

# An Online Learning Algorithm for Technical Trading



Nicholas John Murphy

Supervisor: A/Prof. Tim Gebbie

A dissertation submitted in partial fulfilment of the requirement  
for the Master of Science degree in Advanced Analytics,  
School of Statistical Sciences,  
University of Cape Town.

July 13, 2019

## Abstract

We use an adversarial expert based online learning algorithm to learn the optimal parameters required to maximise wealth trading zero-cost portfolio strategies. The learning algorithm is used to determine the relative population dynamics of technical trading strategies that can survive historical back-testing as well as form an overall aggregated portfolio trading strategy from the set of underlying trading strategies implemented on daily and intraday Johannesburg Stock Exchange data. The resulting population time-series are investigated using unsupervised learning for dimensionality reduction and visualisation. A key contribution is that the overall aggregated trading strategies are tested for statistical arbitrage using a novel hypothesis test proposed by

Jarrow *et al.* [31] on both daily sampled and intraday time-scales. The (low frequency) daily sampled strategies fail the arbitrage tests after costs, while the (high frequency) intraday sampled strategies are not falsified as statistical arbitrages after costs. The estimates of trading strategy success, cost of trading and slippage are considered along with an offline benchmark portfolio algorithm for performance comparison. In addition, the algorithms generalisation error is analysed by recovering a probability of back-test overfitting estimate using a nonparametric procedure introduced by Bailey *et al.* [19]. The work aims to explore and better understand the interplay between different technical trading strategies from a data-informed perspective.

**Keywords:** online learning, technical analysis, portfolio selection, back-testing, statistical arbitrage, overfitting, Johannesburg Stock Exchange

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

## Acknowledgements

I would like to express my gratitude to my supervisor A/Prof. Tim Gebbie. His guidance, dedication, deep insight and knowledge of a wide variety of topics have been key attributes in assisting me in completing this journey. There was never a shortage of input and ideas from him, even when various unforeseen issues arose during the course of this research. I am really grateful to have had the opportunity to work with Tim over the past three years, during my honours and master's studies.

I would also like to thank the Statistical Finance Research Group in the Department of Statistical Sciences for various useful discussions relating to the work. In particular, I would like to thank Etienne Pienaar, Lionel Yelibi and Duncan Saffy. I also thank Michael Gant for numerous valuable discussions with regards to the statistical arbitrage test.

Last but not least, I send my most heartfelt thanks to my parents and my sister, who have provided unconditional support and remained patient with me over the past couple of years. I also thank my friends - without them, this journey would not have been complete. It has been a great experience, with both highs and lows, but those close to me have kept me going with their constant moral support and encouragement so, I thank them for that.

## Related Publications and Pre-prints

Much of the content contained in this paper has been submitted in the form of a pre-print to the arXiv.org e-Print archive and is titled “Learning the population dynamics of technical trading strategies” [\[70\]](#).

## Declaration

I hereby declare that this dissertation submitted in partial fulfilment of the requirement for the Master of Science degree in Advanced Analytics is my own work except where specific reference is made to the work of others. The contents of the dissertation have not been submitted in whole or in part for a prior qualification at this university or any other university.

# Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Overview of this Study . . . . .	17
<b>2</b>	<b>Online Portfolio Selection</b>	<b>18</b>
2.1	Online vs. Offline Algorithms . . . . .	18
2.2	The Portfolio Selection Problem . . . . .	19
2.3	Offline Benchmark Strategies . . . . .	20
2.3.1	Buy-and-Hold Strategy . . . . .	20
2.3.2	Best Stock Strategy . . . . .	20
2.3.3	Constant Rebalancing Strategy . . . . .	20
2.3.3.1	BCRP Benchmark Algorithm . . . . .	21
2.4	Universal Portfolio Algorithm . . . . .	21
2.5	Meta-Learning Algorithms . . . . .	22
<b>3</b>	<b>Technical Analysis</b>	<b>22</b>
3.1	Technical Indicators and Trading Rules . . . . .	23
3.1.1	Simple Moving Average . . . . .	23
3.1.1.1	SMA Indicator: $\mathbf{SMA}_t^c(n)$ . . . . .	23
3.1.1.2	Moving Average Crossover Trading Rule . . . . .	23
3.1.2	Exponential Moving Average . . . . .	24
3.1.2.1	Exponential Moving Average Crossover Rule . . . . .	24
3.1.2.2	EMA Indicator: $\mathbf{EMA}_t^c(n)$ . . . . .	24
3.1.3	Highest High . . . . .	24
3.1.4	Lowest Low . . . . .	25
3.1.5	Ichimoku Kinko Hyo . . . . .	25
3.1.5.1	Ichimoku Kinko Hyo Indicators . . . . .	25
3.1.5.2	Ichimoku Kinko Hyo Strategies . . . . .	25
3.1.6	Momentum . . . . .	27
3.1.6.1	Momentum Indicator: $\mathbf{MOM}_t(n)$ . . . . .	27
3.1.7	Momentum Trading Rule . . . . .	27
3.1.8	Acceleration . . . . .	27
3.1.8.1	Acceleration Indicator: $\mathbf{ACC}_t(n)$ . . . . .	27
3.1.8.2	Acceleration Trading Rule . . . . .	27
3.1.9	Moving Average Convergence/Divergence Oscillator . . . . .	27
3.1.9.1	MACD Indicators: $\mathbf{MACD}_t(n_1, n_2)$ . . . . .	28
3.1.9.2	MACD Trading Rule . . . . .	28
3.1.10	Fast Stochastics . . . . .	28
3.1.10.1	Fast Stochastic Indicators . . . . .	29
3.1.10.2	Fast Stochastic Trading Rule . . . . .	29
3.1.11	Slow Stochastics . . . . .	29
3.1.11.1	Slow Stochastic Indicators . . . . .	29
3.1.11.2	Slow Stochastic Trading Rule . . . . .	29
3.1.12	Relative Strength Index . . . . .	30
3.1.12.1	RSI Indicator: $\mathbf{RSI}_t(n)$ . . . . .	30
3.1.12.2	RSI Trading Rule . . . . .	30
3.1.13	Moving Average Relative Strength Index . . . . .	31
3.1.13.1	MARSI Indicator: $\mathbf{MARSI}_t(n_1, n_2)$ . . . . .	31
3.1.13.2	MARSI Trading Rule . . . . .	31
3.1.14	Bollinger Band . . . . .	32

3.1.14.1	Bollinger Band Indicator: $\mathbf{Boll}_t^m(n)$ . . . . .	32
3.1.14.2	Bollinger Trading Rule . . . . .	32
3.1.15	Price Rate-Of-Change . . . . .	32
3.1.15.1	PROC Indicator: $\mathbf{PROC}_t(n)$ . . . . .	32
3.1.15.2	PROC Trading Rule . . . . .	32
3.1.16	Williams %R . . . . .	33
3.1.16.1	Williams %R Indicator: $\mathbf{Will}_t(n)$ . . . . .	33
3.1.16.2	Williams %R Trading Rule . . . . .	33
3.1.17	Parabolic SAR . . . . .	33
3.1.17.1	SAR Indicator: $\mathbf{SAR}(n)$ . . . . .	33
3.1.17.2	SAR Trading Rule . . . . .	34
3.2	Trend Following and Contrarian Mean Reversion Strategies . . . . .	34
3.2.1	Zero-Cost BCRP . . . . .	34
3.2.2	Zero-Cost Anti-BCRP . . . . .	34
3.2.3	Zero-Cost Anti-Correlation . . . . .	35
<b>4</b>	<b>Data and Data Processing</b> . . . . .	<b>35</b>
4.1	Daily Data . . . . .	35
4.2	Intraday-Daily Data . . . . .	36
4.2.1	Time Bar Aggregation . . . . .	36
<b>5</b>	<b>Learning Technical Trading</b> . . . . .	<b>37</b>
5.1	Expert Generating Algorithm . . . . .	38
5.1.1	Technical Trading . . . . .	38
5.1.2	Transforming Signals into Weights . . . . .	39
5.2	Online Learning Algorithm . . . . .	41
5.3	Algorithm Implementation for Intraday-Daily Trading . . . . .	46
5.4	Trading in Volume-time . . . . .	50
5.5	Transaction Costs and Market Frictions . . . . .	50
5.5.1	Volatility Estimation for Transaction Costs . . . . .	52
5.5.1.1	Daily Data Estimation . . . . .	52
5.5.1.2	Intraday Data Estimation . . . . .	52
<b>6</b>	<b>Testing for Statistical Arbitrage</b> . . . . .	<b>53</b>
6.1	Outline of the Statistical Arbitrage Test Procedure . . . . .	56
<b>7</b>	<b>Probability of Back-test Overfitting</b> . . . . .	<b>57</b>
7.1	Back-test Overfitting Framework . . . . .	58
7.2	CSCV Procedure . . . . .	59
<b>8</b>	<b>Results and Analysis</b> . . . . .	<b>61</b>
8.1	Daily Data . . . . .	61
8.1.1	No Transaction Costs . . . . .	61
8.1.2	Transaction Costs . . . . .	69
8.2	Intraday-Daily Data . . . . .	70
8.2.1	No Transaction Costs . . . . .	72
8.2.2	Transaction Costs . . . . .	77
<b>9</b>	<b>Conclusion</b> . . . . .	<b>78</b>
9.1	Future Work . . . . .	81
<b>10</b>	<b>References</b> . . . . .	<b>82</b>

<b>11</b>	<b>Appendices</b>	<b>86</b>
A	Data Related Appendices . . . . .	86
A.1	JSE TOP 40 Sector Constituents: Daily Data . . . . .	86
A.2	JSE TOP 40 Sector Constituents: Intraday Data . . . . .	86
B	Work flow for Learning Class . . . . .	87

## List of Figures

1	Example of the moving average crossover trading rule implemented on 9 months of AGL data . . . . .	24
2	Example of the Ichimoku Kijun Sen Cross strategy implemented on 9 months of AGL data . . . . .	26
3	Example of the MACD trading rule implemented on 9 months of AGL data . . . . .	28
4	Example of the RSI trading rule implemented on 9 months of AGL data . . . . .	31
5	Illustration of how trades arriving at irregular time intervals are sampled over regular intervals (5-minutes). . . . .	37
6	Relationship between the components of the Online Learning Algorithm . . . . .	46
7	Figure 7(a) illustrates the overall cumulative portfolio wealth ( $\mathbf{S}$ ) of the online learning algorithm (OLA - blue) against the benchmark BCRP strategy (orange) and Figure 7(b) illustrates the associated profits and losses ( $\mathbf{PL}$ ) for daily trading without transaction costs. . . . .	62
8	Figure 8(a) illustrates the expert wealth ( $\mathbf{Sh}$ ) for all $\Omega$ experts for daily data with no transaction costs. Figure 8(b) illustrates the mean expert wealth of all experts for each trading strategy ( $\omega$ ) for daily data with no transaction costs. . . . .	63
9	Figure 9(a) and Figure 9(b) show the latent space of a Variational Autoencoder on the time series of expert wealth's, implemented using Keras in Python. In Figure 9(a) experts are coloured by which of the 4 object clusters they trade whereas in Figure 9(b), experts are coloured by their underlying trading strategy $\omega(i)$ . . . . .	66
10	Histogram of the 5000 simulated Min- $t$ statistics resulting from the CM test implemented on the simulated incremental process given in Eq. (58) along with the Min- $t$ statistic (green) for the overall strategy's profit and loss sequence over the 400-day period stretching from the 30 <sup>th</sup> trading day until the 430 <sup>th</sup> trading day without any account for transactions costs. . . . .	67
11	Probability of the overall trading strategy generating a loss after for each of the 30 days from the first trading day through until the 30 <sup>th</sup> trading day. . . . .	68
12	Probability of back-test overfitting for 30 simulations of the online learning algorithm on daily historic data. . . . .	69
13	Profits and losses ( $\mathbf{PL}$ ) for overall strategy for daily trading with transaction costs. . . . .	70

14	Figure 14(a) illustrates the histogram of the 5000 simulated Min- $t$ statistics resulting from the CM model and the incremental process given in Eq. (58) along with the Min- $t$ statistic (green) for the overall strategy's daily trading profit and loss sequence over the 400-day period stretching from the 30 <sup>th</sup> trading day until the 430 <sup>th</sup> trading day with transactions costs incorporated. Also illustrated is the critical value at the 5% significance level (red). Figure 14(b) shows the probability of the overall trading strategy generating a loss for each of the first 400 trading days. . . . .	71
15	Fig. 15(a) shows the overall cumulative portfolio wealth ( <b>S</b> ) for intraday-daily trading with no transaction costs. Fig. 15(b) illustrates the profits and losses ( <b>PL</b> ) for overall strategy for intraday-daily trading with no transaction costs. . . . .	73
16	Fig. 16(a) illustrates the expert wealth ( <b>Sh</b> ) for all $\Omega$ experts for intraday-daily trading with no transaction costs. Fig. 16(b) illustrates the mean expert wealth of all experts for each trading strategy ( $\omega(i)$ ) for intraday-daily trading with no transaction costs. . . . .	74
17	Histogram of the 5000 simulated Min- $t$ statistics resulting from the profit and loss process without transaction costs taken into account along with the Min- $t$ statistic for the overall strategy (green) and the critical value at the 5% significance level (red). The profit and loss process is extracted from the 6 <sup>th</sup> time bar on the second day, which corresponds to the period when active trading begins, until the 400 <sup>th</sup> time bar hence. This corresponds to roughly four and a half days worth of trading profits. . . . .	75
18	Probability of the overall trading strategy generating a loss after $n$ periods for each $n = 1, \dots, 30$ of the intraday-daily profit and loss process ( <b>PL</b> ) without transaction costs. The profits and losses are taken starting from the 5 <sup>th</sup> time bar of the second day when active trading commences as the first day is required as a look-back for the daily trading component. . . . .	76
19	Probability of back-test overfitting for 20 simulations of the online learning algorithm on intraday-daily historic data. . . . .	77
20	Figure 20(a) and Figure 20(b) illustrate the overall cumulative portfolio wealth ( <b>S</b> ) and profits and losses ( <b>PL</b> ) respectively for intraday-daily trading with transaction costs. . . . .	79
21	Figure 21(a) shows the histogram of the 5000 simulated Min- $t$ statistics resulting from the CM model and the incremental process given in Eq. (58). Also illustrated is the Min- $t$ statistic (green) for the first 400 trading periods from commencement of active trading (5 <sup>th</sup> time bar of the second day) for intraday-daily profit and losses with transaction costs incorporated along with the critical value at the 5% significance level (red). Figure 21(b) illustrates the probability of the overall trading strategy generating a loss after $n$ periods for each $n = 1, \dots, 30$ of the intraday-daily profit and loss process ( <b>PL</b> ) with transaction costs incorporated commencing from the first period of active trading (5 <sup>th</sup> time bar of the second day). . . . .	80
22	State flow diagram for the MATLAB learning class . . . . .	87

## List of Tables

1	Variable definitions . . . . .	12
2	Ichimoku Kijun Sen Cross strategy . . . . .	25
3	Momentum trading rule . . . . .	27
4	Acceleration trading rule . . . . .	27
5	MACD trading rule . . . . .	28
6	Fast stochastic trading rule . . . . .	29
7	Slow stochastic trading rule . . . . .	30
8	Relative strength index (RSI) trading rule . . . . .	30
9	MARSI trading rule . . . . .	31
10	Bollinger trading rule . . . . .	32
11	Price rate of change (PROC) trading rule . . . . .	32
12	Williams %R trading rule . . . . .	33
13	SAR trading rule . . . . .	34
14	Algorithm for the expert generating function. . . . .	42
15	Algorithm for the controls function which transforms trading signals into portfolio controls. In the above algorithm, <b>output_s</b> refers to <i>output signals</i> (combined signals) as discussed in Section 5.1.2 . . . . .	43
16	Algorithm which transforms output signals into portfolio controls using the volatility loading method. The algorithm follows from Algorithm 2 and is part of the control function called by Algorithm 1. Note that $\text{output\_s} \leq 0$ (or $\geq$ ) implies that $\text{output\_s}(i) \leq 0$ (or $\geq$ ) for each $i$ . . . . .	44
17	Algorithm for Online Learning. . . . .	47
18	Group summary statistics of the overall rankings of experts grouped by their underlying strategy ( $\omega(i)$ where $i = 1, \dots, 17$ ) for the daily trading. In brackets next to mean are the mean overall ranking of experts utilising each strategy. . . . .	64
19	Group summary statistics of the expert's profits and losses per period grouped by their underlying strategy ( $\omega(i)$ where $i = 1, \dots, 17$ ). . . . .	65
20	Group summary statistics of the overall rankings of experts grouped by their underlying strategy ( $\omega(i)$ where $i = 1, \dots, 17$ ) for intraday-daily trading. In brackets are the mean overall ranking of experts utilising each strategy. . . . .	72

## List of Variables

Variable	Definition
$\omega$	parameter of underlying strategy of a given expert which corresponds to one of the technical trading strategies or trend-following strategies. $\omega(i)$ is then the $i^{\text{th}}$ strategy from the set of all strategies
$W_1$	number of trading strategies which utilise one parameter
$W_2$	number of trading strategies utilising two parameters
$W$	total number of trading strategies: $W =  \omega  = W_1 + W_2$
$n_1$	vector of short-term look-back parameters
$n_2$	vector of long-term look-back parameters
$L$	number of short-term look-back parameters

$K$	number of short-term look-back parameters
$\mathbf{c}$	object clusters parameter where $\mathbf{c}(i)$ is the $i^{th}$ object cluster
$C$	number of object clusters
$\Omega$	total number of experts (this is different to the bold omega $\mathbf{\Omega}$ used in the probability of back-test overfitting framework)
ADV	average daily volume
$\delta_{liq}$	number of periods to look-back to compute the ADV and choose the $m$ most liquid stocks
$m$	number of stocks to be passed into the expert generating algorithm ranked by their liquidity
$\mathbf{h}_t^n$	$n^{th}$ expert's strategy ( $m + 1$ portfolio controls) for period $t$
$\mathbf{H}_t$	expert control matrix by made up of all $n$ experts' strategies at time $t$ for all $m$ stocks <i>i.e.</i> $\mathbf{H}_t = [\mathbf{h}_t^1, \dots, \mathbf{h}_t^n]$
$\mathbf{Sh}_t$	wealth of all $n$ experts at time $t$
$Sh_t^n$	$n^{th}$ expert's wealth at time $t$
$\mathbf{b}_t$	final overall aggregate portfolio to be used in the following period $t + 1$ which we denote
$\mathbf{S}$	vector of the overall aggregate portfolio compounded wealth over time
$S_t$	overall aggregate portfolio compounded wealth at time $t$
$\mathbf{PL}$	vector of the overall aggregate portfolio cumulative profits and losses
$\mathbf{PL}_t$	the overall aggregate portfolio cumulative profits and losses at time $t$
$nb$ ( $ns$ )	number of buy (sell) signals from the set of output signals
$\sigma_+$ ( $\sigma_-$ )	vector of standard deviations of stocks to be bought (sold)
$\mathbf{w}$	$m + 1$ vector of weights allocated to $m$ stocks and the risk-free asset
$w_{rf}$	weight allocated to the risk-free asset
$t_{min}$	time period at which trading commences (either daily/intraday)
$T$	terminal time of trading (generally daily but can be either daily/intraday)
$\mathbf{X}_t$	$m \times 5$ matrix of stock's OHLCV values at each time period $t$ (either daily/intraday)
$\mathbf{X}_d$	daily OHLCV data
$\mathbf{X}_I$	intraday OHLCV data
$\mathbf{x}_t$	vector of $m + 1$ price relatives at time period $t$
$\mathbf{P}_t^c$	vector of closing prices for all $m$ stocks at time period $t$
$P_{m,t}^c$	closing price of stock $m$ at time period $t$
$t_I$	$t_I^{th}$ time bar of a given day
$T_I$	final (terminal) time bar of a given day (4:30pm)
$\mathbf{H}_{t,T_I}^F$	fused intraday-daily expert control matrix
$\mathbf{h}_{n,t,T_I}^F$	$n^{th}$ expert's controls for all $m$ assets at time bar $t_I$ on the $t^{th}$ day
$\mathbf{Sh}_{t,t_I}^F$	fused intraday-daily expert wealth vector for all $n$ experts for the $t_I^{th}$ time bar on the $t^{th}$ day
$Sh_{n,t,T_I+1}^F$	$n^{th}$ expert's wealth at time bar $t_I$ on the $t^{th}$ day
$\sigma$	volatility of the returns of a stock

---

$\nu(t)$	cumulative discounted trading profits/losses at time $t$
$\Delta\nu_t$	cumulative discounted trading profit/loss increment at time $t$ <i>i.e.</i> $\Delta\nu_t = \nu(t) - \nu(t-1) - 1$
Min- $t$	statistic used to make inferences regarding the no statistical arbitrage null hypothesis
$T_{\text{BL}}$	back-test length for each of the $N$ separate trial simulations of the algorithm
$N$	number of back-test trials performed on independent subsets of historic data ( $N = \lfloor T/T_{\text{BL}} \rfloor$ trial simulations)
$N!$	number of permutations of the set $(1, \dots, N)$ which ranks the $N$ back-test trials
$\mathbf{R}(\bar{\mathbf{R}})$	IS (OOS) performance of the $N$ back-tests for a fixed performance measure (Sharpe Ratio)
$\mathbf{R}^C(\bar{\mathbf{R}}^C)$	performances for a pair of IS (OOS) sample subsets of $\mathbf{R}(\bar{\mathbf{R}})$
$\Omega$	ranking space of $N!$ (This bold and capitalized omega is different from the one used to denote the total number of experts as referred to previously in the study which is not bold ( $\Omega$ ))
$\mathbf{r}(\bar{\mathbf{r}})$	rankings of trials in the vector $\mathbf{R}(\bar{\mathbf{R}})$ with $n^{\text{th}}$ element $r_n(\bar{r}_n)$
$\Omega_n^*$	subspace of vectors of back-test trials' performance rankings $\mathbf{f}$ ( $\{\mathbf{f} \in \Omega   f_n = N\}$ of $\Omega$ where $\mathbf{f} = (f_1, \dots, f_n)$ ), such that the $n^{\text{th}}$ back-test trial has the highest ranking <i>i.e.</i> $f_n = N$

---

Table 1: Variable definitions

# 1 Introduction

Selecting plausible trading strategies, and allocating wealth among these strategies, in order to maximise wealth over multiple decision periods can be a difficult task. An approach to combining strategy selection with wealth maximisation is to use online or sequential machine learning algorithms [1]. Online portfolio selection algorithms attempt to automate a sequence of trading decisions among a set of stocks with the goal of maximizing return in the long run.<sup>1</sup> Such algorithms typically use historical market data to determine, at the beginning of a trading period, a way to distribute their current wealth among a set of stocks. These types of algorithms can use many more features than merely price, so called “side-information” [2], but the principle remains that same. The attraction of this approach is that the investor does not need to have any knowledge about the underlying distributions that could be generating the stock prices (or even if they exist). The investor is left to “learn” the optimal portfolio to achieve maximum wealth using past data directly [1].

Cover [3] introduced a “follow-the-winner” online investment algorithm<sup>2</sup> called the Universal Portfolio (UP) algorithm<sup>3</sup>. The basic idea of the UP algorithm is to allocate capital to a set of experts characterised by different portfolios or trading strategies; and to then let them run while at each iterative step to shift capital from losers to winners to find a final aggregate wealth.

Here our “experts” will be similarly characterised by a portfolio (or trading strategy) where a particular agent makes decisions independently of all other experts. The UP algorithm holds parametrized constant rebalanced portfolio (CRP) strategies as it’s underlying experts. We will have a more generalised approach to generating experts. The algorithm provides a method to effectively distribute wealth among all the CRP experts such that the average log-performance of the strategy approaches the best constant rebalancing portfolio (BCRP) which is the hindsight strategy chosen which gives the maximum return of all such strategies in the long run. The key innovation that Cover [3] provided was a mathematical proof for this claim based on an arbitrary sequences of ergodic and stationary stock return vectors.

