shap value	Features	Mean Shap value	Features	Mean Shap value
Ultimate	0.0001	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_9	0.0220	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_9	0.0017
First_location_of_maximum	0.0000	CMO	0.0107	RSI	0.0013
High	0.0000	RSI	0.0098	TRIX	0.0004
Low	0.0000	ROC_6M	0.0092	Index_mass_quantile_q_0. 7	0.0003
Close	0.0000	Mean_change	0.0074	Augmented_dickey_fuller_attr_ "teststat" _autolag_ "AI C"	0.0003
Ar_coefficient_coeff_5_k_10	0.0000	Time_reversal_asymmetry_statistic_lag_1	0.0046	Ar_coefficient_coeff_0_k_10	0.0003
Index_mass_quantile_q_0. 9	0.0000	ROC_12M	0.0045	Last_location_of_maximum	0.0002
Variation_coefficient	0.0000	Time_reversal_asymmetry_statistic_lag_2	0.0043	Lempel_ziv_complexity_bis_10	0.0002
Fft_aggregated_aggtype_ "variance"	0.0000	Time_reversal_asymmetry_statistic_lag_3	0.0036	Augmented_dickey_fuller_attr_ "pvalue" _autolag_ "AI C"	0.0002
Fft_aggregated_aggtype_ "centroid"	0.0000	Force_Inx	0.0018	CCI	0.0002
Fft_aggregated_aggtype_ "kurtosis"	0.0000	Awesome	0.0013	First_location_of_minimum	0.0001
MACD	0.0000	TRIX	0.0010	First_location_of_maximum	0.0001
Mean_Std_Month	0.0000	BB	0.0006	Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_8	0.0001
Index_mass_quantile_q_0. 8	0.0000	ROC_1M	0.0006	ROC_1M	0.0001
Ar_coefficient_coeff_6_k_10	0.0000	CCI	0.0004	Awesome	0.0001
Count_below_mean	0.0000	Index_mass_quantile_q_0. 7	0.0003	Cid_ce_normalize_False	0.0000
Spkt_welch_density_coeff_2	0.0000	First_location_of_minimum	0.0001	Ultimate	0.0000
Energy_ratio_by_chunks_n um_segments_10_segmenc t_focus_3	0.0000	Minus_DI	0.0001	Plus_DI	0.0000
Partial_autocorrelation_lag_7	0.0000	Ar_coefficient_coeff_0_k_10	0.0001	USD_TWI	0.0000
Spkt_welch_density_coeff_5	0.0000	Augmented_dickey_fuller_attr_ "teststat" _autolag_ "AI C"	0.0001	Last_location_of_minimum	0.0000

Kurtosis	0.0000	Augmented_dickey_fuller_attr_"pvalue"_autolag_"AI C"	0.0001	High	0.0000
Index_mass_quantile_q_0.4	0.0000	Lempel_ziv_complexity_bis_10	0.0001	Cid_ce_normalize_True	0.0000
MFI	0.0000	Last_location_of_minimum	0.0001	Index_mass_quantile_q_0.6	0.0000
Mean_Std_Year	0.0000	CRB_Spot	0.0001	Low	0.0000
Max_Rets_1M	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_8	0.0000	Amih_L	0.0000
Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_0	0.0000	Plus_DI	0.0000	BDI	0.0000
Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_6	0.0000	Ulcer	0.0000	Close	0.0000
Kyle_L	0.0000	Amih_L	0.0000	Fft_aggregated_aggtype_"kurtosis"	0.0000
BDI	0.0000	Cid_ce_normalize_True	0.0000	MACD	0.0000
Permutation_entropy_dim ension_4_tau_1	0.0000	Permutation_entropy_dim ension_4_tau_1	0.0000	Ar_coefficient_coeff_6_k_10	0.0000
Last_location_of_maximum	0.0000	Kyle_L	0.0000	Max_Rets_1M	0.0000
Ulcer	0.0000	USD_TWI	0.0000	Variation_coefficient	0.0000
Index_mass_quantile_q_0.6	0.0000	Index_mass_quantile_q_0.6	0.0000	Partial_autocorrelation_la g_7	0.0000
Minus_DI	0.0000	Index_mass_quantile_q_0.9	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_0	0.0000
Amih_L	0.0000	Cid_ce_normalize_False	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_3	0.0000
Cid_ce_normalize_True	0.0000	MFI	0.0000	Fft_aggregated_aggtype_"c entroid"	0.0000
Cid_ce_normalize_False	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_6	0.0000	Spkt_welch_density_coeff_2	0.0000
USD_TWI	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_0	0.0000	MFI	0.0000
CRB_Spot	-0.0001	Max_Rets_1M	0.0000	Spkt_welch_density_coeff_5	0.0000
Plus_DI	-0.0001	Kurtosis	0.0000	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_6	0.0000
Last_location_of_minimum	-0.0001	Partial_autocorrelation_la g_7	0.0000	Kurtosis	0.0000
Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_8	-0.0001	Variation_coefficient	0.0000	Index_mass_quantile_q_0.4	0.0000
First_location_of_minimum	-0.0002	Spkt_welch_density_coeff_5	0.0000	Count_below_mean	0.0000
Lempel_ziv_complexity_bis_10	-0.0003	Energy_ratio_by_chunks_n um_segments_10_segmenc_t_focus_3	0.0000	Fft_aggregated_aggtype_"variance"	0.0000
Augmented_dickey_fuller_attr_"pvalue"_autolag_"AI C"	-0.0003	Index_mass_quantile_q_0.4	0.0000	Mean_Std_Year	0.0000
Ar_coefficient_coeff_0_k_10	-0.0004	Spkt_welch_density_coeff_2	0.0000	Index_mass_quantile_q_0.8	0.0000
Augmented_dickey_fuller_attr_"teststat"_autolag_"AI C"	-0.0004	Index_mass_quantile_q_0.8	0.0000	Mean_Std_Month	0.0000
BB	-0.0005	Ar_coefficient_coeff_6_k_10	0.0000	Index_mass_quantile_q_0.9	0.0000
CCI	-0.0006	MACD	0.0000	Permutation_entropy_dim ension_4_tau_1	0.0000
ROC_1M	-0.0007	Fft_aggregated_aggtype_"kurtosis"	0.0000	Ar_coefficient_coeff_5_k_10	0.0000