It is important to realise that if there exists some log-optimal portfolio such that no other investment strategy has a greater asymptotic average growth then to achieve this one must have full knowledge of the underlying distribution and of the generating process to achieve such optimality [2, 3, 4, 5]. Such knowledge is unlikely in the context of financial markets. However, strategies which achieve an average growth rate which asymptotically approximates that of the log-optimal strategy is possible given that the underlying asset return process is sufficiently close to being stationary and ergodic. Such a strategy is called *universally consistent*. Gyorfı *et al.* [5] proposed a universally consistent portfolio strategy and provided empirical evidence of a strategy based on nearest-neighbour based experts which reflects such asymptotic log-optimality.

The idea is to match current price dynamics with similar historical dynamics (pattern matching) using a nearest-neighbour search algorithm to select parameters for

---

<sup>1</sup>Here, the long run will depend on the frequency at which trading occurs. This could imply anything from a few days to a few weeks for high frequency trading algorithms and a few months (years) for daily (weekly) trading algorithms

<sup>2</sup>*follow-the-winner* algorithms give greater weightings to better performing experts or stocks

<sup>3</sup>The algorithm was later refined by Cover and Ordentlich [2] (see Section 5.2)

experts. The algorithm was extended by Loonat and Gebbie [6] in order to implement a zero-cost (long/short and self-financing) portfolio selection algorithm via a quadratic approximation derived from the mutual fund separation theorems to allow the algorithm to be more effectively considered in the context statistical arbitrage trading. The algorithm was also re-cast to replicate near-real-time applications using look-up libraries learnt offline. However, there is a computational cost associated with coupling creation of offline pattern libraries - the algorithms are not truly online<sup>4</sup>. A key objective in the implementation online learning in this work is that the underlying experts here are online too; they can be sequentially computed on a moving finite data-window using parameters from the previous time step - rather than having the need to search data-histories or make comparisons with a library of patterns learnt offline.

Here we ignore the pattern matching step in the aforementioned algorithm and rather propose our own expert generating algorithm using tools from technical analysis. Concretely, we replace the pattern-matching expert generating algorithm with a selection of technical trading strategies. *Technical trading* refers to the practice of using trading strategies (rules) derived technical analysis indicators which use mathematical formulas based on prices, volume traded or a combination of both to generate trading signals [7, 8]. They claim to be able to exploit statistically measurable short-term market opportunities in stock prices and volume by studying recurring patterns in historical market data [9, 10, 11]. An abundance of indicators has been developed over the years with some proving to be more successful than others. Indicators perform differently under different market conditions which is why traders will often use multiple indicators to confirm the signal that one indicator gives on a stock with another indicator's signal. Thus, in practice and various studies in the literature, many trading rules generated from indicators are typically back tested on a sufficient amount of historical data to find the rules that perform the best. This is typically known as data mining. One must be careful when using data mining to test the performance of many rules since the chance of a "lucky"<sup>5</sup> performance of a given rule from the set will increase leading to biased results [12]. This is often what motivates the need to implement rigorous statistical analysis and back-tests to determine which rules perform consistently well.

Traditionally, technical analysis has been a visual activity, whereby traders study the patterns and trends in charts, based on price or volume data, and use these diagnostic tools in conjunction with a variety of qualitative market features and news flow to make trading decisions. Many studies have criticised the lack of a solid mathematical foundation for many of the proposed technical analysis indicators [12, 13, 14]. There has also been an abundance of academic literature, utilising technical analysis for the purpose of trading and several studies have attempted to develop indicators and test them in a more mathematically, statistically and numerically sound manner [12, 15, 16]. However, much of this work needs to be viewed with some suspicion - it is extremely unlikely that this or that particular strategy or approach was not the result of some sort of back-test overfitting [17, 18, 19]. Many studies attempt to predict the future movements of prices using technical analysis with mixed success [9, 11, 20, 21].

This work does not address the question: Which, if any, technical analysis based

---

<sup>4</sup>See Section 2.1 for an explanation on online vs offline algorithms

<sup>5</sup>Luck, in the sense that it may lead to a favourable but accidental correspondence between trends in the historical market data and the rule's signals

methods reveal useful information for trading purposes? Rather we aim to bag a collection of technical experts and allow them to compete in an adversarial manner, using the online learning algorithm, to then consider whether the resulting aggregate strategy passes a reasonable test for statistical arbitrage, and has a relatively low probability of being the result of back-test overfitting *i.e.* it could generalise well out-of-sample.

Here we will be concerned with the idea of understanding whether the collective population of technical experts can through time lead to dynamics that can be considered a statistical arbitrage [22] with a reasonably low probability of back-test overfitting [19]. More specifically, can we generate wealth (before costs) using the online aggregation of technical strategies? Then, what broad groups of strategies will emerge as being successful (here in the sense of positive trading profits with declining variance in losses)? When measuring success, of utmost importance is the consideration of the transaction costs inherent in trading. One of the earliest studies which looked at profitability of filter rules<sup>6</sup> revealed that such trading rules were unprofitable after transaction costs were taken into account [32]. Any reasonable book on trading/investing will certainly contain a section on transaction costs, and if not, will at least have short discussions on the impact of costs on back-test performance of such algorithms [7, 12, 15, 33, 34]. Costs are always a plausible explanation for any apparently profitable trading strategy (see [6]), and after costs, there exists a high-likelihood that there was a healthy amount of data overfitting; given that we only have single price paths from history, and have little or no knowledge about the probability of the particular path that has been measured.

Another important consideration for traders is whether a market is efficient, and hence, whether they can consistently profit from trading in the market. Here efficiency refers to the expeditiousness of market prices to incorporate new information at any time. Thus, in an efficient market, all known and relevant information is presumed to be reflected in prices almost instantaneously [12]. This is the idea behind the popular hypothesis that has been researched and debated for many years, and is known as the Efficient Market Hypothesis (EMH), developed in the groundbreaking work of Fama [23]. Historically, financial theory supported the view that markets are in fact efficient resulting in market price moments that follow a random walk or Brownian motion. This implies that past price movements cannot be used to predict future movements and hence, an efficient market is trend-less and unpredictable. In line with this idea is the statement that “an average investor- whether an individual, pension fund, or a mutual fund- cannot hope to consistently beat the market, and the vast resources that such investors dedicate to analysing, picking and trading securities are wasted” [24]. This isn’t to say that an investment (trading) strategy, such as those generated from technical indicators, cannot generate a positive return, but when the strategy is risk-adjusted, then it will not consistently provide better returns than the market benchmark [12]. In order to beat the market benchmark consistently, an investor will have to take on much higher levels of risk, possibly leading to significant losses. Today, it is generally accepted that markets are in fact inefficient.

A similar train of thought to that of the EMH was introduced by Lo [25], which argues that the efficiency of markets is determined by the dynamics of evolution, and

---

<sup>6</sup>Filter rules are a technical trading rule whereby a trader is signalled to take action by buying or selling a stock when it’s price rises or falls by a certain percentage (often in relation to a previous high or low)

introduces a new paradigm called the Adaptive Market Hypothesis (AMH). The AMH had a similar direction to the original approaches suggested by Farmer and Lo [26] and Farmer [27] in applying evolutionary principles to financial markets. Farmer and Lo [28] describe the universe of computer algorithms as a complex ecology of highly specialized, highly diverse, and strongly interacting agents. This in turn leads to the co-evolution of human trading, computer trading, markets and regulators. They argue that the role computers play in markets can only be understood from an ecological and evolutionary perspective, in a sense that these computers are designed by considering historical incidences in markets, but also specific details such as the design of markets, regulations and patterns in trading. As markets conditions constantly change, new computer systems and trading strategies need to be designed as previous systems and strategies often become unprofitable and/or unsuccessful. This is why the adoption of machine learning in automated trading systems has recently exploded, as such systems are often able to adapt and learn which strategies are becoming more (or less) profitable and hence allocate greater (or smaller) capital amounts for such strategies. This has led to the belief that, with less human involvement in active decision making in trading and investing, that markets are bound to become more efficient, however this is highly debatable.

Rather than considering various debates relating the technicalities of market efficiency, we restrict ourselves to market efficiency in the sense used by Fischer Black [29, 30]. This is the situation where some of the short-term information is in fact noise, and that this type of noise is a fundamental property of real markets. Although market efficiency may plausibly hold over the longer term, in the short-term there may be departures that are amenable to tests for statistical arbitrage [31], departures that create incentives to trade, and more importantly departures that may not be easily traded out of the market due to various asymmetries in costs and market access. The proposed trading strategies in this work are tested in this sense.

In order to analyse whether the overall back-tested strategy depicts a candidate statistical arbitrage, we implement a test first proposed by Hogan *et al.* [22] and then further refined by Jarrow *et al.* [31]. Hogan *et al.* provide a plausible technical definition of statistical arbitrage based on a vanishing probability of loss and variance in the trading profits, and then use this to propose a test for statistical arbitrage using a Bonferroni test [22]. This methodology was further extended and generalized by Jarrow *et al.* to account for the asymmetry between desirable positive deviations (profits) and undesirable negative deviations (losses), by including a semi-variance hypothesis instead of the originally constructed variance hypothesis, which does not condition on negative incremental deviations [31]. The so-called Min- $t$  statistic is computed and used in conjunction with a Monte Carlo procedure to make inferences regarding a carefully defined “no statistical arbitrage” null hypothesis.

This is analogous to evaluating market efficiency in the sense of the Noisy efficient market hypothesis [29] whereby a failure to reject the no statistical arbitrage null hypothesis will result in concluding that the market is in fact sufficiently efficient and no persistent anomalies can be consistently exploited by trading strategies over the long term. Traders will always be inclined to employ strategies which depict a statistical arbitrage and especially strategies which have a probability of loss that declines to zero quickly as such traders will often have limited capital and short horizons over which they must provide satisfactory returns (profits) [31].

## 1.1 Overview of this Study

We make the effort here to be very clear that we do not attempt to identify profitable (technical) trading strategies, but rather we will generate a large population of strategies (experts) constructed from various technical trading rules and combinations of the associated parameters of these rules in the attempt to learn the aggregate population dynamics of the set of experts. Expert's will generate trading signals (buy, sell or hold) for each stock held in their portfolio based on the underlying parameters and the necessary historic data implied by the parameters. Once trading signals for the current time period  $t$  have been generated by a given expert, a methodology to transform the signals into a set of portfolio weights (controls) is required.

We introduce a transformation method that computes controls proportional to the relative volatilities of the stocks for which non-zero trading signals were generated, and then normalise the resulting values such that the self-financing and leverage constraints required by the algorithm are satisfied. The resulting controls are then utilised to compute the corresponding expert wealth's. The experts who accumulate the greatest wealth during a trading period, will receive more wealth in the following trading period, and thus contribute more to the final aggregated portfolio. This can be best thought of as some sort of "fund-of-funds" over the underlying collection of trading strategies. This is a meta-expert<sup>7</sup> that aggregates experts that represent all the individual technical trading rules. The overall meta-expert strategy performance is achieved by the online learning algorithm. Equity curves for the individual expert's portfolios (the accumulated trading profit through time) along with performance curves for the overall strategy's wealth and the associated profits and losses are provided.

We perform a back-test of the algorithm on two different data sets over two separate time periods; one using daily data over a six-year period from 1 January 2010 to 29 April 2016 and the other using a mixture of intraday and daily data over a two and a half month period from 2 January 2018 to 21 March 2018. A selection of the fifteen most liquid stocks which constitute the Johannesburg Stock Exchange (JSE) Top 40 shares is utilised for the two separate implementations.<sup>8</sup>

The overall strategy performance is compared to the BCRP strategy to form a benchmark comparison to evaluate the success of our strategy. The overall strategy is then tested for statistical arbitrage to find that in both a daily and intraday-daily data implementation, the strategy depicts a statistical arbitrage. It must be cautioned that this is prior to accounting for costs which can be expected to shift the bias. The key point here (as in [6]) is that it does seem that plausible statistical arbitrages are detected on the meso-scale and in the short-term, however, for reasonable costing, it may well be that these statistical arbitrages exist because they cannot be profitable traded out of the system.

In addition, transaction cost analysis, the overall strategy's probability of loss is computed to get an idea of the convergence of such losses to zero; most traders will be concerned about short-run losses. Finally, we analyse the generalisation error of the overall strategy to get a sense of whether or not the strategy conveys back-test overfitting by estimating the probability of back test overfitting inherent in multiple

---

<sup>7</sup>See Section 2.5 for a discussion on meta-learning algorithms

<sup>8</sup>More details on the data sets can be found in Section 4

simulations of the algorithm on subsets of historic data.

Throughout the study, we use the terms investing and trading interchangeably, however, we should distinguish the differences between the two. Investors buy stocks with the intention of gaining wealth over the long term by holding a stock which they believe will grow over the period. Investors typically use fundamental analysis<sup>9</sup> to identify long-term potential in a company (stock). Traders buy and sell stocks with the intention of making quick short-term profits on price differences of stocks.

The rest of the paper will proceed as follows: Section 2 discusses the differences between online and offline algorithms and introduces online portfolio selection problems, terminology, and various portfolio selection benchmark algorithms. Section 3 describes technical analysis and provides in-depth descriptions of all trading rules implemented in the project. Section 4 describes the data utilised in the study. Section 5 explains the construction of the algorithm including details of how the experts are generated, how their corresponding trading signals are transformed into portfolio controls and a step-by-step break-down of the learning algorithm. In Section 6, we introduce the concept of a statistical arbitrage, including the methodology for implementing a statistical arbitrage test and the probability of loss for a trading strategy. All experiment results of implementations of the algorithm and extensive analysis is presented in Section 8. Section 9 states all final conclusions from the experiments and possible future work.

## 2 Online Portfolio Selection

With regards to a multi-period *online portfolio selection* (OPS) algorithm, which is what our trading algorithm is based on, the goal is to sequentially rebalance the portfolios of each expert by allocating capital among the set of assets held by each expert with the goal of maximizing the investor’s terminal wealth irrespective of risk [2, 3, 6, 36, 37]. Here, the investor basically represents a weighted average portfolio of all the experts where each expert has its own strategy with which it uses to allocate it’s capital among the set of assets. The goal is to find the optimal portfolio by weighting each expert’s portfolio based on their performance. In achieving the growth optimal portfolio, no underlying statistical assumptions are made as the portfolio wealth is solely dependent on the data [2, 3]. The details of OPS algorithms will be discussed in more depth in the following subsections.

### 2.1 Online vs. Offline Algorithms

Traditionally, the design and analysis (back-tests) of many trading algorithms has been conducted on static data sets where the algorithm is executed on an entire batch of data as an input to the algorithm. However, in practice, often the input data is only partially available since some relevant input data arrives in a sequential manner as time moves forward and markets and traders react to evolving market data and conditions [35]. The process whereby the algorithm learns and updates parameters sequentially as data arrives is also known as *online learning* which is in contrast to

---

<sup>9</sup>Fundamental analysis attempts to measure the intrinsic value of a stock by using economic factors (fundamentals) of the company included in financial statements such as Earnings Per Share, Return On Equity, Revenue and price to earnings ratio to name a few [42, 11]

*offline (batch) learning*, whereby the algorithm updates parameters given that an entire batch of static data is available. Contrary to offline learning, where a decision is completely irrelevant to previous decisions, during online learning one decision is dependent to prior decisions made by the algorithm [36]. Online learning becomes more significant when trading is executed at higher frequencies as the algorithm needs to react faster to market events being streamed into the algorithm at high speeds and be computationally efficient enough to make decisions at high rates.

## 2.2 The Portfolio Selection Problem

The basic portfolio selection problem involves an investor who invests his capital in a market with  $m$  stocks over  $T$  trading periods. Suppose that at the  $t^{\text{th}}$  period ( $t = 1, \dots, T$ ), the stocks have closing prices  $\mathbf{P}_t^c = (p_1, \dots, p_m)$  where  $p_i$  is the price of the  $i^{\text{th}}$  stock at time period  $t$ . Let the price changes of the stocks be represented by price relatives<sup>10</sup> which are just ratios of the prices of each stock  $i$  at time  $t$  to the prices at time  $t - 1$ , that is,  $x_{i,t} = \frac{p_{i,t}}{p_{i,t-1}}$  where the price relatives for the  $t^{\text{th}}$  period are given by  $\mathbf{x}_t = (x_{1,t}, \dots, x_{m,t})$  and the sequence of relatives over the entire investment period are given by  $\mathbf{X} = (x_1, \dots, x_T)$ . Resultantly, an investment in stock  $i$  over period  $t$  will increase by a factor of  $x_{i,t}$  [36]. Denote a *portfolio* by  $\mathbf{b}_t = (b_{1,t}, \dots, b_{m,t})$ , where  $b_{i,t}$  represents the amount of wealth allocated to stock  $i$  at the beginning of period  $t$ . The *portfolio strategy* over the  $T$  periods is thus the sequence of portfolios

$$\mathbf{B} = \mathbf{b}_1, \dots, \mathbf{b}_T \quad (1)$$

where  $\mathbf{b}_1 = (1/m, \dots, 1/m)$ . Then, at time  $t$ , the wealth of the investment increases by a factor of

$$S_t = \sum_{i=1}^m b_{i,t} x_{i,t} \quad (2)$$

$$= \mathbf{b}_t^\top \cdot \mathbf{x}_t \quad (3)$$

Suppose we initially invest an amount  $S_0$  and reinvest the wealth after each period subsequent to accounting for profits and losses. In such a case, the portfolio's wealth grows multiplicatively over time [36]. Thus, the total cumulative wealth at the end of  $T$  periods adopting a portfolio strategy  $\mathbf{B}$  with realised price relatives  $\mathbf{X}$  will be

$$S_T(\mathbf{B}, \mathbf{X}) = S_0 \prod_{t=1}^T \mathbf{b}_t^\top \cdot \mathbf{x}_t \quad (4)$$

For future reference to Eq. (4), we will drop the argument for the price relative sequence  $\mathbf{X}$  so that  $S_T(\mathbf{B}) = S_T(\mathbf{B}, \mathbf{X})$ . The aim is for the investor is to sequentially update (rebalance) the portfolio ( $\mathbf{b}_t$ ) at each time  $t$  as he realises the price relative for the previous period ( $\mathbf{x}_{t-1}$ ) in order to achieve some target. The portfolio is rebalanced by buying and selling the stocks based on whether their prices have dropped or risen respectively. Once the  $t^{\text{th}}$  trading period is complete, the price relative  $\mathbf{x}_t$  is realised. Eq. (4) only holds in the long-only portfolio case, however for our purposes, we have

---

<sup>10</sup>Price relatives (simple gross return) are used in most literature on portfolio selection problems but we could replace these with a simple net return given by  $x_{i,t} = \frac{p_{i,t} - p_{i,t-1}}{p_{i,t-1}}$  [36].

to account for the possibility that a short position is taken in a stock. The adjusted equation for the cumulative wealth of a long/short portfolio is as follows [6]

$$S_T(\mathbf{B}, \mathbf{X}) = S_0 \prod_{t=1}^T [\mathbf{b}_t^\top \cdot (\mathbf{x}_t - 1) + 1] \quad (5)$$

## 2.3 Offline Benchmark Strategies

Before discussing the various OPS algorithms from the literature which form the basis of our algorithm, we need to introduce some of the benchmark principles for OPS which we can use to measure the performance of our algorithms against. We will discuss the three most common benchmark strategies and all three of these are offline<sup>11</sup> benchmarks.

### 2.3.1 Buy-and-Hold Strategy

The first benchmark is the *buy-and-hold* (BH) portfolio strategy is the simplest and one of the most popular benchmarks. Here, an investor will buy an initial portfolio of stocks,  $b_1$ , at the beginning of the 1<sup>st</sup> period and hold it until the end of the investment horizon without adjusting it [37, 39]. A specific type of BH strategy is the *uniform buy-and-hold* (UBH) strategy and corresponds to the case where  $b_1 = (1/m, \dots, 1/m)$  *i.e.* an equal holding of each stock in the portfolio.

### 2.3.2 Best Stock Strategy

The *best stock strategy* is just a special case of the BH strategy and is simply the optimal in hindsight BH strategy [38] whereby the investor will invest all his wealth in the best stock (in hindsight).

### 2.3.3 Constant Rebalancing Strategy

The *constant rebalancing portfolio* (CRP) is a portfolio strategy which rebalances the portfolio to have a fixed proportion in every stock at the beginning of each trading period  $t$  ( $\mathbf{b}_t = \mathbf{b}_{t+1}$ ) [39]. Borodin *et al.* [38] show that the optimal in hindsight CRP known as the *best constant rebalancing portfolio* (BCRP) will always performs at least as well as the best stock strategy and can often significantly outperform the best stock by taking advantage of market fluctuations. Furthermore, Cover [3] shows that this portfolio has the greatest exponential growth in hindsight among all possible BH portfolio allocations. CRP strategies form the basis for much of the theory from which our algorithm is developed from. Given that  $\mathbf{b} = \mathbf{B} = \mathbf{b}_1 = \dots = \mathbf{b}_T$  and following from Eq. (4), the cumulative wealth of a long-only CRP strategy with targeted portfolio  $\mathbf{b}$  is

$$S_T(\mathbf{b}) = S_0 \prod_{t=1}^T \mathbf{b}^\top \cdot \mathbf{x}_t \quad (6)$$

and the corresponding cumulative wealth where short selling is permitted

$$S_T(\mathbf{b}) = S_0 \prod_{t=1}^T [\mathbf{b}^\top \cdot (\mathbf{x}_t - 1) + 1] \quad (7)$$

---

<sup>11</sup>See Section 2.1 for the discussion of online vs offline algorithms

### 2.3.3.1 BCRP Benchmark Algorithm

To get an idea of how well our online algorithm performs, we compare its performance to that of the offline BCRP. There are two possible methods to finding the controls for the BCRP strategy: 1) an analytical method that requires the use of Kuhn-Tucker methods to solve a Lagrangian function involving the expected utility of wealth with respect to a constant relative risk aversion [40] 2) brute force Monte Carlo method [41]. To find the portfolio controls of such a strategy, we perform a brute force Monte Carlo approach to generate 5000 random CRP strategies on the entire history of price relatives and choose the BCRP strategy to be the one that returns the maximal terminal portfolio wealth. As a note here, the CRP strategies we consider for the Monte Carlo simulation are long-only portfolios. Section 8 illustrates the performance of this method against that of our proposed learning method which will be introduced and discussed in detail in Section 5.

## 2.4 Universal Portfolio Algorithm

One of the most popular OPS algorithms is the *Universal Portfolio* (UP) algorithm introduced by Cover [3]. Cover introduces a concept known as *universality* in order to classify a specific type of OPS algorithm. The basic idea of these algorithms is to track the BCRP of any arbitrary sequence of price relatives (returns).

The algorithm was later refined by Cover and Ordentlich [2] and was called the  *$\mu$ -Weighted Universal Portfolio*. Here,  $\mu$  is a given distribution on the simplex  $\mathcal{B}_m$  which represents the space of all valid portfolio strategies of dimension  $m$ . Validity requires that  $\mathcal{B}_m = \{b \in \mathbb{R}^m | b_i \geq 0 \text{ and } \sum_{i=1}^m b_i = 1\}$ . Consider  $\Omega$ <sup>12</sup> experts whereby each expert  $\omega$  invests in  $m$  stocks utilising their own CRP strategy. Thus, each expert will have his own unique fixed portfolio allocation, say  $\mathbf{b}^\omega \in \mathcal{B}_m$ . Suppose that a proportion of wealth  $d\mu(\mathbf{b}^\omega)$  is invested into each expert. Then, following from Eq. (6), the  $t$ -period wealth of the  $\omega$ <sup>th</sup> expert be given by  $S_t(\mathbf{b}^\omega)d\mu(\mathbf{b}^\omega)$  where

$$S_t(\mathbf{b}^\omega) = \prod_{t=1}^T \mathbf{b}^\omega \cdot \mathbf{x}_t \quad (8)$$

and  $S_0(\mathbf{b}^\omega) = 1$ .

In the case that the simplex is continuous, the update rule for the universal portfolio at the beginning of the  $(t+1)$ <sup>th</sup> time period is given by [2]

$$\mathbf{b}_{t+1}^{UP} = \frac{\int_{\mathcal{B}_m} \mathbf{b} S_t(\mathbf{b}) d\mu(\mathbf{b})}{\int_{\mathcal{B}_m} S_t(\mathbf{b}) d\mu(\mathbf{b})} \quad (9)$$

where the rule is initialised with  $b_1 = (\frac{1}{m}, \dots, \frac{1}{m})$ . However, if the simplex is discrete (which is what we assume in this study<sup>13</sup>), the universal portfolio for  $\Omega$  experts is given by

$$\mathbf{b}_{t+1}^{UP} = \frac{\sum_{\omega=1}^{\Omega} \mathbf{b}^\omega S_t(\mathbf{b}^\omega)}{\sum_{\omega=1}^{\Omega} S_t(\mathbf{b}^\omega)} \quad (10)$$

<sup>12</sup> $\omega$  can either be finite or infinite. If  $\mathcal{B}_m$  is a continuous simplex then there are infinite experts ( $\omega = \infty$ ) and if the simplex is discrete then there are a finite number of experts

<sup>13</sup>Finite experts exist *i.e.* finite trading strategies

Eq. (10) will form the basis of the updating rule for our learning algorithm (see Section 5.2). We can show that the terminal wealth of the UP algorithm  $S_T(\mathbf{b}^{UP})$  is in fact the average of all the experts wealth's [37]

$$\begin{aligned}
S_T(\mathbf{b}^{UP}) &= \prod_{t=1}^T \mathbf{x}_t^\top \cdot \mathbf{b}_t^{UP} \\
&= \prod_{t=1}^T \sum_{i=1}^m x_{i,t} \cdot b_{i,t}^{UP} \\
&= \prod_{t=1}^T \sum_{i=1}^m x_{i,t} \cdot \frac{\sum_{\omega=1}^{\Omega} \mathbf{b}_i^\omega S_{t-1}(\mathbf{b}^\omega)}{\sum_{\omega=1}^{\Omega} S_{t-1}(\mathbf{b}^\omega)} \quad (\text{by Eq. (10)}) \\
&= \prod_{t=1}^T \frac{\sum_{\omega=1}^{\Omega} \sum_{i=1}^m (x_{i,t} \cdot \mathbf{b}_i^\omega) S_{t-1}(\mathbf{b}^\omega)}{\sum_{\omega=1}^{\Omega} S_{t-1}(\mathbf{b}^\omega)} \\
&= \prod_{t=1}^T \frac{\sum_{\omega=1}^{\Omega} S_t(\mathbf{b}^\omega)}{\sum_{\omega=1}^{\Omega} S_{t-1}(\mathbf{b}^\omega)} \quad \text{since } S_t(\mathbf{b}^\omega) = S_{t-1} \sum_{i=1}^m \mathbf{b}_i^\omega \cdot x_{i,t} \\
&= \frac{\sum_{\omega=1}^{\Omega} \prod_{t=1}^T S_t(\mathbf{b}^\omega)}{\sum_{\omega=1}^{\Omega} \prod_{t=1}^T S_{t-1}(\mathbf{b}^\omega)} \\
&= \frac{\sum_{\omega=1}^{\Omega} S_T(\mathbf{b}^\omega)}{\sum_{\omega=1}^{\Omega} S_0(\mathbf{b}^\omega)} \\
&= \frac{1}{\Omega} \sum_{\omega=1}^{\Omega} S_T(\mathbf{b}^\omega) \quad \text{since } \sum_{\omega=1}^{\Omega} S_0(\mathbf{b}^\omega) = \sum_{\omega=1}^{\Omega} 1 = \Omega
\end{aligned}$$

Hence, the UP algorithm can be considered as a BH strategy of all  $\Omega$  experts.

## 2.5 Meta-Learning Algorithms

Meta-learning algorithms are very similar to the Universal Portfolio algorithm in that they are a 'follow-the-winner' algorithm and take a performance weighted average of a set of underlying strategies however instead of using CRP-experts as in the Universal Portfolio algorithm, these algorithms handle a variety of different classes of experts [36]. Our proposed trading algorithm will mimic a meta-learning algorithm since each expert's trading strategy will form a different class within the algorithm. Majority of the experts used in our algorithm will be developed from technical analysis and are discussed comprehensively in the following section.

## 3 Technical Analysis

Technical Analysis was formed on principles from Dow Theory [42] and uses the history of market action<sup>14</sup> to predict future movements [20]. Over the past 2 decades, the use of technical analysis for the generation of trading rules has been vast in academic literature and results have varied.

Technicians believe that anything that can possibly affect the price is actually reflected in the price and hence claim that market action should reflect all shifts in

---

<sup>14</sup>Market action refers to price and volume information/data

supply and demand [42] leading to the belief that market action is the best source of information. This implies that the technician is indirectly studying fundamentals since the data will reflect the up and down movements for given stocks in the market, however, it is important to note that technicians are not at all concerned with the reasons for these movements. They will make trading decisions based solely on market action and leave it to the fundamental analysts to explain the reasons for historical movements using news and other data.

We utilise fourteen trading rules built from technical analysis. There are typically three types of trade entry techniques: trend-following, oscillators and patterns [15]. The set of rules we consider contain a mixture of such techniques. A further three online portfolio selection techniques are implemented and are discussed in Section 3.2.

### 3.1 Technical Indicators and Trading Rules

We follow Creamer and Freund [9] and Kestner [15] in introducing and describing some of the more popular technical analysis indicators as well as a few others that are widely available. We also describe in detail the trading rules associated with the technical indicators which generate buy, sell and hold signals, most of them described as in Creamer and Freund [9].

#### 3.1.1 Simple Moving Average

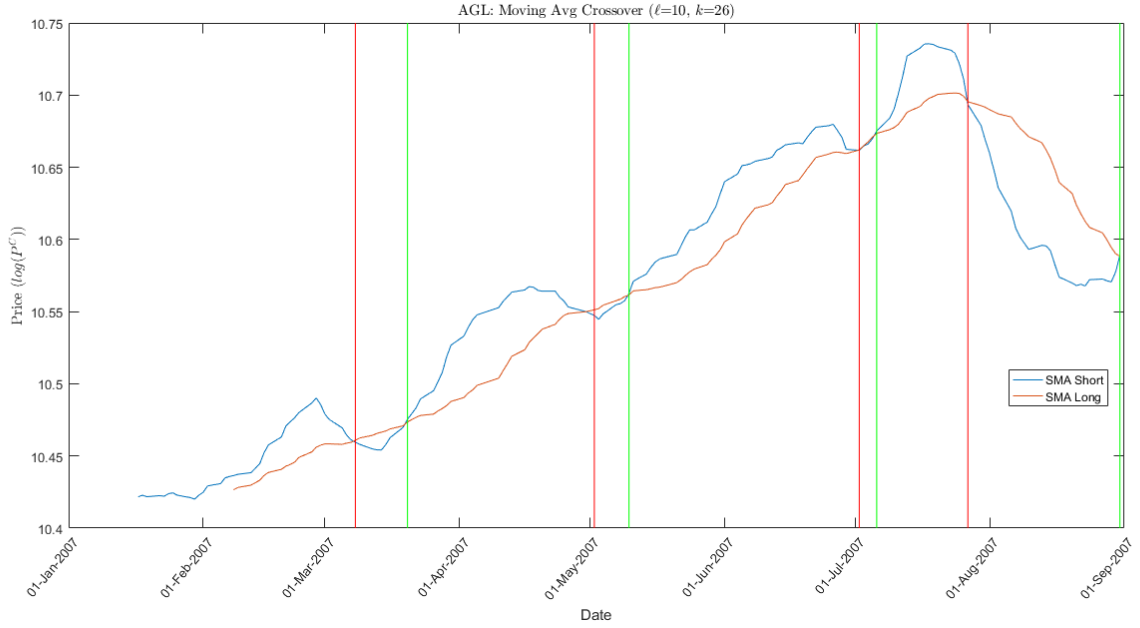
The most common moving average is the *simple moving average* (SMA). The SMA is the mean of a time series (typically of closing prices) over the last  $n$  trading days and is usually updated every trading period to take into account more recent data and drop older values. The smaller the value of  $n$ , the closer the moving average will fit to the price data

##### 3.1.1.1 SMA Indicator: $SMA_t^c(n)$

$$SMA_t(\mathbf{P}^c, n) = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i}^c \quad (11)$$

##### 3.1.1.2 Moving Average Crossover Trading Rule

The moving average crossover rule uses two SMA's, a short SMA and a longer SMA. A buy signal occurs when the faster (shorter) moving average crosses above the slower (longer) moving average, and a sell signal occurs when the shorter moving average crosses below the longer moving average.



**Figure 1:** Moving average crossover trading rule implemented on 8 months of Anglo American PLC (AGL) closing prices from 01-01-2007 to 30-08-2007. The log of the closing price (rather than just the closing price) is used for illustrative purposes. Green and red vertical lines represent buy and sell signals respectively.

### 3.1.2 Exponential Moving Average

The exponential moving average (EMA) makes use of today's close price, yesterday's moving average value and a smoothing factor ( $\alpha$ ). The smoothing factor will determine how quickly the exponential moving average responds to current market prices [15]. A simple moving average is used for the initial EMA value.

#### 3.1.2.1 Exponential Moving Average Crossover Rule

The calculation is identical to the Moving Average Crossover rule above however instead of using a SMA, an EMA is used.

#### 3.1.2.2 EMA Indicator: $EMA_t^c(n)$

$$EMA(P^c, n) = \alpha P_t^c + (1 - \alpha)EMA(P_{t-1}^c, n) \quad (12)$$

where  $\alpha = \frac{2}{n+1}$

### 3.1.3 Highest High

Highest High is the greatest high price in the last  $n$  periods.

$$HH(n) = \max(P_n^h) \quad (13)$$

where the vector with high prices of last  $n$  periods is given by  $P_n^h = (P_{t-n}^h, P_{t-n+1}^h, P_{t-n+2}^h, \dots, P_t^h)$

### 3.1.4 Lowest Low

Lowest Low is the smallest low price in the last  $n$  periods.

$$LL(n) = \min(\mathbf{P}_n^l) \quad (14)$$

where the vector with low prices of last  $n$  periods is given by  $\mathbf{P}_n^l = (P_{t-n}^l, P_{t-n+1}^l, P_{t-n+2}^l, \dots, P_t^l)$

### 3.1.5 Ichimoku Kinko Hyo

The Ichimoku Kinko Hyo (at a glance equilibrium chart) system consists of five lines and the Kumo (cloud) [43, 44, 45, 46]. The five lines all work in concert to produce the end result. The size of the Kumo is an indication of the current market volatility, where a wider Kumo is a more volatile market.

#### 3.1.5.1 Ichimoku Kinko Hyo Indicators

1. **Tenkan-sen (Conversion Line):**  $(HH(n_1) + LL(n_1))/2$
2. **Kijun-sen (Base Line):**  $(HH(n_2) + LL(n_2))/2$
3. **Chikou Span (Lagging Span):** Close plotted  $n_2$  days in the past
4. **Senkou Span A (Leading Span A):**  $(\text{Conversion Line} + \text{Base Line})/2$
5. **Senkou Span B (Leading Span B):**  $(HH(n_3) + LL(n_3))/2$
6. **Kumo (Cloud):** Area between the Leading Span A and the Leading Span B form the Cloud

Ichimoku uses three key time periods for its input parameters: Typically,  $n_1 = 7$ ,  $n_2 = 22$ , and  $n_3 = 44$ . We will keep the  $n_1$  parameter fixed at 7 but vary the other two look-back parameters  $n_2$  and  $n_3$ . If the learning algorithm did have three free model parameters,  $n_1$  could also be varied. We choose to fix  $n_1$  since it impacts the Conversion line which has the nearest correspondence to the closing price.

#### 3.1.5.2 Ichimoku Kinko Hyo Strategies

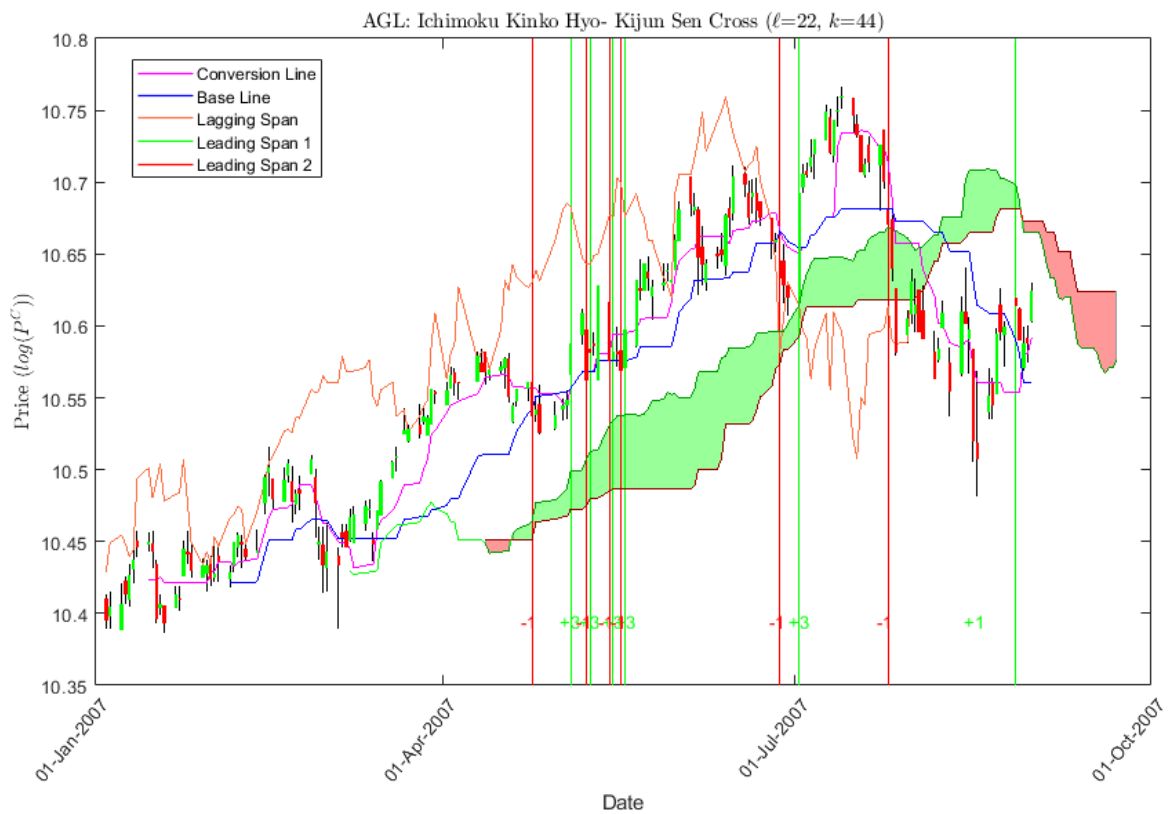
The Kijun Sen Cross strategy is one of the most powerful and reliable trading strategies within the Ichimoku system due to the fact that it can be used on nearly all time frames with exceptional results [45].

Decision	Condition
<i>Buy</i>	Kijun Sen crosses the closing price curve from the bottom up
<i>Sell</i>	Kijun Sen crosses the closing price curve from the top down
<i>Hold</i>	otherwise

**Table 2:** Ichimoku Kijun Sen Cross strategy

The buy and sell (bullish and bearish) signals are classified into three major classifications: strong, neutral and weak. The strength is determined by where the crossover occurs in relation to the cloud:

$$\begin{aligned}
 \text{Strong} &: \begin{cases} \text{buy,} & \text{bullish cross happens above the kumo} \\ \text{sell,} & \text{bearish cross happens below the kumo} \end{cases} \\
 \text{Neutral} &: \begin{cases} \text{buy,} & \text{bullish cross happens within the kumo} \\ \text{sell,} & \text{bearish cross happens within the kumo} \end{cases} \\
 \text{Weak} &: \begin{cases} \text{buy,} & \text{bullish cross happens below the kumo} \\ \text{sell,} & \text{bearish cross happens above the kumo} \end{cases}
 \end{aligned}$$



**Figure 2:** Ichimoku Kijun Sen Cross strategy trading rule implemented on 9 months of Anglo American PLC (AGL) closing prices from 01-01-2007 to 30-09-2007. The log of the closing price (rather than just the closing price) is used for illustrative purposes. Green and red candle sticks refer to bullish and bearish trading days respectively. When the Leading span 1 is greater than the Leading span 2, the outlook is bullish (green cloud) and vice versa for a bearish outlook (red cloud). The number next to each of the vertical green and red lines indicate the strength of the buy and sell signals respectively.

### 3.1.6 Momentum

#### 3.1.6.1 Momentum Indicator: $MOM_t(n)$

Momentum gives the change in the closing price over the past  $n$  periods

$$MOM_t(n) = P_t^c - P_{t-n}^c \quad (15)$$

#### 3.1.7 Momentum Trading Rule

Decision	Condition
<i>Buy</i>	$MOM_{t-1}(n) \leq EMA_t(MOM_t(n), \lambda)$ & $MOM_t(n) > EMA_t(MOM_t(n), \lambda)$
<i>Sell</i>	$MOM_{t-1}(n) \geq EMA_t(MOM_t(n), \lambda)$ & $MOM_t(n) < EMA_t(MOM_t(n), \lambda)$
<i>Hold</i>	otherwise

**Table 3:** Momentum trading rule

### 3.1.8 Acceleration

#### 3.1.8.1 Acceleration Indicator: $ACC_t(n)$

Acceleration measures the change in momentum between two consecutive periods  $t$  and  $t - 1$

$$ACC_t(n) = MOM_t(n) - MOM_{t-1}(n) \quad (16)$$

#### 3.1.8.2 Acceleration Trading Rule

Decision	Condition
<i>Buy</i>	$ACCEL_{t-1}(n) + 1 \leq 0$ & $ACCEL_t(n) + 1 > 0$
<i>Sell</i>	$ACCEL_{t-1}(n) + 1 \geq 0$ & $ACCEL_t(n) + 1 < 0$
<i>Hold</i>	otherwise

**Table 4:** Acceleration trading rule

### 3.1.9 Moving Average Convergence/Divergence Oscillator

The Moving Average Convergence/Divergence (MACD) oscillator is a momentum indicator developed by Gerald Appel and attempts to determine whether traders are accumulating stocks or distributing stocks. It is calculated by computing the difference between a short-term and a long-term moving average. The idea is then to compute the signal line, which is accomplished by taking an exponential moving average of the MACD determines instances at which to buy (oversold) and sell (oversold) when used in conjunction with the MACD [11].

### 3.1.9.1 MACD Indicators: $\text{MACD}_t(n_1, n_2)$

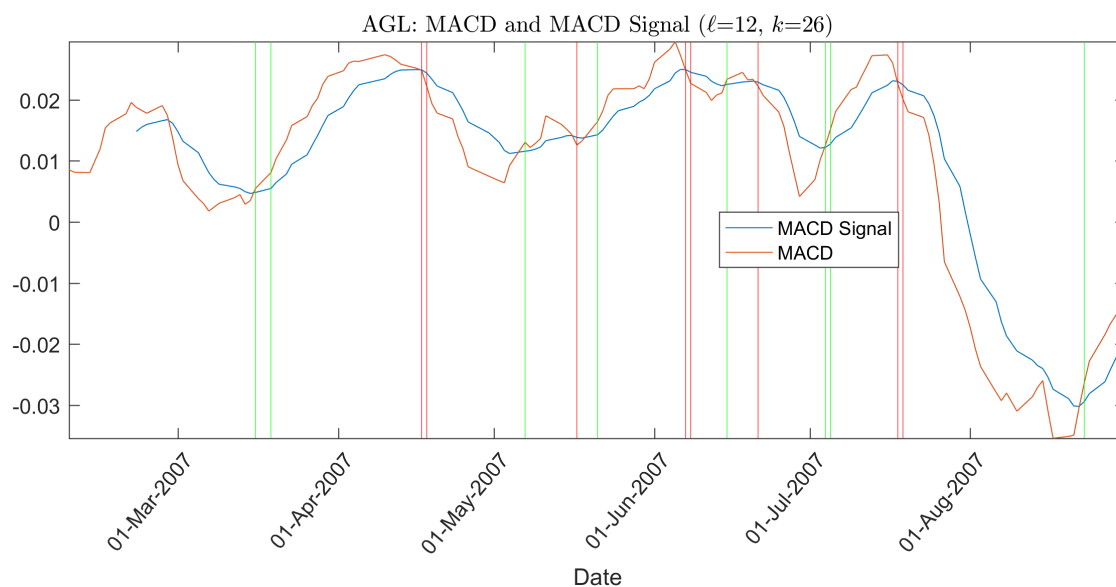
The MACD indicator is computed using the following steps:

1.  $\text{LongEMA}_t = \text{EMA}_t(P^c, n_2)$
2.  $\text{ShortEMA}_t = \text{EMA}_t(P^c, n_1)$
3.  $\text{MACD}_t(n_1, n_2) = \text{ShortEMA}_t - \text{LongEMA}_t$
4.  $\text{SignalLine}_t(n_1, n_2, n_3) = \text{EMA}_t(\text{MACD}_t(n_2, n_1), n_3)$
5.  $\text{MACDS}_t(n_1, n_2, n_3) = \text{MACD}_t(n_1, n_2) - \text{SignalLine}_t(n_1, n_2, n_3)$

### 3.1.9.2 MACD Trading Rule

Decision	Condition
<i>Buy</i>	$\text{MACD}_{t-1}(n_2, n_1) \leq \text{MACDS}_t(n_2, n_1, n_3)$ & $\text{MACD}_t(n_2, n_1) > \text{MACDS}_t(n_2, n_1, n_3)$
<i>Sell</i>	$\text{MACD}_{t-1}(n_2, n_1) \geq \text{MACDS}_t(n_2, n_1, n_3)$ & $\text{MACD}_t(n_2, n_1) < \text{MACDS}_t(n_2, n_1, n_3)$
<i>Hold</i>	otherwise

**Table 5:** MACD trading rule



**Figure 3:** MACD trading rule implemented on 9 months of Anglo American PLC (AGL) closing prices from 01-01-2007 to 30-08-2007. Green and red vertical lines represent buy and sell signals respectively.

### 3.1.10 Fast Stochastics

Fast stochastic oscillator shows the location of the closing price relative to the high-low range, expressed as a percentage, over a given number of periods as specified by

a look-back parameter.

### 3.1.10.1 Fast Stochastic Indicators

**Fast%K<sub>t</sub>(n) :**

$$\text{Fast\%K}_t(n) = \frac{P_t^c - \text{LL}(n)}{\text{HH}(n) - \text{LL}(n)} \quad (17)$$

**Fast%D<sub>t</sub>(n) :**

$$\text{Fast\%D}_t(n) = \text{SMA}_t(\text{Fast\%K}_t(n), 3) \quad (18)$$

### 3.1.10.2 Fast Stochastic Trading Rule

Decision	Condition
<i>Buy</i>	$\text{Fast\%K}_{t-1}(n) \leq \text{Fast\%D}_t(n)$ & $\text{Fast\%K}_t(n) > \text{Fast\%D}_t(n)$
<i>Sell</i>	$\text{Fast\%K}_{t-1}(n) \geq \text{Fast\%D}_t(n)$ & $\text{Fast\%K}_t(n) < \text{Fast\%D}_t(n)$
<i>Hold</i>	otherwise

**Table 6:** Fast stochastic trading rule

### 3.1.11 Slow Stochastics

The slow stochastic oscillator is very similar to the fast stochastic indicator in that it shows the location of the closing price relative to the high-low range over a given number of periods but only differs in the way that it is calculated. It is in fact just a moving average of the fast stochastic indicator. The fast stochastic indicator will typically be more sensitive to the closing price and will thus result in more frequent trading signals.