Index_mass_quantile_q_0_7	-0.0007	Fft_aggregated_aggtpe_"c_entroid"	0.0000	Kyle_L	0.0000
Force_Inx	-0.0010	Fft_aggregated_aggtpe_"variance"	0.0000	Ulcer	0.0000
Awesome	-0.0013	Mean_Std_Month	0.0000	CRB_Spot	0.0000
TRIX	-0.0014	Mean_Std_Year	0.0000	Minus_DI	-0.0001
Time_reversal_asymmetry_statistic_lag_3	-0.0029	Count_below_mean	0.0000	CMO	-0.0001
Time_reversal_asymmetry_statistic_lag_2	-0.0037	BDI	0.0000	BB	-0.0001
Time_reversal_asymmetry_statistic_lag_1	-0.0041	Ar_coefficient_coeff_5_k_10	0.0000	ROC_12M	-0.0002
ROC_12M	-0.0043	Close	0.0000	Time_reversal_asymmetry_statistic_lag_1	-0.0005
Mean_change	-0.0064	Low	0.0000	Time_reversal_asymmetry_statistic_lag_2	-0.0006
ROC_6M	-0.0083	High	0.0000	Time_reversal_asymmetry_statistic_lag_3	-0.0007
CMO	-0.0106	Ultimate	-0.0001	Force_Inx	-0.0008
RSI	-0.0112	First_location_of_maximum	-0.0001	ROC_6M	-0.0009
Energy_ratio_by_chunks_n_um_segments_10_segment_focus_9	-0.0237	Last_location_of_maximum	-0.0002	Mean_change	-0.0010

Table 49: Feature importances of the FX Random Forest model.

“Bullish” regime		“Bearish” regime		“Other” regime	
Features	Mean Shap value	Features	Mean Shap value	Features	Mean Shap value
Ar_coefficient_coeff_6_k_10	0.0018	ROC_6M	0.0266	ROC_12M	0.0060
BDI	0.0005	ROC_12M	0.0155	Mean_change	0.0052
Lempel_ziv_complexity_bins_10	0.0002	Mean_change	0.0120	CMO	0.0036
CCI	0.0001	Time_reversal_asymmetry_statistic_lag_3	0.0032	RSI	0.0029
Last_location_of_maximum	0.0001	TRIX	0.0026	Augmented_dickey_fuller_attr_“pvalue”_“autolag”_“AIC”	0.0028
BB	0.0001	10Y bond yield	0.0018	TRIX	0.0017
High	0.0000	Energy_ratio_by_chunks_n_um_segments_10_segment_focus_9	0.0016	ROC_6M	0.0010
Ar_coefficient_coeff_7_k_10	0.0000	Time_reversal_asymmetry_statistic_lag_2	0.0013	BB	0.0005
US_CPI	-0.0001	Augmented_dickey_fuller_attr_“pvalue”_“autolag”_“AIC”	0.0013	Last_location_of_minimum	0.0004
Time_reversal_asymmetry_statistic_lag_1	-0.0001	Time_reversal_asymmetry_statistic_lag_1	0.0003	CCI	0.0004
Ultimate	-0.0003	Last_location_of_minimum	0.0002	Ultimate	0.0004
ROC_1M	-0.0004	ROC_1M	0.0001	ROC_1M	0.0003
Time_reversal_asymmetry_statistic_lag_2	-0.0005	Ar_coefficient_coeff_7_k_10	0.0000	High	0.0003
CMO	-0.0005	US_CPI	0.0000	US_CPI	0.0001
Last_location_of_minimum	-0.0007	Ultimate	-0.0001	Last_location_of_maximum	0.0001
Energy_ratio_by_chunks_n_um_segments_10_segment_focus_9	-0.0009	BDI	-0.0001	Ar_coefficient_coeff_7_k_10	0.0000
RSI	-0.0012	Lempel_ziv_complexity_bins_10	-0.0002	Lempel_ziv_complexity_bins_10	0.0000
10Y bond yield	-0.0016	Last_location_of_maximum	-0.0002	Time_reversal_asymmetry_statistic_lag_1	-0.0002

Time_reversal_asymmetry_statistic_lag_3	-0.0019	High	-0.0003	10Y bond yield	-0.0003
Augmented_dickey_fuller_attr_"pvalue"_"autolag_"AI C"	-0.0041	CCI	-0.0005	BDI	-0.0004
TRIX	-0.0043	BB	-0.0006	Ar_coefficient_coeff_6_k_10	-0.0006
Mean_change	-0.0171	Ar_coefficient_coeff_6_k_10	-0.0012	Energy_ratio_by_chunks_n um_segments_10_segm en_t_focus_9	-0.0007
ROC_12M	-0.0215	RSI	-0.0017	Time_reversal_asymmetry_statistic_lag_2	-0.0009
ROC_6M	-0.0276	CMO	-0.0031	Time_reversal_asymmetry_statistic_lag_3	-0.0013