#### 3.1.11.1 Slow Stochastic Indicators

**Slow%K<sub>t</sub>(n):**

$$\text{Slow\%K}_t(n) = \text{SMA}_t(\text{Fast\%K}_t(n), 3) \quad (19)$$

**Slow%D<sub>t</sub>(n):**

$$\text{Slow\%D}_t(n) = \text{SMA}_t(\text{Slow\%K}_t(n), 3) \quad (20)$$

#### 3.1.11.2 Slow Stochastic Trading Rule

Decision	Condition
----------	-----------

<i>Buy</i>	$\text{Slow}\%K_{t-1}(n) \leq \text{Slow}\%D_t(3) \ \& \ \text{Slow}\%K_t(n) > \text{Slow}\%D_t(3)$
<i>Sell</i>	$\text{Slow}\%K_{t-1}(n) \geq \text{Slow}\%D_t(3) \ \& \ \text{Slow}\%K_t(n) < \text{Slow}\%D_t(3)$
<i>Hold</i>	otherwise

**Table 7:** Slow stochastic trading rule

### 3.1.12 Relative Strength Index

Relative Strength Index (RSI) compares the periods that stock prices finish up (closing price higher than the previous previous) against those periods that stock prices finish down (closing price lower than the previous period) [9].

#### 3.1.12.1 RSI Indicator: $\text{RSI}_t(n)$

$$\text{RSI}_t(n) = 100 - \frac{100}{1 + \frac{\text{SMA}_t(\mathbf{P}_n^{\text{up}}, n_1)}{\text{SMA}_t(\mathbf{P}_n^{\text{dwn}}, n_1)}} \quad (21)$$

where [9]

$$P_t^{\text{up}} = \begin{cases} P_t^c & \text{if } P_{t-1}^c < P_t^c \\ \text{NaN} & \text{otherwise} \end{cases} \quad (22)$$

$$P_t^{\text{dwn}} = \begin{cases} P_t^c & \text{if } P_{t-1}^c > P_t^c \\ \text{NaN} & \text{otherwise} \end{cases} \quad (23)$$

and

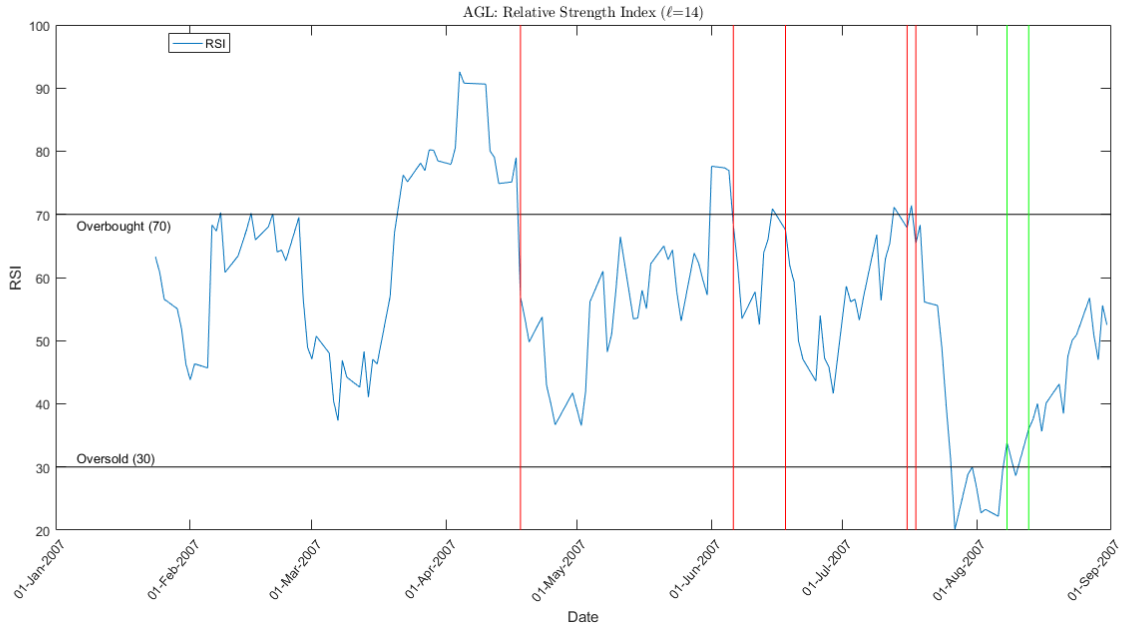
$$\mathbf{P}_n^{\text{up}} = (P_{t-n}^{\text{up}}, P_{t-n+1}^{\text{up}}, \dots, P_t^{\text{up}}) \quad (24)$$

$$\mathbf{P}_n^{\text{dwn}} = (P_{t-n}^{\text{dwn}}, P_{t-n+1}^{\text{dwn}}, \dots, P_t^{\text{dwn}}) \quad (25)$$

#### 3.1.12.2 RSI Trading Rule

Decision	Condition
<i>Buy</i>	$\text{RSI}_{t-1}(n) \leq 30 \ \& \ \text{RSI}_t(n) > 30$
<i>Sell</i>	$\text{RSI}_{t-1}(n) \geq 70 \ \& \ \text{RSI}_t(n) < 70$
<i>Hold</i>	otherwise

**Table 8:** Relative strength index (RSI) trading rule



**Figure 4:** RSI trading rule implemented on 9 months of Anglo American PLC closing prices from 01-01-2007 to 30-08-2007. Green and red vertical lines represent buy and sell signals respectively.

### 3.1.13 Moving Average Relative Strength Index

Moving Average Relative Strength Index (MARS<sub>I</sub>) is an indicator that smooths out the action of RSI indicator [47].

#### 3.1.13.1 MARS<sub>I</sub> Indicator: $MARS_{I_t}(n_1, n_2)$

MARS<sub>I</sub> is calculated by simply taking an  $n_2$ -period SMA of the RSI indicator.

$$MARS_{I_t}(n_1, n_2) = SMA(RSI_t(n_1), n_2) \quad (26)$$

#### 3.1.13.2 MARS<sub>I</sub> Trading Rule

Rather buying or selling when RSI crosses upper and lower thresholds (30 and 70) as in the RSI trading rule above, buy and sell signals are generated when the SMA of the MARS<sub>I</sub> crosses above or below the thresholds [47].

Decision	Condition
<i>Buy</i>	$MARS_{I_{t-1}}(n) \leq 30 \ \& \ MARS_{I_t}(n) > 30$
<i>Sell</i>	$MARS_{I_{t-1}}(n) \geq 70 \ \& \ MARS_{I_t}(n) < 70$
<i>Hold</i>	otherwise

**Table 9:** MARS<sub>I</sub> trading rule

### 3.1.14 Bollinger Band

Bollinger bands uses a SMA ( $Boll_t^m(n)$ ) as it's reference point (known as the median band) with regards to the upper and lower Bollinger bands denoted by  $Boll_t^u(n)$  and  $Boll_t^d(n)$  respectively and are calculated as functions of standard deviations ( $s$ ). When the closing price crosses above (below) the upper (lower) Bollinger band, it is a sign that the market is overbought (oversold) [9].

#### 3.1.14.1 Bollinger Band Indicator: $Boll_t^m(n)$

$$\begin{aligned} Boll_t^m(n) &= SMA_t^c(n) \\ \text{Upper Bollinger band: } & Boll_t^m(n) + s\sigma_t^2(n) \\ \text{Lower Bollinger band: } & Boll_t^m(n) - s\sigma_t^2(n) \end{aligned} \quad (27)$$

where  $s$  is typically chosen to be 2.

#### 3.1.14.2 Bollinger Trading Rule

Decision	Condition
<i>Buy</i>	$P_{t-1}^c \geq Boll_t^d(n) \ \& \ P_t^c \geq Boll_t^u(n)$
<i>Sell</i>	$P_{t-1}^c \leq Boll_t^d(n) \ \& \ P_t^c > Boll_t^u(n)$
<i>Hold</i>	otherwise

**Table 10:** Bollinger trading rule

### 3.1.15 Price Rate-Of-Change

#### 3.1.15.1 PROC Indicator: $PROC_t(n)$

The rate of change of the time series of closing prices  $P_t^c$  over the last  $n$  periods expressed as a percentage

$$PROC_t(n) = 100 \cdot \frac{P_t^c - P_{t-n}^c}{P_{t-n}^c} \quad (28)$$

#### 3.1.15.2 PROC Trading Rule

Decision	Condition
<i>Buy</i>	$PROC_{t-1}(n) \leq 0 \ \& \ PROC_t(n) > 0$
<i>Sell</i>	$PROC_{t-1}(n) \geq 0 \ \& \ PROC_t(n) < 0$
<i>Hold</i>	otherwise

**Table 11:** Price rate of change (PROC) trading rule

### 3.1.16 Williams %R

#### 3.1.16.1 Williams %R Indicator: $Will_t(n)$

Williams Percent Range (Williams %R) is calculated similarly to the fast stochastic oscillator and shows the level of the close relative to the highest high in the last  $n$  periods

$$Will_t(n) = \frac{HH(n) - P_t^c}{HH(n) - LL(n)} \cdot (-100) \quad (29)$$

#### 3.1.16.2 Williams %R Trading Rule

Decision	Condition
<i>Buy</i>	$Will_{t-1}(n) \geq -20$ & $Will_t(n) < -80$
<i>Sell</i>	$Will_{t-1}(n) \leq -20$ & $Will_t(n) > -80$
<i>Hold</i>	otherwise

**Table 12:** Williams %R trading rule

### 3.1.17 Parabolic SAR

Parabolic Stop and Reverse (SAR), developed by J. Wells Wilder, is a trend indicator formed by a parabolic line made up of dots at each time step [48]. The dots are formed using the most recent Extreme Price and an acceleration factor (AF), 0.02, which increases each time a new Extreme Price (EP) is reached. The AF has a maximum value of 0.2 to prevent it from getting too large. Extreme Price represents the highest (lowest) value reached by the price in the current up-trend (down-trend). The acceleration factor determines where in relation to the price the parabolic line will appear by increasing by the value of the AF each time a new EP is observed and thus affects the rate of change of the Parabolic SAR.

#### 3.1.17.1 SAR Indicator: $SAR(n)$

The steps involved in calculating the SAR indicator are as follows:

1. initialise variables: trend is initially set to 1 (up-trend), EP to zero,  $AF_0$  to 0.02,  $SAR_0$  to the closing price at time zero ( $P_0^c$ ), lastHigh to high price at time zero ( $P_0^h$ ) and lastLow to the low price at time zero ( $P_0^l$ )
2. update parameters: EP, lastHigh, lastLow and AF based on where the current high is in relation to the lastHigh (up-trend) or where the current low is in relation to the lastLow (down-trend)
3. compute the next period SAR value: update time  $t + 1$  SAR value,  $SAR_{t+1}$ , using Eq. (30)
4. modify the SAR value and the parameters for a change in trend: modify the  $SAR_{t+1}$  value, AF, EP lastLow, lastHigh and the trend based on the trend and its value in relation to the current low  $P_t^l$  and current high  $P_0^h$
5. go to next time period and return to step 2

Below is the formula for the Parabolic SAR for time  $t+1$  calculated using the previous value at time  $t$ :

$$\text{SAR}_{t+1} = \text{SAR}_t + \alpha(\text{EP} - \text{SAR}_t) \quad (30)$$

### 3.1.17.2 SAR Trading Rule

Decision	Condition
<i>Buy</i>	$\text{SAR}_{t-1} \geq P_{t-1}^c$ & $\text{SAR}_t < P_t^c$
<i>Sell</i>	$\text{SAR}_{t-1} \leq P_{t-1}^c$ & $\text{SAR}_t > P_t^c$
<i>Hold</i>	otherwise

**Table 13:** SAR trading rule

## 3.2 Trend Following and Contrarian Mean Reversion Strategies

The zero-cost BCRP (trend following), zero-cost anti-BCRP and zero-cost anti-correlation (both contrarian mean-reverting) algorithms are explained in more detail in the following subsections.

### 3.2.1 Zero-Cost BCRP

Zero-cost BCRP is the zero-cost (long/short and self-financing) version of the BCRP strategy and is a trend following algorithm in that long positions are taken in stocks during upward trends while short positions are taken during downward trends. The idea is to first find the portfolio controls that maximise the expected utility of wealth using all in-sample price relatives according to a given constant level of risk aversion. The resulting portfolio equation is what is known to be the Mutual Fund Separation Theorem [40]. The second set of portfolio controls in the mutual fund separation theorem (active controls) is what we will use as the set of controls for the zero-cost BCRP strategy and are given by:

$$\mathbf{b} = \frac{1}{\gamma} \Sigma^{-1} \left[ \mathbb{E}[\mathbf{R}] - \mathbf{1} \frac{\mathbf{1}^\top \Sigma^{-1} \mathbb{E}[\mathbf{R}]}{\mathbf{1}^\top \Sigma^{-1} \mathbf{1}} \right] \quad (31)$$

where  $\Sigma^{-1}$  is the inverse of the covariance matrix of returns for all  $m$  stocks,  $\mathbb{E}[\mathbf{R}]$  is vector of expected returns of the stocks,  $\mathbf{1}$  is a vector of ones of length  $m$  and  $\gamma$  is the risk aversion parameter. Eq. (31) is the risky Optimal Tactical Portfolio (OTP) that takes optimal risky bets given the risk aversion  $\gamma$ . The risk aversion is selected during each period  $t$  such that the controls are unit leverage and hence  $\sum_i^m |b_i| = 1$ .

The covariance matrix and expected returns for each period  $t$  are computed using the set of price relatives from today back  $\ell$  days (short-term look-back parameter).

### 3.2.2 Zero-Cost Anti-BCRP

Zero-cost anti-BCRP is exactly the same as zero-cost BCRP except that we reverse the sign of the expected returns vector such that  $\mathbb{E}[\mathbf{R}] = -\mathbb{E}[\mathbf{R}]$ .

### 3.2.3 Zero-Cost Anti-Correlation

Zero-cost anti-correlation (Z-Anticor) is an adapted version of the Anticor algorithm developed in Borodin *et al.* [38] to allow for long/short portfolios. The first step is to extract the price relatives for the two most recent sequential windows each of length  $\ell$ . Let  $\boldsymbol{\mu}_2^\ell$  and  $\boldsymbol{\mu}_1^\ell$  denote the average log-returns of the  $\ell$  price relatives in the most recent window ( $\mathbf{x}_t^{t-\ell+1}$ ) and the price relatives in the window prior to that ( $\mathbf{x}_{t-\ell}^{t-2\ell+1}$ ) respectively. Also, let the lagged covariance matrix and lagged correlation matrix be defined as follows:

$$\Sigma^\ell = \frac{1}{\ell-1} [(\mathbf{x}_{t-\ell}^{t-2\ell+1} - \mathbf{1}) - \mathbf{1}^\top \boldsymbol{\mu}_1^\ell]^\top [(\mathbf{x}_t^{t-\ell+1} - \mathbf{1}) - \mathbf{1}^\top \boldsymbol{\mu}_2^\ell] \quad (32)$$

$$P_{ij}^\ell = \frac{\Sigma_{ij}^\ell}{\sqrt{\Sigma_{ij}^\ell \Sigma_{ij}^\ell}} \quad (33)$$

Z-Anticor then computes the claim that each pair of stocks have on one another, denoted  $\text{claim}_{i \rightarrow j}^\ell$ , which is the claim of stock  $j$  on stock  $i$ . This is the extent to which we want to shift our allocation from stock  $i$  to stock  $j$  [38].  $\text{claim}_{i \rightarrow j}^\ell$  exists and is thus non-zero if and only if  $\boldsymbol{\mu}_2 > \boldsymbol{\mu}_1$  and  $P_{ij} > 0$ . The claim is then calculated as

$$\text{claim}_{i \rightarrow j}^\ell = P_{ij}^\ell + \max(-P_{ii}^\ell, 0) + \max(-P_{jj}^\ell, 0) \quad (34)$$

The adaptation we propose for zero-cost portfolios for the amount of transfer that take places from stock  $i$  to stock  $j$  is given by:

$$\text{transfer}_{i \rightarrow j}^\ell = \frac{1}{3} \text{claim}_{i \rightarrow j}^\ell \quad (35)$$

Finally, we calculate the expert control for the  $i^{\text{th}}$  stock in period  $t+1$  as follows:

$$\mathbf{h}_{t+1}^n(i) = \mathbf{h}_t^n(i) + \sum_i [\text{transfer}_{j \rightarrow i}^\ell - \text{transfer}_{i \rightarrow j}^\ell] \quad (36)$$

Each control is then normalised in order to ensure unit leverage on the set of controls.

## 4 Data and Data Processing

### 4.1 Daily Data

The daily data is sourced from Thomson Reuters and contains data corresponding to all stocks listed on the JSE Top 40<sup>15</sup>. The data set consists of data for 42 stocks over the period 01-01-2005 to 29-04-2016 however we will only utilise the stocks which traded more than 60% of the time over this period. Removing such stocks leaves us with a total of 31 stocks. The data comprises of the opening price ( $P^o$ ), closing price ( $P^c$ ), lowest price ( $P^l$ ), the highest price ( $P^h$ ) and daily traded volume ( $V$ ) (OHLCV). In additions to these 31 stocks, we also require a risk-free asset for balancing the portfolio. We make the choice of trading the Short Term Fixed Interest (STeFI)

<sup>15</sup>Please refer to Appendix A.1 for the full names of the Bloomberg ticker symbols below.

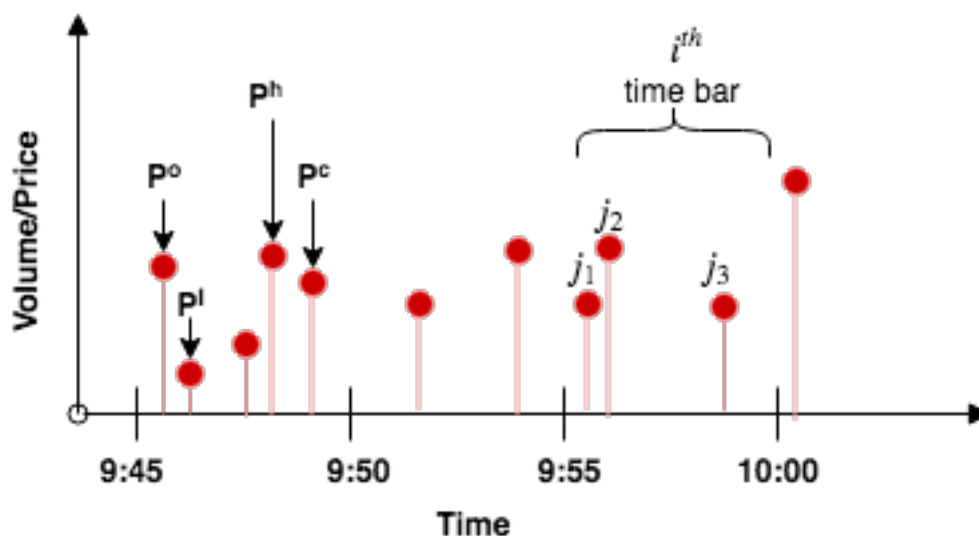
index. The STeFI benchmark is a proprietary index that measures the performance of Short Term Fixed Interest or money market investment instruments in South Africa. It is constructed by Alexander Forbes (and formerly by the South African Futures Exchange (SAFEX)) and has become the industry benchmark for short-term cash equivalent investments (up to 12 months) [60].

## 4.2 Intraday-Daily Data

Bloomberg is the source of all tick (intraday) data used in this paper. The data set consists of 30 of the Top 40 stocks on the JSE from 02-01-2018 to 29-06-2018. The data is then sampled at 5-minute intervals to create an OHLCV entry for all 5-minute intervals over the 6-month period. We remove the first 10 minutes and last 20 minutes of the continuous trading session (9:00-16:50) as the market is relatively illiquid and volatile during these times which may lead to spurious trade decisions. We are thus left with 88 OHLCV entries for each stock on any given day. In addition to the intraday data, daily OHLCV data for the specified period is required for the last transaction on any given day. As in the daily data case, we make use of the STeFI index as the risk-free asset and hence the daily entries for the STeFI index are included in this data set. The data was sourced from a Bloomberg terminal using the R Bloomberg API, *Rblpapi*, and all data processing is done in MATLAB to get the data into the required form for the learning algorithm.

### 4.2.1 Time Bar Aggregation

Once we have extracted the transaction only data, we can then convert the tick data into regularly sampled time bars (intervals). There are two main factors contributing to the reason for which we transform the irregularly sampled tick data into regularly sampled time bars (see Figure 5 for illustration), the first being noise reduction, and the second being too much or too little data entering the trading system [11, 49]. One major contributing factor to the noise is the presence of outlier transactions which may lead to the system activating a transaction based on a false signal [11]. The issue with too much (too little) data is due to the asynchronous format of the tick data whereby there may be periods whereby there are too many (too few) data points for the learning algorithm to make appropriate buy and sell decisions. The case in which there is an abundance of data being streamed into the system is of particular concern in online learning algorithms as the algorithm may not be able to process the data in adequate time to make decisions and may even halt due to a shortage of storage.



**Figure 5:** Illustration of how trades arriving at irregular time intervals are sampled over regular intervals (5-minutes).

In order to compute from the OHLCV bar data, we compute the opening price ( $P^o$ ), closing price ( $P^c$ ), lowest price ( $P^l$ ), the highest price ( $P^h$ ) and the total volume traded in the 5-minute time bar is totalled ( $V$ ) to get a complete OHLCV entry for each 5-minute interval  $i$  (see Figure 5). The order of the prices in each interval is taken as reported by the exchange or data provider.

## 5 Learning Technical Trading

Rather than using a back-test in the attempt to find the single most profitable strategy, we produce a large population of trading strategies (experts) and use an adaptive algorithm to aggregate the performances of the experts to arrive at a final portfolio to be traded. The idea of the online learning algorithm is to consider a population of experts created using a large set of technical trading strategies generated from a variety of parameters and to form an aggregated portfolio of stocks to be traded by considering the wealth performance of the expert population. During each trading period, experts trade and execute buy (1), sell (-1) and hold (0) signals independent of one another based on each of their individual strategies. The signals are then transformed into a set of portfolio weights (controls) such that their sum is identical to zero (zero-cost) and the strategy becomes self-funding. We also require that the portfolio is unit leveraged and hence the absolute sum of controls is equal to one. This is to avoid having to introduce a margin account into the trading mechanics.<sup>16</sup> Based on each individual expert's accumulated wealth up until some time  $t$ , a final aggregate portfolio for the next period  $t + 1$  is formed by creating a performance weighted combination of the experts. Experts who perform better in period  $t$  will have a larger relative contribution toward the aggregated portfolio to be implemented in period  $t + 1$  than those who perform poorly. Below, we describe the methodology for generating the expert population.

<sup>16</sup>We could in fact have considered leveraged trading but avoided this for simpler trading mechanics

## 5.1 Expert Generating Algorithm

### 5.1.1 Technical Trading

Technical trading, as mentioned above, refers to the practice of using trading rules derived from technical analysis indicators based on prices (OHLC), volume traded or a combination of both (OHLCV) to generate trading signals. In addition to the set of technical trading strategies, we implement adapted versions<sup>17</sup> of three other popular portfolio selection algorithms each of which has been adapted to generate zero-cost portfolio controls. An explanation of these three algorithms is provided in Section 3.2 while each of the technical strategies are described in Section 3.1.

In order to produce the broad population of experts, we consider combinations among a set of four model parameters. The first of these parameters is the underlying strategy of a given expert,  $\omega$ , which corresponds to the set of technical trading and trend-following strategies where the total number of different trading rules is denoted by  $W$ . Each of the rules requires at most two parameters to generate a buy, sell or hold signal at each time period  $t$ . The two parameters represent the number of short and long-term look-back periods necessary for the indicators used in the rules. These parameters will determine the amount of historic data considered in the computation of each rule. We will denote the vector of short-term parameters by  $\mathbf{n}_1$  and the long-term parameters by  $\mathbf{n}_2$  which make up two of the four model parameters. Let  $L = |\mathbf{n}_1|$  and  $K = |\mathbf{n}_2|$ <sup>18</sup> be the number of short-term and long-term look-back parameters respectively. Also, we denote the number of trading rules which utilise one parameter by  $W_1$  and the number of trading rules utilising two parameters by  $W_2$  and hence  $W = W_1 + W_2$ .

The final model parameter, denoted by  $\mathbf{c}$ , refers to object clusters where  $\mathbf{c}(i)$  is the  $i^{\text{th}}$  object cluster and  $C$  is the number of object clusters. We will consider four object clusters; the trivial cluster which contains all the stocks and the three major sector clusters of stocks on the JSE, namely, Resources, Industrials and Financials<sup>19</sup>. The algorithm will loop over all combinations of these four model parameters calling the appropriate strategies ( $\omega$ ), stocks ( $\mathbf{c}$ ) and amount of historic data ( $\mathbf{n}_1$  and  $\mathbf{n}_2$ ) to create a buy, sell or hold signal at each time period  $t$ . Each combination of  $\omega(i)$  for  $i = 1, \dots, W$ ,  $\mathbf{c}(j)$  for  $j = 1, \dots, C$ ,  $\mathbf{n}_1(\ell)$  for  $\ell = 1, \dots, L$  and  $\mathbf{n}_2(k)$  for  $k = 1, \dots, K$  will represent an expert. It is clear that some experts may trade all the stocks (trivial clusters) and others will trade subsets of the stocks (Resources, Industrials and Financials). It is also important to note that for rules with two parameters, the loop over the long-term parameters will only activate at indices  $k$  for which  $\mathbf{n}_1(\ell) < \mathbf{n}_2(k)$  where  $\ell$  and  $k$  represent the loop index over the short and long-term parameters respectively. The total number of experts,  $\Omega$ , is then given by

$$\begin{aligned} \Omega &= \text{no. of experts with 1 parameter} \\ &\quad + \text{no. of experts with 2 parameter} \\ &= C \cdot L \cdot W_1 + C \cdot W_2 \cdot \sum \left[ \sum (\mathbf{n}_2 > \max(\mathbf{n}_1)) : \right. \\ &\quad \left. \sum (\mathbf{n}_2 > \min(\mathbf{n}_1)) \right] \end{aligned}$$

<sup>17</sup>Adapted to allow for long/short trading rather than long-only

<sup>18</sup> $|\cdot|$  denotes the dimension of a vector

<sup>19</sup>See Appendix A.1 for a breakdown of the three sectors into their constituents for daily data and Appendix A.2 for the corresponding breakdown for intraday data

We will denote each expert's strategy<sup>20</sup> by  $\mathbf{h}_t^n$  which is an  $(m + 1) \times 1$  vector representing the portfolio weights of the  $n^{\text{th}}$  expert for all  $m$  stocks and the risk-free asset at time  $t$ . Here,  $m$  refers to the chosen number of stocks to be passed into the expert generating algorithm. As mentioned above, from the set of  $m$  stocks, each expert will not necessarily trade all  $m$  stocks (unless the expert trades the trivial cluster), since of those  $m$  stocks, only a hand full of stocks will fall into a given sector constituency. This implies that even though we specify each expert's strategy ( $\mathbf{h}_t^n$ ) to be an  $(m + 1) \times 1$ , we will just set the controls to zero for the stocks which the expert does not trade in their portfolio. Denote the expert control matrix  $\mathbf{H}_t$  made up of all  $n$  experts' strategies at time  $t$  for all  $m$  stocks *i.e.*  $\mathbf{H}_t = [\mathbf{h}_t^1, \dots, \mathbf{h}_t^n]$ . In order to choose the  $m$  stocks to be traded, we take the  $m$  most liquid stocks over a specified number of days denoted by  $\delta_{\text{liq}}$ . We make the choice of using average daily volume (ADV) as a proxy for liquidity.<sup>21</sup> ADV is simply the average volume traded for a given stock over a period of time. The ADV for stock  $m$  over the past  $\delta_{\text{liq}}$  periods is

$$\text{ADV}_m = \frac{1}{\delta_{\text{liq}}} \sum_{t=1}^{\delta_{\text{liq}}} V_t^m \quad (37)$$

where  $V_t^m$  is the volume of the  $m^{\text{th}}$  stock at period  $t$ . The  $m$  stocks with the largest ADV will then be fed into the algorithm for trading.

### 5.1.2 Transforming Signals into Weights

In this section we describe how each individual expert's set of trading signals at each time period  $t$  are transformed into a corresponding set of portfolio weights (controls) which constitute the expert's strategy ( $\mathbf{h}_t^n$ ). For the purpose of generality, we refer to the stocks traded by a given expert as  $m$  even though the weights of many of these  $m$  stocks will be zero for multiple periods as the expert will only be considering a subset of these stocks depending on which object cluster the expert trades.

Suppose it is currently time period  $t$  and the  $n^{\text{th}}$  expert is trading  $m$  stocks. Given that there are  $m$  stocks in the portfolio,  $m$  trading signals will need to be produced at each trading period. The risk-free assets purpose will be solely to balance the portfolio given the set of trading signals. Given the signals for the current time period  $t$  and previous period  $t - 1$ , all hold signals at time  $t$  are replaced with the corresponding non-zero signals from time  $t - 1$  as the expert retains his position in these stocks<sup>22</sup>. All non-hold signals at time  $t$  are of course not replaced by the previous periods signals as the expert has taken a completely new position in the stock. This implies that when the position in a given stock was short at period  $t - 1$  for example and the current periods ( $t$ ) signal is long then the expert takes a long position in the stock rather than neutralising the previous position. Before computing the portfolio controls, we compute a combined signal vector made up of signals from time period  $t - 1$  and time  $t$  using the idea discussed above. We will refer to this combined set of signals as *output signals*. We then consider four possible cases of the output signals at time  $t$  for a given expert:

<sup>20</sup>When we refer to strategy, we are talking about the weights of the stocks in the expert's portfolio. As mentioned previously, we will also refer to these weights as controls

<sup>21</sup>Other indicators of liquidity do exist such as the width of the bid-ask spread and market depth, however ADV provides a simple approximation of liquidity

<sup>22</sup>Only the position is retained from the previous period (long/short) not the magnitude of the weight held in the stock

1. All output signals are hold (0)
2. All output signals are non-negative (0 or 1)
3. All output signals are non-positive (0 or -1)
4. There are combinations of buy, sell and hold signals (0, 1 and -1) in the set of output signals

Due to the fact that cases 2 (long-only) and 3 (short-only) exist, we need to include a risk-free asset in the portfolio so that we can enforce the self-financing constraint; the controls must sum to zero  $\sum_i w_i = 0$ . We refer to such portfolios as *zero-cost* portfolios. In addition, we implement a leverage constraint by ensuring that the absolute value of the controls sums to unity:  $\sum_i |w_i| = 1$ .

To compute the controls for case 1, we set all stock weights and the risk-free asset weight to zero so that the expert does not allocate any capital in this case since the output signals are all zero.

For case 2, we compute the standard deviations of the set of stocks which resulted in buy (positive) signals from the output signals using their closing prices over the last 90 days for daily trading and the last 90 trading periods for intraday-daily trading<sup>23</sup> and use these standard deviations to allocate a weight that is proportional to the volatility (more volatile stocks receive higher weight allocations). Let the number of buy signals from the set of output signals be denoted by  $nb$  and denote the vector of standard deviations of stocks with non-zero output signals by  $\sigma_+$ . Then the weight allocated to stocks with positive signals is given by

$$\mathbf{w} = 0.5 \cdot \frac{1}{\sum_i \sigma_+(i)} \cdot \sigma_+ \quad (38)$$

where the lowest value of  $\sigma_+(i)$  corresponds to the least volatile stock and vice versa for large  $\sigma_+(i)$ . This equation ensures that  $\sum_i w_i = 0.5$ . We then short the risk-free asset with a weight of one half ( $w_{rf} = -0.5$ ). This allows us to borrow using the risk-free asset and purchase the corresponding stocks within which we take a long position.

Case 3 is similar to Case 2 above, however instead of having positive output signals, all output signals are negative. Again, we compute standard deviations of the set of stocks which resulted in sell (negative) signals from the output signals using their closing prices over the last last 90 days for daily trading and the last 90 trading periods for intraday-daily trading. Let the number of sell signals from the set of output signals be denoted by  $ns$  and denote the vector of standard deviations of stocks with non-zero output signals by  $\sigma_-$ . Then the weight allocated to stocks which have short positions is given by

$$\mathbf{w} = -0.5 \cdot \frac{1}{\sum_i \sigma_-(i)} \cdot \sigma_- \quad (39)$$

We then take a long position in the risk-free asset with a weight of one half ( $w_{rf} = 0.5$ ).

For case 4, we use the similar methodology to that discussed above in Case 2 and

---

<sup>23</sup>See Section 5.3 for details on intraday-daily trading

3. To compute the weights for the short assets we use the formula

$$\mathbf{w} = -0.5 \cdot \frac{1}{\sum_i \sigma_-(i)} \cdot \boldsymbol{\sigma}_- \quad (40)$$

Similarly, for the long assets we have

$$\mathbf{w} = 0.5 \cdot \frac{1}{\sum_i \sigma_+(i)} \cdot \boldsymbol{\sigma}_+ \quad (41)$$

We then set the risk-free rate to be equal to  $\sum_i w_i$  in order to enforce the self-financing and fully invested constraints. Finally, assets which had hold signals have their weights are set to zero.

The method described above is what we will refer to as the *volatility loading* method for transforming signals into controls. A second method is considered, called the *inverse volatility loading* method, and is defined similarly to the method described above, however, instead of multiplying through by the volatility vector in each of the above cases, we multiply through by the inverse of the volatility vector (element-wise inverses). We will not implement the inverse volatility loading method in this study as the results of the two methods are similar.

Algorithm 1 shows the algorithm outline for the Expert Generating Algorithm. The Expert Generating Algorithm calls the *controls* function which transforms trading signals into portfolio controls. The *controls* function is made up of two parts, the first being to compute the output signals as discussed in Section 5.1.2 which is outlined in Algorithm 2 and the second part is used to transform the output signals into portfolio controls which is outlined in Algorithm 3.

## 5.2 Online Learning Algorithm

Given that we now have a population of experts each with their own controls,  $\mathbf{h}_t^n$ , we implement the online learning algorithm to aggregate the expert's strategies at time  $t$  based on their performance and form a final single portfolio to be used in the following period  $t + 1$  which we denote  $\mathbf{b}_t$ . The aggregation scheme used is inspired by the Universal Portfolio (UP) strategy taken from the work done by [2, 3] and a modified version proposed by [5]. Although, due to the fact that we have several different base experts as defined by the different trading strategies rather than Cover's [3] constant rebalanced UP strategy, our algorithm is better defined as a meta-learning algorithm [36]. We use the subscript  $t$  since the portfolio is created using information only available at time  $t$  even though the portfolio is implemented in the following time period. The algorithm will run from the initial time  $t_{min}$ <sup>24</sup> which is taken to be 2 until terminal time  $T$ .  $t_{min}$  is required to ensure there is sufficient data to compute a return for the first active trading day. We must point out here that experts will only actively begin making trading decisions once there is sufficient data to satisfy their look-back parameter(s) and subsequently, since the shortest look-back parameter is 4 periods, the first trading decisions will only be made during day 5. The idea is to take in  $m$  stock's OHLCV values at each time period which we will denote by  $\mathbf{X}_t$ . We then compute the price relatives at each time period  $t$  given by  $\mathbf{x}_t = (x_{1,t}, \dots, x_{m,t})$

---

<sup>24</sup>This is the time at which data begins being fed into the expert generating algorithm however experts will only begin making active trading decisions once they have sufficient data (6 data points is the smallest look-back period for any expert)

**Algorithm 1** Expert Generating Algorithm**Require:**


---

```

1: 1. OHLCV prices up to current time  $\mathbf{X}_t$ 
   2. short-term parameters  $\mathbf{n}_1$ 
   3. long-term parameters  $\mathbf{n}_2$ 
   4. set of strategies to be considered  $\omega$ 
   5. set of cluster indices to be considered  $\mathbf{c}$ 
   6. current portfolio controls  $\mathbf{b}_t$ 
   7. past agent-controls  $\mathbf{H}_{t-1}$ 
   8. current agent-controls  $\mathbf{H}_t$ 
2: Expert_index = 0
3:
4: for  $t = t_{min}$  to  $T$  do
5:   for  $c = 1$  to  $C$  do
6:     for  $w = 1$  to  $W$  do
7:       Define  $w^{th}$  strategy as string and convert to function
8:
9:       for  $\ell = 1$  to  $L$  do
10:         $\ell_1 = \mathbf{n}_1(\ell)$ 
11:        for  $k = 1$  to  $K$  do
12:          if  $w^{th}$  strategy only has 1 parameter then
13:            break
14:          end if
15:           $k_1 = \mathbf{n}_1(k)$ 
16:          if  $k_1 > \ell_1$  then
17:            Expert_index = Expert_index + 1
18:            Call controls function to compute weights for  $w^{th}$  strategy  $\rightarrow$ 
               $\mathbf{h}_{n,t}^{c,\ell,k} = \mathbf{w}$ 
19:          else
20:            continue
21:          end if
22:        end for
23:        if Strategy has 1 parameter then
24:          Call controls function to compute weights for  $w^{th}$  strategy  $\rightarrow \mathbf{h}_{n,t}^{c,\ell} =$ 
             $\mathbf{w}$ 
25:        end if
26:      end for
27:    end for
28:  end for
29: end for

```

---

Table 14: Algorithm for the expert generating function.

**Algorithm 2** Compute *output signals***Require:**


---

```

1: 1. past expert controls  $\mathbf{h}_{t-1}^n$ 
   2. current period signals  $\mathbf{s}$ 
2: initialise combined signals:  $\text{output\_s} = \text{zeros}(\text{size}(\mathbf{s}))$ 
3: if all previous controls ( $\mathbf{h}_{t-1}^n$ ) were NaN's or zeros then
4:    $\text{output\_s} = \mathbf{s}$ 
5: else
6:   for  $i = 1$  to  $\text{length}(\mathbf{s})$  do
7:     if  $\mathbf{s}(i) == 0$  then
8:        $\text{output\_s}(i) = \text{sign}(\mathbf{h}_{t-1}^n(i))$ 
9:     else
10:       $\text{output\_s}(i) = \mathbf{s}(i)$ 
11:    end if
12:  end for
13: end if

```

---

**Table 15:** Algorithm for the controls function which transforms trading signals into portfolio controls. In the above algorithm, **output\_s** refers to *output signals* (combined signals) as discussed in Section 5.1.2

where  $x_{m,t} = \frac{P_{m,t}^c}{P_{m,t-1}^c}$  and where  $P_{m,t}^c$  is the closing price of stock  $m$  at time period  $t$ . Expert controls are generated from the price relatives for the current period  $t$  to form the expert control matrix  $\mathbf{H}_t$ . From the corresponding expert control matrix, the algorithm will then compute the expert performance  $\mathbf{S}\mathbf{h}_t$  which is the associated wealth of all  $n$  experts at time  $t$ . Denote the  $n^{\text{th}}$  expert's wealth at time  $t$  by  $Sh_t^n$ . We then form the final aggregated portfolio, denoted by  $\mathbf{b}_t$ , by aggregating the expert's wealth using the agent mixture update rules.

The relatively simplistic learning algorithm is incrementally implemented online but offline it can be parallelised across experts [6]. Given the expert controls from the Expert Generating Algorithm ( $\mathbf{H}_t$ ), the online learning algorithm is implemented by carrying out the following steps [6]:

1. **Update portfolio wealth:** Given the portfolio control  $b_{m,t-1}$  for the  $m^{\text{th}}$  asset at time  $t - 1$ , we update the portfolio wealth for the  $t^{\text{th}}$  period

$$\Delta S_t = \sum_{m=1}^M b_{m,t-1}(x_{m,t+1} - 1) + 1 \quad (42)$$

$$S_t = S_{t-1}\Delta S_t \quad (43)$$

$S_t$  represents the compounded cumulative wealth of the overall aggregate portfolio and  $\mathbf{S} = S_1, \dots, S_t$  will denote the corresponding vector of aggregate portfolio wealth's over time. Here the realised price relatives for the  $t^{\text{th}}$  period and the  $m^{\text{th}}$  asset,  $x_{m,t}$ , are combined with the portfolio controls for the previous period to obtain the realised portfolio returns for the current period  $t$ .  $\Delta S_t - 1$  is in fact the profits and losses for the current trading period  $t$ . Thus, we will use it

---

**Algorithm 3** Transform signals to controls: volatility loading method

---

**Require:**

```

1: 1. OHLCV prices up to current time  $\mathbf{X}_t$ 
   2. output signals output_s
2:
3: extract closing prices from  $\mathbf{X}_t$ :  $\mathbf{P}^c$ 
4:
5: if all output_s equal zero then
6:    $\mathbf{w} = \text{zeros}(\text{length}(\text{output\_s})+1)$ 
7:
8: else if output_s  $\geq 0$  then
9:
10:  compute standard deviations of closing prices over last 120 days for stocks where
     elements of output_s  $> 0$ :
11:
12:   $\mathbf{vol}^+ = \text{std}(\mathbf{P}^c(\text{output\_s} > 0, \text{end}-119))$ 
13:   $\mathbf{w} = [0.5 * \text{output\_s}; -0.5]$ 
14:   $\mathbf{w}(\text{output\_s} > 0) = (1/\text{sum}(\mathbf{vol}^+)) * \mathbf{w}(\text{output\_s} > 0) \cdot \mathbf{vol}^+$ 
15:
16: else if output_s  $\leq 0$  then
17:  compute standard deviations of closing prices over last 120 days for stocks where
     elements of output_s  $< 0$ :
18:
19:   $\mathbf{vol}^- = \text{std}(\mathbf{P}^c(\text{output\_s} < 0, \text{end}-119))$ 
20:   $\mathbf{w} = [0.5 \cdot \text{output\_s}; 0.5]$ 
21:   $\mathbf{w}(\text{output\_s} < 0) = (1/\text{sum}(\mathbf{vol}^-)) \cdot \mathbf{w}(\text{output\_s} < 0) \cdot \mathbf{vol}^-$ 
22:
23: else
24:  compute standard deviations of closing prices over last 120 days for stocks where
     elements of output_s  $> 0$  and where output_s  $< 0$ :
25:
26:   $\mathbf{vol}^+ = \text{std}(\mathbf{P}^c(\text{output\_s} > 0, \text{end}-119))$ 
27:   $\mathbf{vol}^- = \text{std}(\mathbf{P}^c(\text{output\_s} < 0, \text{end}-119))$ 
28:   $\mathbf{w} = [\text{output\_s}; 0]$ 
29:   $\mathbf{w}(\text{output\_s} > 0) = 0.5 \cdot \left( \frac{1}{\text{sum}(\text{abs}(\mathbf{vol}^+))} \right) \cdot \mathbf{vol}^+ \cdot \mathbf{w}(\text{output\_s} > 0)$ 
30:   $\mathbf{w}(\text{output\_s} < 0) = 0.5 \cdot \left( \frac{1}{\text{sum}(\text{abs}(\mathbf{vol}^-))} \right) \cdot \mathbf{vol}^- \cdot \mathbf{w}(\text{output\_s} < 0)$ 
31:   $\mathbf{w}(\text{end}) = \text{sum}(\mathbf{w})$ 
32: end if
33: return  $\mathbf{w}$ 

```

---

**Table 16:** Algorithm which transforms output signals into portfolio controls using the volatility loading method. The algorithm follows from Algorithm 2 and is part of the control function called by Algorithm 1. Note that  $\text{output\_s} \leq 0$  (or  $\geq$ ) implies that  $\text{output\_s}(i) \leq 0$  (or  $\geq$ ) for each  $i$

to update the algorithms overall cumulative profits and losses which is given by

$$PL_t = PL_{t-1} + \Delta S_t - 1 \quad (44)$$

2. **Update expert wealth:** The expert controls  $\mathbf{H}_t$  were determined at the end of time-period  $t - 1$  for time period  $t$  by the expert generating algorithm for  $\Omega$  experts and  $M$  objects about which the experts make expert capital allocation decisions. At the end of the  $t^{\text{th}}$  time period the performance of each expert  $n$ ,  $Sh_t^n$ , can be computed from the change in the price relatives  $x_{m,t}$  for the each of the  $M$  objects in the investment universe considered using the closing prices at the start,  $P_{m,t-1}^c$ , and the end of the  $t^{\text{th}}$  time increment,  $P_{m,t}^c$ , using the expert controls.

$$\Delta Sh_t^n = \left[ \sum_{m=1}^M \mathbf{h}_t^n (x_{m,t} - 1) \right] + 1 \quad (45)$$

$$Sh_t^n = Sh_{t-1}^n \cdot \Delta Sh_t^n \quad (46)$$

3. **Update expert mixtures:** We consider a UP [3, 5, 6] inspired expert mixture update rule as follows: the mixture of the  $n^{\text{th}}$  expert for the next time increment,  $t + 1$ , is equivalent to the accumulated expert wealth up until time  $t$  and will be used as the update feature for the next unrealised increment subsequent appropriate normalisation

$$q_{n,t+1} = Sh_t^n \quad (47)$$

4. **Renormalise expert mixtures:** As mentioned previously, we will consider experts such that the leverage is set to unity for zero-cost portfolios: 1.)  $\sum_n q_n = 0$  and 2.)  $\nu = \sum_n |q_n| = 1$ . We will not consider the long-only experts (absolute experts as in Loonat and Gebbie [6]), but only consider experts whom satisfy the prior two conditions which we will refer to as *active* experts. This in fact allows for shorting of one expert against another; then due to the nature of the mixture controls, the resulting portfolio becomes self-funding.

$$q_{n,t+1} = \frac{q_{n,t+1} - \frac{1}{N} \sum_{n=1}^{\Omega} q_{n,t+1}}{\sum_{n=1}^{\Omega} |q_{n,t+1} - \frac{1}{N} \sum_{n=1}^{\Omega} q_{n,t+1}|} \quad (48)$$

5. **Update portfolio controls:** The portfolio controls  $b_{m,t}$  are updated at the end of time period  $t$  for time period  $t + 1$  using the expert mixture controls  $q_{n,t+1}$  from the updated learning algorithm and the vector of expert controls  $\mathbf{h}_t^n$  for each expert  $n$  from the expert generating algorithms using information from time period  $t$ . We then take a weighted average over all  $n$  experts by taking the sum with respect to  $n$

$$b_{m,t+1} = \sum_n q_{n,t+1} \mathbf{h}_t^n \quad (49)$$

The strategy is to implement the portfolio controls, wait until the end of the increment, measure the features (OHLCV values), update the experts and then re-apply the learning algorithm to compute the expert mixtures and portfolio controls for the next time increment. For details on the actual algorithm, please refer to Algorithm 4. The relationships between the various components of the learning algorithm are illustrated in Section 5.2 below. Furthermore, a detailed diagram of the MATLAB learning class is illustrated in Appendix B Figure 22.

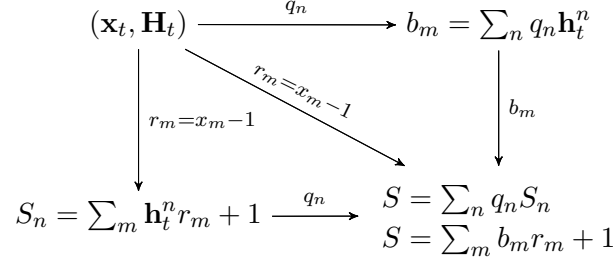


Figure 6: Relationship between the components of the Online Learning Algorithm

### 5.3 Algorithm Implementation for Intraday-Daily Trading

Intraday trading poses a whole set of new issues that need to be considered and it isn't as straightforward as simply plugging the data into the algorithm and treating it as if it is similar to daily data just sampled at more regular intervals. The main issue with this approach is the deviation in the prices at the end of day  $t - 1$  and the start of day  $t$ . Often the deviation is significant which causes returns to blow up and it will most certainly cause the technical trading strategies to generate spurious trading signals. Trading on an intraday time scale also contains inherently different dynamics to trading on a daily time scale.

We implement the learning algorithm on a combination of daily and intraday data whereby decisions made on the daily time scale are made completely independent of those made on the intraday time scale but the dynamics of the associated wealth's generated by the processes are fused together. We will refer to trading using a combination of daily and intraday data as *intraday-daily* trading. The best way to think about it is to consider the experts as trading throughout the day, making decisions based solely on intraday data while compounding their wealth, and once a trading decision is made at the final time bar, the expert makes one last trading decision on that day based on daily historic OHLCV data (the look-back periods will be based on passed trading days and not on time bars for that day). The daily trading decision can be thought of as just being the last time bar of the day where we are just using different data to make the decision. The methodology for each of the intraday and daily trading mechanisms are almost exactly as explained in Section 5.2 above however there are a couple of alterations to the algorithm. As in the daily data implementation, a given expert will begin making trading decisions as soon as there is a sufficient amount of data available to them. The idea is to begin the algorithm from day two so that there will be sufficient data to compute a return on the daily time scale. We then loop over the intraday time bars from 9:15am to 4:30pm on each given day.

To introduce some notation for intraday-daily trading, let  $\mathbf{Sh}_{t,t_I}^F$ <sup>25</sup> be the expert wealth vector for all  $n$  experts for the  $t_I^{th}$  time bar on the  $t^{th}$  day and denote by  $\mathbf{H}_{t,t_I}^F$  the associated expert control matrix. The superscript  $F$  refers to the 'fused' daily and intraday matrices. More specifically,  $\mathbf{H}_{t,t_I}^F$  will contain the 88 intraday expert controls

<sup>25</sup>Expert wealth is computed as before with  $SH_{n,t,t_I}^F = SH_{n,t,t_I-1}^F \cdot dSh_{n,t,t_I}^F$  where  $dSh_{n,t,t_I}^F$  is the  $n^{th}$  experts return at time bar  $t_I$  on day  $t$ .

---

**Algorithm 4** Online Learning Algorithm

---

**Require:**

```

1: 1. updated agent-controls  $\mathbf{H}_{t+1}$ 
   2. current price relatives  $\mathbf{x}_t$ 
   3. current portfolio controls  $\mathbf{b}_t$ 
   4. current agent-controls  $\mathbf{H}_t$ 
   5. past agent-wealth  $\mathbf{S}\mathbf{h}_{t-1}$ 
   6. past portfolio wealth  $S_{t-1}$ 
2:
3: for  $t = t_{min}$  to  $T$  do
4:   Update portfolio wealth:
5:    $S_t = S_{t-1}(\mathbf{b}_t(\mathbf{x}_t^T - 1) + 1)$ 
6:
7:   Update expert wealth's:
8:    $Sh_t^n = Sh_{t-1}^n(\mathbf{h}_t^n(\mathbf{x}_t^T - 1) + 1)$ 
9:
10:  Update expert mixtures:
11:   $q_{n,t+1} = Sh_t^n$ 
12:
13:  Renormalise the expert mixtures:
14:
15:  
$$q_{n,t+1} = \begin{cases} \sum_n q_{n,t+1} = 1, & q_{n,t+1} \geq 0 \\ \sum_n |q_{n,t+1}| = 1, & \sum_n q_{n,t+1} = 0 \end{cases}$$

16:  Update the portfolio:
17:   $\mathbf{b}_{t+1} = \sum_n q_{n,t+1} \mathbf{h}_{t+1}^n$ 
18:
19:  Leverage corrections:
20:  if  $(\nu = \sum_m |b_{m,t}|) \neq 1$  then
21:    Renormalise controls:
22:     $b_{n,t+1} = \frac{1}{\nu} b_{n,t+1}$ 
23:    Renormalise mixtures:
24:     $q_{n,t+1} = \frac{1}{\nu} q_{n,t+1}$ 
25:  end if
26: end for
27: return  $(\mathbf{b}_{t+1}, Sh_t^n, S_t, q_{n,t+1})$ 

```

---

**Table 17:** Algorithm for Online Learning.

followed by the end of day expert controls based on daily closing OHLCV data for each given day over the trading horizon. Denote  $T_I$  as the final time bar in a day (4:30pm). The  $n^{\text{th}}$  expert's wealth accumulated up until the final time bar  $T_I$ <sup>26</sup> on day  $t$ ,  $Sh_{n,t,T_I+1}^F$ , is calculated from the  $n^{\text{th}}$  column of the expert control matrix, denoted  $\mathbf{h}_{n,t,T_I}^F$ , from the previous period  $T_I$  and is computed solely from intraday data for day  $t$ . Overall portfolio controls for the final intraday trade on day  $t$  ( $\mathbf{b}_{t,T_I+1}$ ) are computed as before along with the overall portfolio wealth  $S_{t,T_I+1}$ . This position is held until the close of the day's trading when the closing prices of the  $m$  stocks  $\mathbf{P}_t^C$  are revealed. Once the closing prices are realised, the final intraday position is closed. That is, an offsetting trade of  $-\mathbf{b}_{t,T_I+1}$  is made at the prices  $\mathbf{P}_t^C$ . This profit/loss is then compounded onto  $S_{t,T_I+1}$ . Thus, no intraday positions are held overnight. The experts will then make one final trading decision based on the daily OHLCV data given that the closing price is revealed and will look-back on daily historic OHLCV data to make these decisions. The  $n^{\text{th}}$  expert's wealth  $Sh_{n,t,T_I+2}^F$  is updated using controls  $\mathbf{h}_{n,t,T_I+1}^F$ . The corresponding portfolio controls for all  $m$  stocks are computed for the daily trading decision on day  $t$  to be implemented at time  $t+1$  ( $\mathbf{b}_{t+1}$ ), the returns (price relatives) for day  $t$  are computed ( $\mathbf{r}_t = \frac{\mathbf{P}_t^C}{\mathbf{P}_{t-1}^C}$ ) and the cumulative wealth is  $S_{T_I+2} = S_{t,T_I+1} \cdot (\mathbf{b}_t \cdot (\mathbf{x}_t - 1) + 1)$  where  $S_{t,T_I+1} = S_{t,T_I} \cdot (\mathbf{b}_{T_I}(\mathbf{x}_{T_I} - 1))$  with  $\mathbf{r}_{T_I} = \frac{\mathbf{P}_{T_I}^C}{\mathbf{P}_{T_I-1}^C}$ . The daily position  $\mathbf{b}_{t+1}$  is then held until the end of the following day or possibly further into the future (until new daily data portfolio allocations are made). This completes trading for day  $t$ .

At the beginning of day  $t+1$ , the expert wealth  $Sh_{n,t+1,1}^F$ <sup>27</sup> is set back to unity. Setting experts wealth back to 1 at the beginning of the day rather than compounding on the wealth from the previous day is due to the fact that learning on intraday data between days is not possible due to the fact that conditions in the market have completely changed. Trading will begin by computing expert controls  $\mathbf{h}_{n,t+1,2}^F$  for the second time bar, however all experts will not have enough data to begin trading since the shortest look-back parameter is 4 and hence controls will all be set to zero. As the trading day proceeds, experts will begin producing non-zero controls as soon as there is sufficient data to satisfy the amount of data needed for a given look-back parameter. Something to note here is that due to the fact that the STeFI index<sup>28</sup> (risk-free asset) is only posted daily, we utilise the same STeFI value for trading throughout the day. Finally, in order to differentiate between daily OHLCV data and intraday OHLCV data, we will denote them as  $\mathbf{X}_d$  and  $\mathbf{X}_I$  respectively. The algorithm outline for intraday-daily trading is illustrated in Algorithm 5.

---

**Algorithm 5** Intraday-Daily Algorithm
 

---

**Require:**

- 1: 1. OHLCV daily prices  $\mathbf{X}_d$
2. OHLCV intraday time bars  $\mathbf{X}_I$
3. vector indicating index for start of each day **uniqueday**
- 2: initialise daily price relatives:  $\mathbf{ret}_d = \text{repeat}(1, m+1)$
- 3:
- 4: **for**  $t = 2$  to  $T$  **do**

---

<sup>26</sup> $T_I$  will always be equal to 88 as there are 88 5-minute time bars between 9:15am and 4:30pm

<sup>27</sup>We do not start the trading day at the first time bar  $t_I = 1$  since we need to compute a return which requires 2 data points.

<sup>28</sup>See Section 4 for more details on the STeFI index

```

5:
6:   initialise intraday price relatives for  $t$ -th day:
7:   ret $I$  = repeat(1,  $m + 1$ )
8:
9:   initialise expert wealths:
10:  Sh $F$ (uniqueday( $t$ )+ $t$ -1) = repeat(1)
11:
12:  repeat daily STEFI for all time bars:
13:  STEFI $I$  = repeat(STEFI( $t$ ),uniqueday( $t+1$ ) - uniqueday( $t$ ))
14:
15:  for  $t_I = \text{uniqueday}(t)+1$  to  $\text{uniqueday}(t+1)-1$  do
16:
17:    get closing prices for time bars  $t_I - 1$  and  $t_I$ :
18:     $P_{I-1}^c$  and  $P_I^c$ 
19:
20:    compute price relatives for current time bar and append to previous period
    price relatives:
21:    ret $I$  = [ret $I$ ;  $\mathbf{P}_I^c(t)/\mathbf{P}_I^c(t-1)$ ]
22:
23:    run expert generating algorithm:
24:    expert_gen( $t_0, t, \mathbf{ret}_I$ )
25:
26:    run online learning algorithm:
27:    online_learn( $t_0, t, \mathbf{ret}_I$ )
28:
29:  end for
30:
31:  get closing prices for day  $t - 1$  and day  $t$ :  $\mathbf{P}_d^c$ 
32:
33:  compute price relatives for day  $t$ :
34:  ret $d$  = [ret $d$ ;  $\mathbf{P}_d^c(t)/\mathbf{P}_d^c(t-1)$ ]
35:
36:  run expert generating algorithm:
37:  expert_gen( $t, \text{uniqueday}(t+1)$ )
38:
39:  run online learning algorithm:
40:  online_learn( $t, \text{uniqueday}(t+1)$ )
41:
42: end for
43: for  $c = 1$  to  $C$  do
44:   for  $w = 1$  to  $W$  do
45:    Define  $w^{th}$  strategy as string and convert to function
46:
47:    for  $\ell = 1$  to  $L$  do
48:      $\ell_1 = \mathbf{n}_1(\ell)$ 
49:     for  $k = 1$  to  $K$  do
50:      if  $w^{th}$  only has 1 parameter then
51:       break
52:      end if
53:       $k_1 = \mathbf{n}_1(k)$ 
54:      if  $k_1 > \ell_1$  then
55:       Expert_index = Expert_index + 1

```

---

```

56:             Call controls function to compute weights for  $w^{th}$  strategy  $\rightarrow$ 
                 $\mathbf{h}_t^{n,c,\ell,k} = \mathbf{w}$ 
57:             else
58:                 continue
59:             end if
60:         end for
61:         if Strategy has 1 parameter then
62:             Call controls function to compute weights for  $w^{th}$  strategy  $\rightarrow \mathbf{h}_t^{n,c,\ell} =$ 
                 $\mathbf{w}$ 
63:         end if
64:     end for
65: end for
66: end for

```

---

## 5.4 Trading in Volume-time

Rather than trading at the end of each day (daily) or at the end of each time bar during a trading day (intraday), it is often advantageous to trade in volume-time [50]. The proposition is to trade each time a fixed volume threshold has been reached. The idea is that, in volume-time, we are making trading decisions at times that conform to the tempo of the market and the arrival of information into the market rather than just making decisions based on clock time which tells us very little about what may have happened in the market. For example, suppose that some economic event or announcement occurs at 3:01pm. Your algorithm will only be making the next trading decision at 3:05pm (assuming 5-minute time bars) and therefore may miss out on profit making opportunities, whereas in volume-time, the algorithm will be triggered due to the excitement of traders in the market and increased trading volume and hence allows your trading system/strategy the opportunity to react accordingly [51]. One issue with a slow-paced market with very little volume over time intervals is that in such cases it may cause most technical indicators to flatten out which in turn may lead to false signals. Trading in volume-time thus becomes especially effective as we move towards smaller time scales as time becomes of very little importance in high volume markets.

## 5.5 Transaction Costs and Market Frictions

Apart from the (direct) transaction fees (commissions) charged by exchanges for the trading of stocks, there are various other costs (indirect) that need to be considered when trading such assets. Each time a stock is bought or sold there are unavoidable costs and it is imperative that a trader takes into account these costs. The other three most important components of these transaction costs besides commissions charged by exchanges are the spread<sup>29</sup>, price impact and opportunity cost [52].

To estimate indirect transaction costs (TC) for each period  $t$ , we will consider is the *square-root formula* [53]

$$\text{TC} = \text{Spread} + \sigma \cdot \sqrt{\frac{n}{\text{ADV}}} \quad (50)$$

where:

---

<sup>29</sup>Spread = best ask price minus best bid price

1. **Volatility of the returns of a stock ( $\sigma$ ):** See Section 5.5.1 below.
2. **Average daily volume of the stock (ADV):** ADV is computed using the previous 90 days trading volumes for daily trading and the previous 5 days intraday trading volume for intraday-daily trading.
3. **Number of shares traded ( $n$ ):** The number of shares traded ( $n$ ) is taken to be five tenths of a basis point of ADV for each stock per day for daily trading. The number of stocks traded for intraday-daily trading is 1% of ADV for the entire portfolio per day which is then split evenly among all trading periods and between all 15 stocks to arrive at a final value of 0.00012 (1%/85/15) of ADV per stock per trading period.
4. **Spread:** Spread is assumed to be 1bps per day (1%% /pd) for daily trading. For intraday-daily trading, we assume 12bps per day<sup>30</sup> which we then split evenly over the day to incur a cost of 0.0012/85bps per time bar.

The use of the square-root rule in practice dates back many years and is used as a pre-trade transaction cost estimate [53]. The first term in Eq. (50) can be regarded as the term representing the *slippage*<sup>31</sup> or *temporary price impact* and results due to our demand for liquidity [54]. This cost will only impact the price at which we execute our trade at and not the market price (hence subsequent transactions). The second term in Eq. (50) is the (transient) price impact which will not only affect the price of the first transaction but also the price of subsequent transactions by other traders in the market however the impact decays over time as a power-law [55]. In the following subsection, we will discuss how the volatility ( $\sigma$ ) is estimated for the square-root formula. Technically,  $\sigma$ ,  $n$  and ADV in Eq. (50) should each be defined by a vector representing the volatilities, number of stocks traded and ADV of each stock in the portfolio respectively however for the sake of generality we will write it as a constant, thus representing the volatility for a single portfolio stock.

In addition to the indirect costs associated with slippage and price impact as accounted for by the square-root formula, we include direct costs such as the borrowing of trading capital, the cost of regulatory capital and the various fees associated with trading on the JSE [6]. Such costs will also account for small fees incurred in incidences where short-selling has taken place. For the daily data implementation, we assume a total direct cost of 4bps per day. This assumption is purely made to approximately match the total daily transaction cost assumption made by Loonat and Gebbie [6] which is used an approximate to real daily trading costs. For the intraday-daily implementation a total direct cost of 70bps per day is assumed (following Loonat and Gebbie [6]) which we then split evenly over each days' active trading periods (85 time bars since first expert only starts trading after the 5<sup>th</sup> time bar) to get a cost of 70bps/85 per period.

For daily trading, we recover an average daily transaction cost of roughly 9.38bps which is approximately the same as the assumed 10bps by Loonat and Gebbie [6]. Loonat and Gebbie argue that for intraday trading, it is difficult to avoid a direct and indirect cost of about 50-80bps per day in each case leaving a conservative estimate of total costs to be approximately 160bps per day. We realise an overall average cost per period of 1.5bps while the average cost per day assuming we trade for 85 periods throughout each day is roughly 130bps (85\*1.5) for intraday-daily trading.

<sup>30</sup>We follow the conservative approach taken by Loonat and Gebbie [6] as opposed to the moderate approach where 9bps per day is assumed

<sup>31</sup>Slippage is often calculated as the difference between the price at which an order for a stock is placed and the price at which the trade is executed

### 5.5.1 Volatility Estimation for Transaction Costs

In this section we will discuss different methods for calculating the estimates for volatility ( $\sigma$ ) for daily and intraday data in the square-root formula (Eq. (50)).

#### 5.5.1.1 Daily Data Estimation

The volatility of daily prices at each day  $t$  is taken to be the standard deviation of closing prices over the last 90 days. If 90 days have not passed, then the standard deviation will be taken over the number of days available so far.

#### 5.5.1.2 Intraday Data Estimation

The volatility for each intraday time bar  $t_I$  on day  $t$  is dependent on the time of day. For the first 15 time bars, the volatility is taken to be a forecast of a GARCH(1,1) model which has been fitted on the last 60 returns of the previous day  $t - 1$ . The reason for this choice is that the market is very volatile during the opening hour as well as the fact that there will be relatively few data points to utilise when computing the volatility. The rest of the days volatility estimates are computed using the Realized Volatility (RV) method [56]. RV is one of the more popular methods for estimating volatility of high-frequency returns<sup>32</sup> computed from tick data. The measure estimates volatility by summing up intraday squared returns at short intervals (eg. 5 minutes). Andersen *et al.* [56] propose this estimate for volatility at higher frequencies and derive it by showing that RV is an approximate of quadratic variation under the assumption that log returns are a continuous time stochastic process with zero mean and no jumps. The idea is to show that the RV converges to the continuous time volatility (quadratic variation) [57], which we will now demonstrate.

Assume that the instantaneous returns of observed log stock prices ( $p_t$ ) with unobservable latent volatility ( $\sigma_t$ ) scaled continuously through time by a standard Wiener process ( $dW_t$ ) can be generated by the continuous time martingale [57]

$$dp_t = \sigma_t dW_t \quad (51)$$

Then it follows that the conditional variance of the single period returns,  $r_{t+1} = p_{t+1} - p_t$  is given by

$$\sigma_t^2 = \int_t^{t+1} \sigma_s^2 ds \quad (52)$$

Eq. (52) is also known as the *integrated volatility* for the period  $t$  to  $t + 1$ .

Suppose the sampling frequency of the tick data into regularly spaced time intervals is denoted by  $f$  such that between period  $t - 1$  and  $t$  there are  $f$  continuously compounded returns. Then

$$r_{t+1/f} = p_{t+1/f} - p_t \quad (53)$$

Hence, we get the *Realised Volatility* (RV) based on  $f$  intraday returns between

---

<sup>32</sup>Most commonly refers to returns over intervals shorter than one day. This could be minutes, seconds or even milliseconds.

periods  $t + 1$  and  $t$  as

$$RV_{t+1} = \sum_{i=1}^f r_{t+i/f}^2 \quad (54)$$

The argument here is that provided we sample at frequent enough time steps ( $f$ ), the volatility can be observed theoretically from the sample path of the return process and hence [57, 58]

$$\lim_{f \rightarrow \infty} \left( \int_t^{t+1} \sigma_s^2 ds - \sum_{i=1}^f r_{t+i/f}^2 \right) = 0 \quad (55)$$

which says that the RV of a sequence of returns asymptotically approaches the integrated volatility and hence the RV is a reasonable estimate of current volatility levels.

## 6 Testing for Statistical Arbitrage

To test the overall trading strategy for statistical arbitrage, we implement a novel statistical test originally proposed by [22] and later modified by [31] by applying it to the overall strategy's profit and losses  $\mathbf{PL}$ . The idea is to axiomatically define the conditions under which a statistical arbitrage exists and assume a parametric model for incremental trading profits in order to form a null hypothesis derived from the union of several sub-hypotheses which are formulated to facilitate empirical tests of statistical arbitrage. The modified test, proposed by [31], called the Min- $t$  test, is derived from a set of restrictions imposed on the parameters defined by the statistical arbitrage null hypothesis and is applied to a given trading strategy to test for statistical arbitrage. The Min- $t$  statistic is argued to provide a much more efficient and powerful statistical test compared to the Bonferroni inequality used in [22]. The lack of statistical power is reduced when the number of sub-hypotheses increases and as a result, the Bonferroni approach is unable to reject an incorrect null hypothesis leading to a large Type II error.

To set the scene and introduce the concept of a statistical arbitrage, suppose that in some economy, a stock (portfolio)<sup>33</sup>  $s_t$  and a money market account  $B_t$ <sup>34</sup> are traded. Let the stochastic process  $(x(t), y(t) : t \geq 0)$  represent a zero initial cost trading strategy that trades  $x(t)$  units of some portfolio  $s_t$  and  $y(t)$  units of the money market account at a given time  $t$ . Denote the cumulative trading profits at time  $t$  by  $V_t$ . Let the time series of discounted cumulative trading profits generated by the trading strategy be denoted by  $\nu(t_1), \nu(t_2), \dots, \nu(t_T)$  where  $\nu(t_i) = \frac{V_{t_i}}{B_{t_i}}$  for each  $i = 1, \dots, T$ . Denote the increments of the discounted cumulative profits at each time  $i$  by  $\Delta\nu_i = \nu(t_i) - \nu(t_{i-1})$ . Then, a statistical arbitrage is defined as:

**Definition 1** (Statistical Arbitrage [22, 31]). A *statistical arbitrage* is a zero-cost, self-financing trading strategy  $(x(t) : t \geq 0)$  with cumulative discounted trading profits  $\nu(t)$  such that

1.  $\nu(0) = 0$

<sup>33</sup>In our study, we will be considering a portfolio

<sup>34</sup>The money market account is initialised at one unit of a currency *i.e.*  $B_0 = 1$ .

2.  $\lim_{t \rightarrow \infty} \mathbb{E}^P[\nu(t)] > 0$
3.  $\lim_{t \rightarrow \infty} P[\nu(t) < 0] = 0$
4.  $\lim_{t \rightarrow \infty} \text{Var}[\Delta\nu(t) | \Delta\nu(t) < 0] = 0$

In other words, a statistical arbitrage is a trading strategy that 1) has zero initial cost, 2) in the limit has positive expected discounted cumulative profits, 3) in the limit has a probability of loss that converges to zero and 4) variance of negative incremental trading profits (losses) converge to zero in the limit. It is clear that deterministic arbitrage stemming from traditional financial mathematics is in fact a special case of statistical arbitrage [59].

In order to test for statistical arbitrage, assume that the incremental discounted trading profits evolve over time according to the process

$$\Delta\nu_i = \mu i^\theta + \sigma i^\lambda z_i \quad (56)$$

where  $i = 1, \dots, T$ . There are two cases to consider for the innovations: 1)  $z_i$  i.i.d  $N(0,1)$  normal uncorrelated random variables satisfying  $z_0 = 0$  or 2)  $z_i$  follows an MA(1) process given by:

$$z_i = \epsilon_i + \phi \epsilon_{i-1} \quad (57)$$

in which case the innovations are non-normal and correlated. Here,  $\epsilon_i$  are i.i.d.  $N(0,1)$  normal uncorrelated random variables. It is also assumed that  $\Delta\nu_0 = 0$  and in case of our algorithm  $\nu_{t_{min}} = 0$ . We will refer the first model (normal uncorrelated innovations) as the unconstrained mean (UM) model and the second model (non-normal and correlated innovations) as the unconstrained mean with correlation (UMC) model. Furthermore, we refer to the corresponding models with  $\theta = 0$  as the constrained mean (CM) and constrained mean with correlation (CMC) respectively which assume constant incremental profits over time and hence have an incremental profit process given by:

$$\Delta\nu_i = \mu + \sigma i^\lambda z_i \quad (58)$$

The discounted cumulative trading profits for the UM model at terminal time  $T$  discounted back to the initial time which are generated by a trading strategy are given by

$$\nu(T) = \sum_{i=1}^T \Delta\nu_i \sim N\left(\mu \sum_{i=1}^T i^\theta, \sigma^2 \sum_{i=1}^T i^{2\lambda}\right) \quad (59)$$

From Eq. (59), it is straightforward to show that the log-likelihood function for the discounted incremental trading profits is given by

$$\begin{aligned} \ell(\mu, \sigma^2, \lambda, \theta | \Delta\nu) &= \log L(\mu, \sigma^2, \lambda, \theta | \Delta\nu) \\ &= -\frac{1}{2} \sum_{i=1}^T \log(\sigma^2 i^{2\lambda}) \\ &\quad - \frac{1}{2\sigma^2} \sum_{i=1}^T \frac{1}{i^{2\lambda}} (\Delta\nu_i - \mu i^\theta)^2 \end{aligned} \quad (60)$$

The probability of a trading strategy generating a loss after  $n$  periods is as follows [31]

$$\Pr\{\text{Loss after } n \text{ periods}\} = \Phi\left(\frac{-\mu \sum_{i=1}^n i^\theta}{\sigma(1+\phi)\sqrt{\sum_{i=1}^n i^{2\lambda}}}\right) \quad (61)$$

where  $\Phi(\cdot)$  denotes the cumulative standard normal distribution function. For the CM model, Eq. (61) is easily adjusted by setting  $\phi$  and  $\theta$  equal to zero. This probability converges to zero at a rate that is faster than exponential.

As mentioned previously, to facilitate empirical tests of statistical arbitrage under Definition 1, a set of sub-hypotheses are formulated to impose a set of restrictions on the parameters of the underlying process driving discounted cumulative incremental trading profits and are as follows:

**Proposition 1** (UM Model Hypothesis [31]). *Under the four axioms defined in Definition 1, a trading strategy generates a statistical arbitrage under the UM model if the discounted incremental trading profits satisfy the intersection of the following four sub-hypotheses jointly:*

1.  $H_1 : \mu > 0$
2.  $H_2 : -\lambda > 0$  or  $\theta - \lambda > 0$
3.  $H_3 : \theta - \lambda + \frac{1}{2} > 0$
4.  $H_4 : \theta + 1 > 0$

An intersection of the above sub-hypotheses defines a statistical arbitrage and as by De Morgan's Laws<sup>35</sup>, the null hypothesis of no statistical arbitrage is defined by a union of the sub-hypotheses. Hence, the no statistical arbitrage null hypothesis is the set of sub-hypotheses which are taken to be the complement of each of the sub-hypotheses in Proposition 1:

**Proposition 2** (UM Model Alternative Hypothesis [22, 31]). *Under the four axioms defined in Definition 1, a trading strategy does not generate a statistical arbitrage if the discounted incremental trading profits satisfy any one of the following four sub-hypotheses:*

1.  $H_1 : \mu \leq 0$
2.  $H_2 : -\lambda \leq 0$  or  $\theta - \lambda \leq 0$
3.  $H_3 : \theta - \lambda + \frac{1}{2} \leq 0$
4.  $H_4 : \theta + 1 \leq 0$

The null hypothesis is not rejected provided that a single sub-hypothesis holds. The Min- $t$  test is then used to test the above null hypothesis of no statistical arbitrage by considering each sub-hypothesis separately using the t-statistics  $t(\hat{\mu})$ ,  $t(-\hat{\lambda})$ ,  $t(\hat{\theta} - \hat{\lambda})$ ,  $t(\hat{\theta} - \hat{\lambda} + 0.5)$ , and  $t(\hat{\theta} + 1)$  where the hats denote the Maximum Likelihood Estimates (MLE) of the parameters. The Min- $t$  statistic is defined as [31]

$$\text{Min-}t = \text{Min}\{t(\hat{\mu}), t(\hat{\theta} - \hat{\lambda}), t(\hat{\theta} - \hat{\lambda} + 0.5), \text{Max}[t(-\hat{\lambda}), t(\hat{\theta} + 1)]\} \quad (62)$$

---

<sup>35</sup>This states that the complement of the intersection of sets is the same as the union of their complements.

The intuition is that the Min- $t$  statistic returns the smallest test statistic which is the sub-hypothesis which is closest to being accepted. The no statistical arbitrage null is then rejected if  $\text{Min-}t > t_c$  where  $t_c$  depends on the significance level of the test which we will refer to as  $\alpha$ . Since the probability of rejecting cannot exceed the significance level  $\alpha$ , we have the following condition for the probability of rejecting the null at the  $\alpha$  significance level

$$\Pr\{\text{Min-}t > t_c | \mu, \lambda, \theta, \sigma\} \leq \alpha \quad (63)$$

What remains is for us to compute the critical value  $t_c$ . We will implement a Monte Carlo simulation procedure to compute  $t_c$  which we describe in more detail in Section 6.1 step 5 below.

## 6.1 Outline of the Statistical Arbitrage Test Procedure

The steps involved in testing for statistical arbitrage are outlined below:

1. **Trading increments  $\Delta\nu_i$ :** From the vector of cumulative trading profits and losses, compute the increments  $(\Delta\nu_1, \dots, \Delta\nu_T)$  where  $\Delta\nu_i = \nu(t_i) - \nu(t_{i-1})$ .
2. **Perform MLE:** Compute the likelihood function as given in Eq. (60) and maximise it to find the estimates of the four parameters, namely,  $\hat{\mu}, \hat{\sigma}, \hat{\theta}$  and  $\hat{\lambda}$ . The log-likelihood function will obviously be adjusted depending on whether the CM ( $\theta = 0$ ) or UM test is implemented. We will only consider the CM test in this study. Since MATLAB's built-in constrained optimization algorithm<sup>36</sup> only performs minimization, we minimize the negative of the log-likelihood function *i.e.* maximise the log-likelihood.
3. **Standard errors:** From the estimated parameters in the MLE step above, compute the negative Hessian estimated at the MLE estimates which is indeed the Fisher Information (FI) matrix denoted by  $\mathbf{I}(\Theta)$ . In order to compute the Hessian, the analytical partial derivatives are derived from Eq. (60). Standard errors are then taken to be the square roots of the diagonal elements of the inverse of  $\mathbf{I}(\Theta)$  since the inverse of the Fisher information matrix is an asymptotic estimator of the covariance matrix.
4. **Min- $t$  statistic:** Compute the t-statistics for each of the sub-hypotheses which are given by  $t(\hat{\mu}), t(-\hat{\lambda}), t(\hat{\theta} - \hat{\lambda}), t(\hat{\theta} - \hat{\lambda} + 0.5)$ , and  $t(\hat{\theta} + 1)$  and hence the resulting Min- $t$  statistic given by Eq. (62). Obviously,  $t(\hat{\theta} - \hat{\lambda}), t(\hat{\theta} - \hat{\lambda} + 0.5)$  and  $t(\hat{\theta} + 1)$  will not need to be considered for the CM test.
5. **Critical values:** Compute the critical value at the  $\alpha$  significance level using the Monte Carlo procedure (uncorrelated normal errors) and Bootstrapping (correlated non-normal errors)

(a) **CM model**

First, simulate 5000 different profit process using Eq. (58) with  $(\mu, \lambda, \sigma^2) = (0, 0, 0.01)$ <sup>37</sup>. For each of the 5000 profit processes, perform MLE to get estimated parameters, the associated t-statistics and finally the Min- $t$  statistics.  $t_c$  is taken to be the  $1-\alpha$  quantile of the resulting distribution of Min- $t$  values.

<sup>36</sup>Here we are referring to MATLAB's *fmincon* function

<sup>37</sup> $t_c$  is maximised when  $\mu$  and  $\lambda$  are zero.  $\sigma^2$  is set equal to 0.01 to approximate the empirical MLE estimate [31].

6. **P-values:** Compute the empirical probability of rejecting the null hypothesis at the  $\alpha$  significance level using Eq. (63) by utilising the critical value from the previous step and the simulated Min- $t$  statistics.
7. **n-Period Probability of Loss:** Compute the probability of loss after  $n$  periods for each  $n = 1, \dots, T$  and observe the number of trading periods it takes for the probability of loss to converge to zero (or below 5% as in the literature). This is done by computing the MLE estimates for the vector  $(\Delta\nu_1, \Delta\nu_2, \dots, \Delta\nu_n)$  for each given  $n$  and substituting these estimates into Eq. (61).

There were various issues when implementing the UM statistical arbitrage test on the overall strategies profits and losses. In the original implementation,  $R$ 's *optim* function with the L-BFGS-B method which allows box for constraints whereby that is each variable can be given a lower and/or upper bound. The only reason that constrained optimisation must be used is due to the fact that the variance must be non-negative. All other parameters are free to vary. It was apparent that the optimisation algorithm was not able to find the maximum (minimum) of the log-likelihood function as the score equations were non-zero. Another major issue was the fact that the inverse FI matrix, required to compute the standard errors of the ML estimates, had negative diagonal elements which lead to complex-valued standard errors. The first trial solution to this problem was to replace the numerical Hessian (negative FI matrix) computed by the *optimHess* function by the analytically derived Hessian. This did not seem to alleviate the problem and we began looking at other methods to estimate the ML parameters. It was decided that a Markov Chain Monte Carlo method may be best suited for this, as there was a strong possibility that the probability distribution of the underlying process was bimodal. Recoding everything in MATLAB and using the in-built constrained optimisation function *fmincon* solved the aforementioned issues with regards to the CM test but not the UM test. There were also a variety of issues with the optimization involved in the Monte Carlo simulation used to produce the critical values and hence it was decided that it was sufficient to remain with the CM implementation for our purposes.

## 7 Probability of Back-test Overfitting

When designing an automated trading system (algorithm), it is always recommended that the system be simulated on historical data in order to test their performance. This is known as a back-test and is a process by which the series of profits and losses that such strategy would have generated had that algorithm been run over that time period is computed [18].

When measuring the performance of a back-tested strategy, there are two different readings: in-sample (IS) performance and out-of-sample (OOS) performance. IS performance is simulated over a sample of data used in the design of the trading strategy which can be referred to as the “training set”. OOS performance is simulated of the sample of data used to test the trading strategy which is also known as the “testing set”. Bailey *et al.* [18] heavily criticise recent studies which claim to have designed profitable investment or trading strategies since many of these studies are only based on IS statistics without evaluating OOS performance. This may lead to a phenomenon called *overfitting* which occurs when a trading model targets particular observations rather than a general structure [18]. The authors state that it is relatively simple to overfit a trading strategy so that it has good IS performance, however, for a back-test

to be realistic the IS and OOS performance must be consistent with one another.

Given that it is imperative that we assess whether the proposed strategy is able to generalise well on OOS (unseen) data, a nonparametric methodology is implemented to estimate the extent to which the algorithm is overfitting IS data. This is what will be referred to as the estimating the probability of back-test overfitting (PBO) and the procedure to compute such estimates is called combinatorially symmetric cross-validation (CSCV) [19]. Typically, an investor/researcher will run many ( $N$ ) trial back-tests to select the parameter combinations which optimise the performance of the algorithm (usually based on some performance evaluation criterion such as the Sharpe Ratio). The idea is to perform CSCV on the matrix of performance series of length  $T_{BL}$ <sup>38</sup> for  $N$  separate trial simulations of the algorithm.

Here we must be clear that from here on when we refer to IS, we do not mean the “training set” per say, during which the moving average look-back parameters were calculate for example. Rather, we refer to IS as being the subset of observations utilised in selecting the optimal strategy from the  $N$  back-test trials.

In the case of the algorithm proposed in this study, since the large set of trialled parameters form the basis of the learning algorithm in the form of the experts, we cannot observe the effect of different parameters settings on the overall strategy as these are already built into the underlying algorithm. Rather, we will run  $N$  trial back-test simulations on independent subsets of historical data to get an idea of how the algorithm performs on unseen data. We can then implement the CSCV procedure on the matrix of profits and losses resulting from the trials to recover a PBO estimate. Essentially there is no training of parameters taking place in our model as all parameter combinations are considered and the weights of the performance weighted average of the expert’s strategies associated with the different parameters are “learnt”.

More specifically, we choose a back-test length  $T_{BL}$  for each subset and split the entire history of OHLCV data into subsets of this length. The learning algorithm is then implemented on each subset to produce  $N = \lfloor T/T_{BL} \rfloor$  profit and loss time series. Note that the subsets will be completely independent from one another as there is no overlapping of the data that each separate simulation is run on. A matrix  $M$  is then constructed by taking the profits and losses over time for each of the back-test simulations. This matrix will form the first step of the CSCV procedure which is explained in detail in Section 7.2 but first, in the following subsection, we will introduce the theory required to define back-test overfitting, and hence, the probability of back-test overfitting.

## 7.1 Back-test Overfitting Framework

Consider the triple  $(\mathcal{T}, \mathcal{F}, \mathbb{P})$  with  $\mathcal{T}$  representing a sample space of pairs of IS and OOS realisations, and  $\mathcal{F}$  and  $\mathbb{P}$  an appropriate filtration and probability measure respectively. Given a time series of profits and losses for each of  $N$  trial back-tests, apply a performance measure (such as the Sharpe ratio) to each time series. Let  $\mathbf{R} = (R_1, R_2, \dots, R_N)$  and  $\overline{\mathbf{R}} = (\overline{R}_1, \overline{R}_2, \dots, \overline{R}_N)$  be random vectors on  $(\mathcal{T}, \mathcal{F}, \mathbb{P})$  representing the IS and OOS performance of the  $N$  back-tests for a fixed performance measure, respectively. Denote by  $\mathbf{R}^C$  and  $\overline{\mathbf{R}}^C$  the performances for a pair of IS and

---

<sup>38</sup>The subscript BL stands for back-test length

OOS sample subsets of  $\mathbf{R}$  and  $\overline{\mathbf{R}}$  respectively.

Let  $\Omega^{39}$  represent the ranking space of  $N!$  permutations of the set  $(1, \dots, N)$  which ranks the  $N$  back-test trials. Denote the rankings of the vectors  $\mathbf{R}$  and  $\overline{\mathbf{R}}$  by random vectors  $\mathbf{r}$  and  $\overline{\mathbf{r}}$  respectively.<sup>40</sup> Then, given a subset  $\Omega_n^* = \{\mathbf{f} \in \Omega | f_n = N\}$  of  $\Omega$  where  $\mathbf{f} = (f_1, \dots, f_n)$ , which consists of a vector of back-test trials' performance rankings  $\mathbf{f}$  such that the  $n^{\text{th}}$  back-test trial has the highest ranking  $f_n = N$ , back-test overfitting is then defined as:

**Definition 2** (Back-test Overfitting [19]). A back-test trial process overfits if a trial with optimal IS performance has an expected ranking which lies below the median OOS. Mathematically, back-test overfitting is defined as

$$\sum_{n=1}^N \mathbb{E}[\overline{r}_n | \mathbf{r} \in \Omega_n^*] \cdot \mathbb{P}[\mathbf{r} \in \Omega_n^*] \leq N/2 \quad (64)$$

The associated probability of back-test overfitting is stated in the following definition:

**Definition 3** (Probability of Back-test Overfitting [19]). A back-test trial with optimal performance IS is not necessarily optimal OOS and has a non-null probability that the performance IS ranks below the median OOS performance. This is what is defined as probability of back-test overfitting (PBO). More specifically,

$$\text{PBO} = \sum_{n=1}^N \mathbb{P}[\overline{r}_n \leq N/2 | \mathbf{r} \in \Omega_n^*] \cdot \mathbb{P}[\mathbf{r} \in \Omega_n^*] \quad (65)$$

Hence, a back-test process overfits if the expected OOS performance of trials selected IS is smaller than the median OOS rank of all trials. Here, it must be noted that this definition holds irrespective of the model calibration parameters of the underlying strategies which, as mentioned previously, refers the expert parameters in the case of the algorithm proposed in this paper. Thus, overfitting is defined in the sense of the trial back-test process and not the model calibration process which leads to a model-free and nonparametric estimate of PBO [19].

## 7.2 CSCV Procedure

We follow Bailey *et al.* [19] in outlining the CSCV procedure while to estimate the PBO. The only step that differs between the implementation in this study and the proposed methodology is the formation of the matrix of performance series.

1. **Compute the matrix of performance series  $\mathbf{M}$ :** Form the matrix  $M$  (of dimension  $T_{\text{BL}} \times N$ ) as the profits and losses of subsets of length  $T_{\text{BL}}$  taken from the entire history of OHLCV data to produce  $N = \lfloor T/T_{\text{BL}} \rfloor$  trial simulations.
2. **Partition the rows of  $\mathbf{M}$ :** Partition the rows of  $\mathbf{M}$  into an even number  $S$  of disjoint submatrices where each submatrix  $\mathbf{M}_s$  for  $s = 1, \dots, S$  is of equal dimension  $(\frac{T_{\text{BL}}}{S} \times N)$ . We make the choice of setting  $S = 8$  in our implementation.

<sup>39</sup>This bold and capitalized omega ( $\Omega$ ) is different from the one used to denote the total number of experts as referred to previously in the study which is not bold ( $\Omega$ )

<sup>40</sup>For example, given  $\mathbf{R} = (1.2, 0.98, 0.7)$  then  $\mathbf{r} = (3, 2, 1)$

3. **Construct combinations of submatrices of  $\mathbf{M}_s$ :** Construct all combinations  $C_S^{S/2}$  of length  $S/2$  of the sequence  $1, \dots, S$ . This comprises of different combinations of  $S/2$  submatrices  $\mathbf{M}_s$ . The total number of combinations is [19]

$$\binom{S}{S/2} = \prod_{i=0}^{S/2-1} \frac{S-i}{S/2-i} \quad (66)$$

4. **Compare IS and OOS performance for each combination  $C_S^{S/2}(i)$  for  $i = 1, \dots, S$ :**

- (a) **Form training set  $J^{\text{train}}$ :** Join the  $S/2$  submatrices  $\mathbf{M}_s$  that constitute  $C_S^{S/2}(i)$  in their original order to get the training matrix  $J_{(T/S)(S/2) \times N}^{\text{train}} = J_{(T/2) \times N}^{\text{train}}$
- (b) **Form testing set  $J^{\text{test}}$ :** Set the test set as the complement of the training set in  $M$ :  $J^{\text{test}} = M \setminus J^{\text{train}}$ .<sup>41</sup> Thus,  $J_{(T/2) \times M}^{\text{test}}$  is a matrix of rows of  $\mathbf{M}$  not in  $J^{\text{train}}$ .
- (c) **Compute IS performance statistics and rank them:** Compute the performance statistic (we use the Sharpe Ratio) for each column of  $J^{\text{train}}$  to form the vector  $R^C$  of dimension  $N$ . Rank the components of  $\mathbf{R}^C$  to get the IS ranking of the  $N$  back-test trials denoted  $\mathbf{r}^C$ .
- (d) **Compute OOS performance statistics and rank them:** Repeat 4c on  $J^{\text{test}}$  to construct OOS performance statistics  $\bar{\mathbf{R}}^C$  and the corresponding OOS rankings  $\bar{\mathbf{r}}^C$ .
- (e) **Determine best performing trial IS:** Determine the element  $n^*$  in the rankings vector IS such that  $\mathbf{r}^C \in \Omega_{n^*}^*$ . That is, find  $n^*$  such that  $\mathbf{r}_n^C \leq \mathbf{r}_{n^*}^C \forall n = 1, \dots, N \Rightarrow n^* = \arg \max_n \{\mathbf{r}_n^C\}$ .
- (f) **Determine relative rank of the OOS performance associated with the chosen IS trial:** Denote the relative rank of OOS element corresponding to the performance of the best trial chosen IS sample  $n^*$  by  $\bar{\omega}_C$  where

$$\bar{\omega}_C = \frac{\bar{\mathbf{r}}_{n^*}^C}{N+1} \quad (67)$$

In other words, we want to determine the relative rank of  $\bar{\mathbf{r}}_{n^*}^C$  within  $\bar{\mathbf{r}}^C$ .

- (g) **Determine constituency between IS and OOS performances:** Define the logit ranks

$$\lambda_C = \ln\left(\frac{\bar{\omega}_C}{1-\bar{\omega}_C}\right) \quad (68)$$

Larger values of  $\lambda_C$  will imply consistency between IS and OOS performance, and hence low levels of back-test overfitting. In addition,  $\lambda_C = 0$  when  $\bar{\mathbf{r}}_{n^*}^C$  is identical to the median of  $\bar{\mathbf{r}}^C$ .

5. **Construct the distribution of logit ranks OOS:** Plot the histogram of logits  $\lambda_C$  for  $C \in C_S^{S/2}(i) \forall i = 1, \dots, S$
6. **Compute the PBO estimate:** The empirical PBO estimate is the relative

<sup>41</sup>  $A \setminus B$  denotes the complement of  $B$  in  $A$

number of negative logit ranks

$$\text{PBO} = \frac{1}{\text{no. of logits}} \sum_C \lambda_C \cdot \mathbf{I}_{\{\lambda_C \leq 0\}} \quad (69)$$

This is the discrete version of the theoretical continuous cumulative probability distribution  $\int_{-\infty}^0 f(\lambda) d\lambda$  where  $f(\lambda)$  is the relative frequency at which  $\lambda$  occurs across all combinations  $C \in C_S^{S/2}(i) \forall i = 1, \dots, S$ . Hence, PBO is simply the cumulative probability that the out-of-sample relative rank of the best back-test trial chosen in-sample performs worse than the median out-of-sample rank.

## 8 Results and Analysis

### 8.1 Daily Data

In this section, we implement the various algorithms described above in order to plot a series of graphs for daily JSE Top 40 data as discussed in Section 4 above. We will plot five different graphs: first is the overall portfolio wealth over time which corresponds to  $S_t$  as described above, second, the cumulative profit and losses over time  $\text{PL}_t$ , third, the relative population wealth of experts corresponds to the wealth accumulated over time by each of the experts competing for wealth in the algorithm  $\mathbf{Sh}_t$  and finally, the relative population wealth of the strategies which takes the mean over all experts for each given trading strategy to generate an accumulated wealth path for each technical trading rule.

For the purpose of testing the learning algorithm, we will identify the 15 most liquid stocks over one year prior to the start of active trading. The stocks ranked from most to least liquid are as follows: FSRJ.J, OMLJ.J, CFRJ.J, MTNJ.J, SLMJ.J, NTCJ.J, BILJ.J, SBKJ.J, WHLJ.J, AGLJ.J, SOLJ.J, GRTJ.J, INPJ.J, MNDJ.J and RMHJ.J.

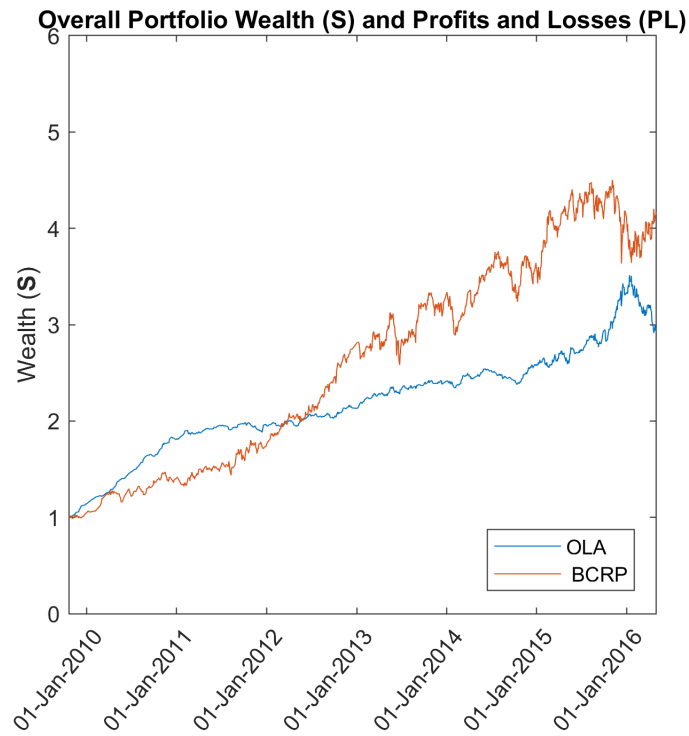
#### 8.1.1 No Transaction Costs

Barring transaction costs, it's clear that the portfolio makes favourable cumulative returns on equity over the 6-year period as is evident in Figure 7(a). The performance of the online learning algorithm (blue) is similar to that of the benchmark BCRP strategy (orange) which is promising as the original literature proves that the algorithm should track such a benchmark in the long-run. Figure 7(b) shows that the overall strategy provides consistent positive trading profits over the entire trading horizon. Figure 8(a) shows the expert wealth for all  $\Omega$  experts and Figure 8(b) illustrates the corresponding mean wealth enumerated over all expert's wealth's for each strategy  $\omega(i)$ . These figures show that on average, the underlying experts perform fairly poorly compared to the overall strategy however there is evidence that some experts make satisfactory returns over the period.

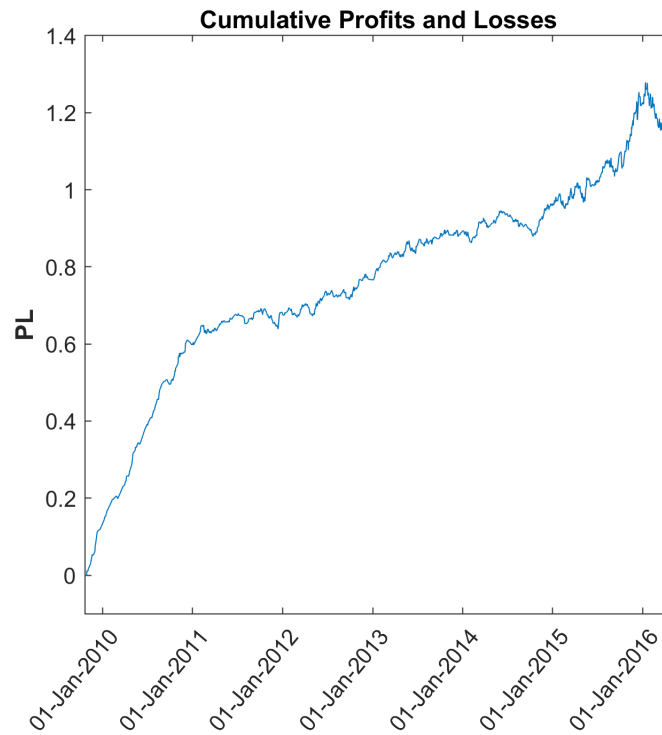
Table 18 and Table 19 provide the group summary statistics of the terminal wealth's of experts and of the expert's profits and losses over the entire trading horizon respectively where experts are grouped based on their underlying strategy  $\omega(i)$ . The online Z-Anticor<sup>42</sup> algorithm produces the best expert (maximum terminal wealth)

---

<sup>42</sup>Please refer to Section 3 for a detailed description of the various trading rules mentioned in the tables

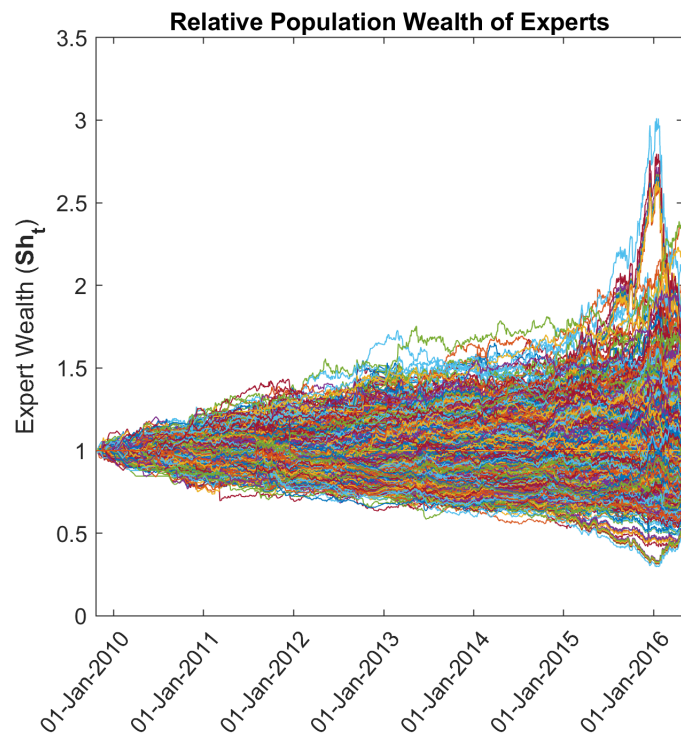


(a)

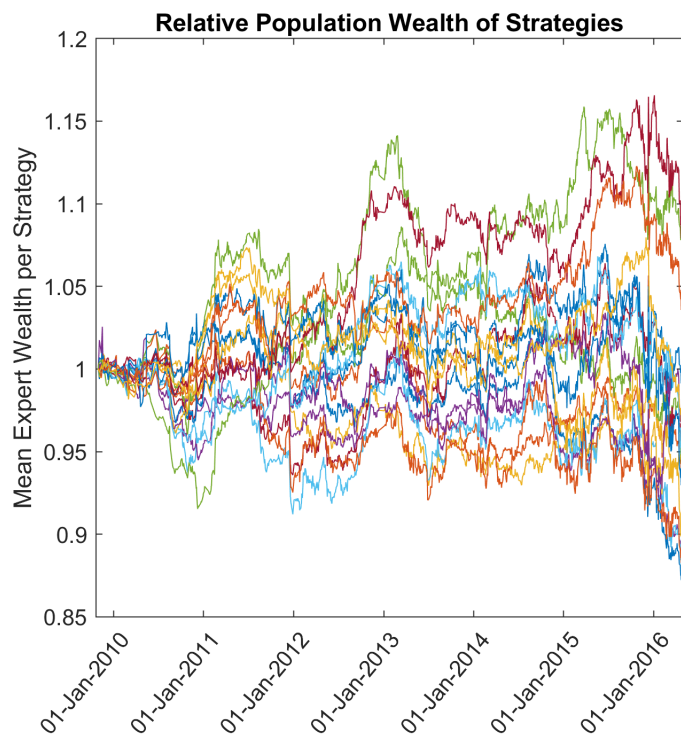


(b)

**Figure 7:** Figure 7(a) illustrates the overall cumulative portfolio wealth ( $S$ ) of the online learning algorithm (OLA - blue) against the benchmark BCRP strategy (orange) and Figure 7(b) illustrates the associated profits and losses ( $PL$ ) for daily trading without transaction costs.



(a)



(b)

**Figure 8:** Figure 8(a) illustrates the expert wealth ( $Sh$ ) for all  $\Omega$  experts for daily data with no transaction costs. Figure 8(b) illustrates the mean expert wealth of all experts for each trading strategy ( $\omega$ ) for daily data with no transaction costs.

followed closely by the slow stochastic rule while Z-Anticor also produces experts with the greatest mean terminal wealth over all experts (column 2). In addition, Z-Anticor produces expert's with wealth's that vary the most (highest standard deviation). Williams %R produces the worst expert by quite a long way (minimum terminal wealth). The trading rule with the worst mean terminal wealth and worst mean ranking are SAR and slow stochastic respectively. With regards to the expert's profits and losses (Table 19), the momentum rule (MOM) produces the expert with the greatest profit in a single period. SAR followed by Anti-Z-BCRP produce the worst and second worst mean profit/loss per trading period respectively whereas Z-Anticor and Z-BCRP achieve the best mean profit/loss per trading period.

Strategy	Mean (mean rank)	St. Dev.	Min	Max
EMA X-over	0.8739 (673.6343)	0.1767	0.5216	1.4493
Ichimoku Kijun Sen	0.9508 (623.3194)	0.2313	0.5424	1.5427
MACD	0.9504 (657.7639)	0.1750	0.5601	1.6065
Moving Ave X-over	0.8895 (632.6944)	0.1930	0.5206	1.4505
ACC	1.0994 (736.5833)	0.3131	0.5283	1.9921
BOLL	1.0499 (569.1944)	0.3536	0.6076	1.7746
Fast Stochastic	0.9995 (778.6111)	0.3699	0.6006	1.8555
MARSI	1.0723 (639.3611)	0.2081	0.6947	1.6917
MOM	1.0403 (681.4444)	0.1353	0.7349	1.3595
Online Anti-Z-BCRP	0.7579 (731.9444)	0.1935	0.4649	1.0924
Online Z-Anticor	1.3155 (694.5278)	0.4388	0.6363	2.3886
Online Z-BCRP	1.2818 (652.8611)	0.2637	0.8561	1.8341
PROC	0.8963 (718.0833)	0.1631	0.6305	1.2161
RSI	1.1339 (757.3889)	0.2544	0.6440	1.7059
SAR	0.7314 (654.1111)	0.0619	0.6683	0.8683
Slow Stochastic	1.1135 (793.2222)	0.3302	0.6955	2.1023
Williams %R	0.9416 (728.6944)	0.3150	0.4662	1.5131

**Table 18:** Group summary statistics of the overall rankings of experts grouped by their underlying strategy ( $\omega(i)$  where  $i = 1, \dots, 17$ ) for the daily trading. In brackets next to mean are the mean overall ranking of experts utilising each strategy.

Strategy	Mean	St. Dev.	Min	Max
EMA X-over	-0.00010	0.00633	-0.09745	0.08074
Ichimoku Kijun Sen	-0.00004	0.00723	-0.10467	0.06157
MACD	-0.00003	0.00725	-0.15993	0.08074
Moving Ave X-over	-0.00009	0.00644	-0.15993	0.11482
ACC	0.00007	0.00760	-0.15993	0.08028
BOLL	0.00002	0.00711	-0.06457	0.06480
Fast Stochastic	-0.00001	0.00847	-0.06469	0.06279
MARSI	0.00006	0.00612	-0.06788	0.06527
MOM	0.00004	0.00603	-0.06051	0.15820

Table 18 and Table 19.

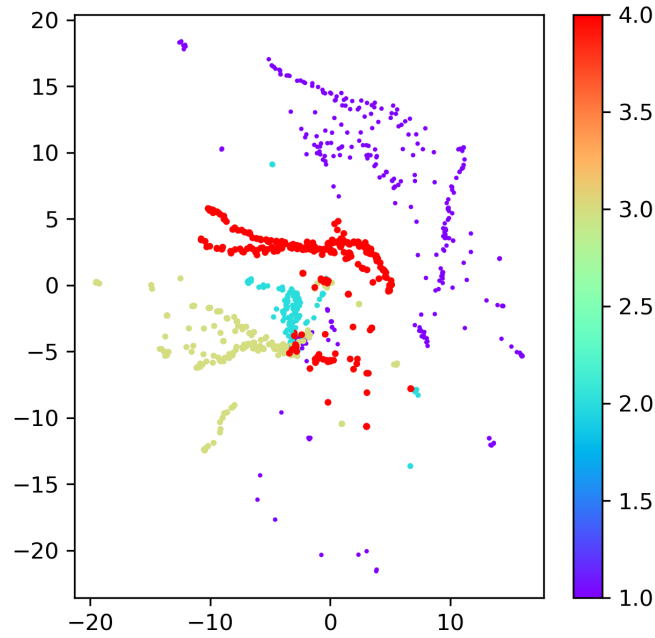
Online Anti-Z-BCRP	-0.00022	0.00773	-0.09847	0.09336
Online Z-Anticor	0.00021	0.00759	-0.06475	0.09773
Online Z-BCRP	0.00021	0.00771	-0.09336	0.09847
PROC	-0.00007	0.00733	-0.10467	0.09745
RSI	0.00010	0.00666	-0.06460	0.09745
SAR	-0.00023	0.00724	-0.10467	0.08724
Slow Stochastic	0.00009	0.00809	-0.06480	0.06820
Williams %R	-0.00006	0.00815	-0.06820	0.06317

**Table 19:** Group summary statistics of the expert’s profits and losses per period grouped by their underlying strategy ( $\omega(i)$  where  $i = 1, \dots, 17$ ).

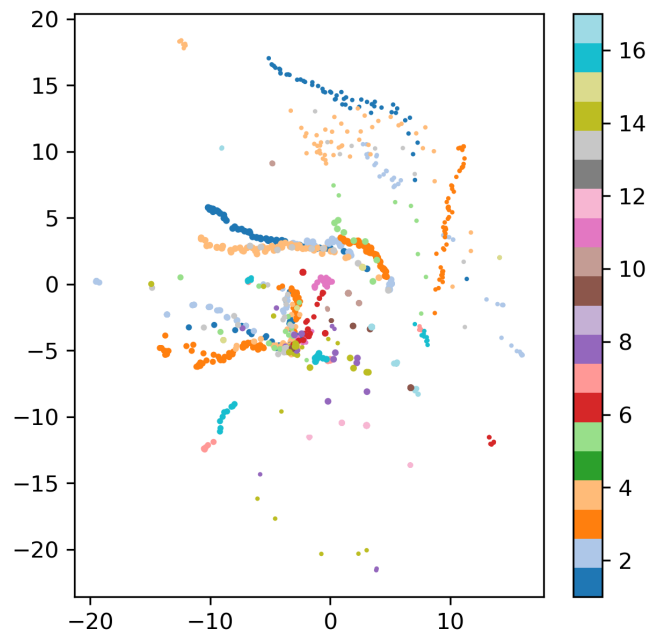
Figure 9(a) illustrates the 2-D plot of the latent space of a Variational Autoencoder (VAE) for the time series of wealth’s of all the experts with experts coloured by object cluster. It is not surprising that the expert’s wealth time series show quite well-defined clusters in terms of the stocks which experts choose to trade, as the stocks that each expert trades will be directly related to the decisions they make given the incoming data, and hence the corresponding returns (wealth) they achieve.

To provide some sort of comparison, in Figure 9(b) we plot the same results as above but this time we colour the experts in terms of their underlying strategy  $\omega(i)$ . The VAE seems to be able to pick up much clearer similarities (dissimilarities) between the experts based on the stocks they trade compared to which strategy they utilise providing evidence that the achieved wealth has a much stronger dependence on the stock choice rather than the chosen strategy. This may be an important point to consider and gives an indication that it may be worth considering more sophisticated ways to choose the stocks to trade rather than developing more (profitable) strategies. A discussion on the features that should be considered by a quantitative investment manager in assessing an assets usefulness is provided in Samo and Hendricks [61].

Next, we implement the CM test for statistical arbitrage on the daily cumulative profits and losses (**PL**) for the strategy without transaction costs. In order to have a result that is synonymous with [31], we choose a period of 400 days to test the overall trading strategy. We test the realised profits and losses for the 400-day period stretching from the 30<sup>th</sup> trading day until the 430<sup>th</sup> trading day. This is to allow for the algorithm to initiate and leave enough time for majority of the experts to have sufficient data to begin making trading decisions. Having simulated the 5000 different Min- $t$  statistics as in Section 6.1 step 5 (a) using simulations of the profit process in Eq. (58), Figure 10 illustrates the histogram of Min- $t$  values. The critical value  $t_c$  is then computed as the 0.95-quantile of the simulated distribution which refers to a significance level of  $\alpha = 5\%$  and is illustrated by the red vertical line. The resulting critical value is  $t_c = 0.7263$ . The Min- $t$  resulting from the realised incremental profits and losses of the overall strategy is 3.0183 (vertical green line). By Eq. (63), we recover a p-value of zero. Thus, we can conclude that there is significant evidence to reject the null of no statistical arbitrage at the 5% significance level.

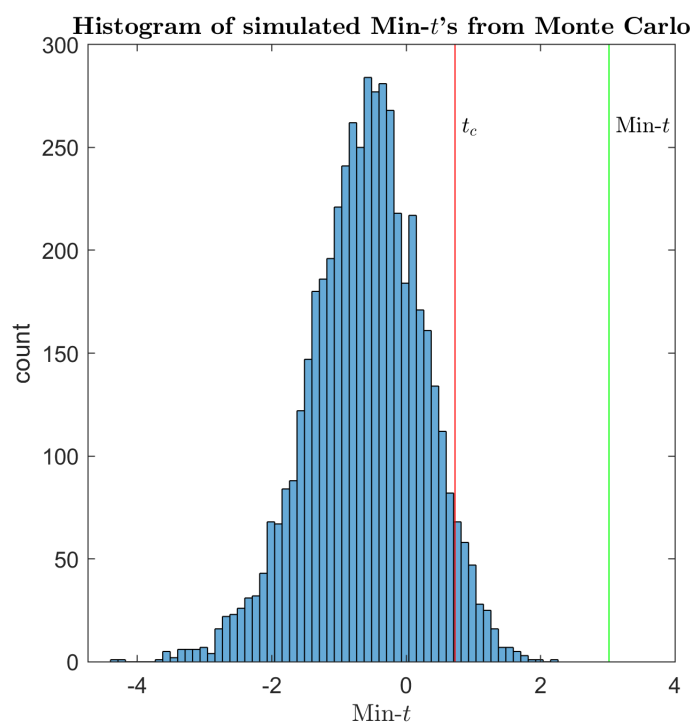


(a)



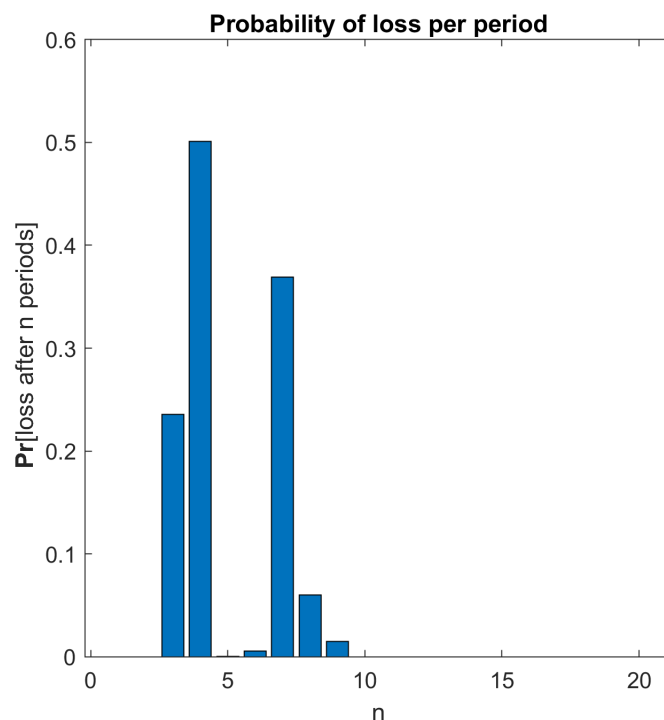
(b)

**Figure 9:** Figure 9(a) and Figure 9(b) show the latent space of a Variational Autoencoder on the time series of expert wealth's, implemented using Keras in Python. In Figure 9(a) experts are coloured by which of the 4 object clusters they trade whereas in Figure 9(b), experts are coloured by their underlying trading strategy  $\omega(i)$ .



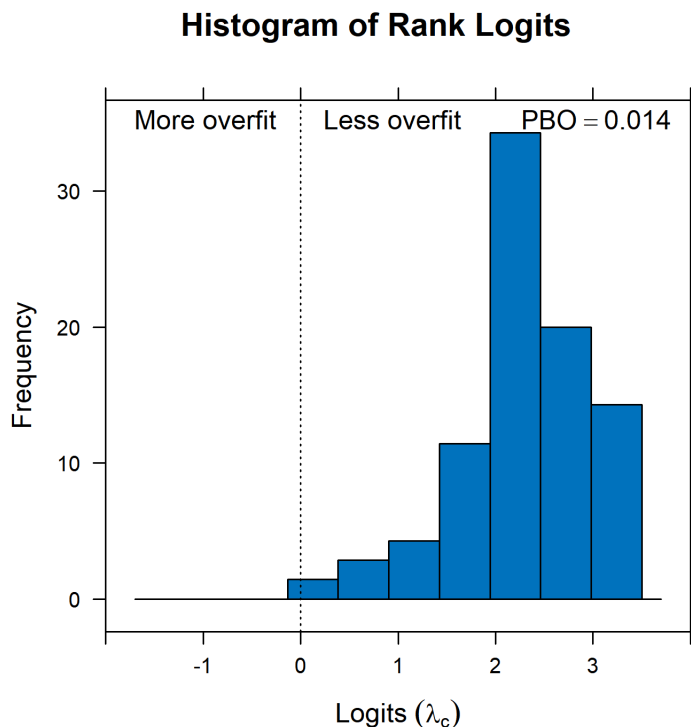
**Figure 10:** Histogram of the 5000 simulated  $\text{Min-}t$  statistics resulting from the CM test implemented on the simulated incremental process given in Eq. (58) along with the  $\text{Min-}t$  statistic (green) for the overall strategy's profit and loss sequence over the 400-day period stretching from the 30<sup>th</sup> trading day until the 430<sup>th</sup> trading day without any account for transactions costs.

In addition to testing for statistical arbitrage, we also report the number of days it takes for the probability of loss of the strategy to decline below 5% using Eq. (63) adjusted for the case of the CM model. As discussed in Section 6.1 step 7, for each  $n = 1, \dots, T$ , we perform MLE for  $\Delta\nu_{1:n}$  to get the parameter estimates. We then substitute these estimates into Eq. (63) to get an estimate of the probability of loss for the  $n^{\text{th}}$  period. This is all done in terms of the CM model. Figure 11 below illustrates the probability of loss for each of the first 30 trading days where we compute the probability of loss of the profit and loss process from the first trading period up to the  $n^{\text{th}}$  period for each  $n = 1, \dots, 30$ . As is evident from Figure 11, it takes roughly 10 periods for the probability of loss to decline below 5%.



**Figure 11:** Probability of the overall trading strategy generating a loss after for each of the 30 days from the first trading day through until the 30<sup>th</sup> trading day.

Finally, we analyse the probability of back-test overfitting inherent in the algorithms by simulating the algorithm on separate subsets of historic data as described in Section 7. For the daily data implementation, subsets of  $T_{BL} = 60$  days will be used for each individual simulation. The idea is to run the learning algorithm on the first  $T_{BL}$  days, step forward  $T_{BL}$  days and re-run the algorithm. This process is continued until there is not a sufficient amount of data to step forward  $T_{BL}$  days. This means a total of  $N = 30$  simulations of such length are run on the set of data described in Section 8.1. We then form the return matrix  $M$  as required in Section 7.2 step 1 which consists of the profits and losses of each of the  $N$  simulations to recover  $M_{60 \times 30}$ . The rest of the CSCV procedure is then implemented as outlined in Section 7.2. Illustrated in Figure 12 is the associated histogram of logits  $\lambda_C$  for  $C \in C_S^{S/2}(i) \forall i = 1, \dots, S$ . The resulting PBO is roughly 1% which indicates a very small presence of back-test overfitting.

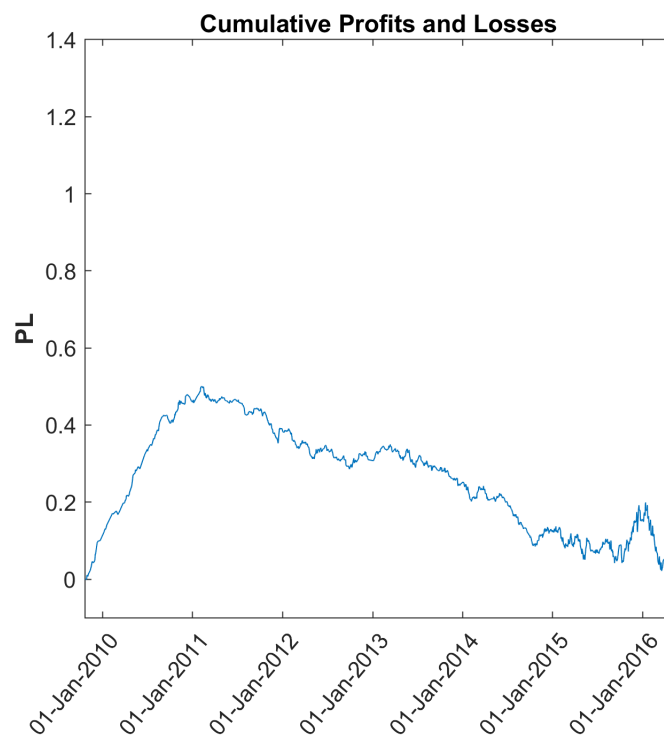


**Figure 12:** Probability of back-test overfitting for 30 simulations of the online learning algorithm on daily historic data.

### 8.1.2 Transaction Costs

In this section we reproduce the results from above but this time including transaction costs for daily trading as discussed in Section 5.5. Once direct and indirect (Eq. (50)) costs have been computed, the idea is to subtract off the transaction cost from the profit and losses of each day and compound the resulting value onto  $S_{t-1}$  to get the wealth for period  $t$ . These daily profit and losses are added to get the cumulative profit and loss **PL**.

It is clear from Figure 13, which illustrates the profits and losses (**PL**) of the overall strategy less the transaction costs for each period, that consistent losses are incurred when transaction costs are incorporated. Furthermore, there is no evidence to reject the no statistical arbitrage null hypothesis (see Proposition 2) as the Min- $t$  statistic resulting from the overall strategy is well below the critical value at the 95<sup>th</sup> percentile of the histogram as illustrated in Figure 14(a). Moreover, although the probability of loss (see Eq. (61)) of the strategy with transaction costs included initially converges to zero, it soon after begins oscillating between a probability of zero and one before eventually settling on one (Figure 14(b)).



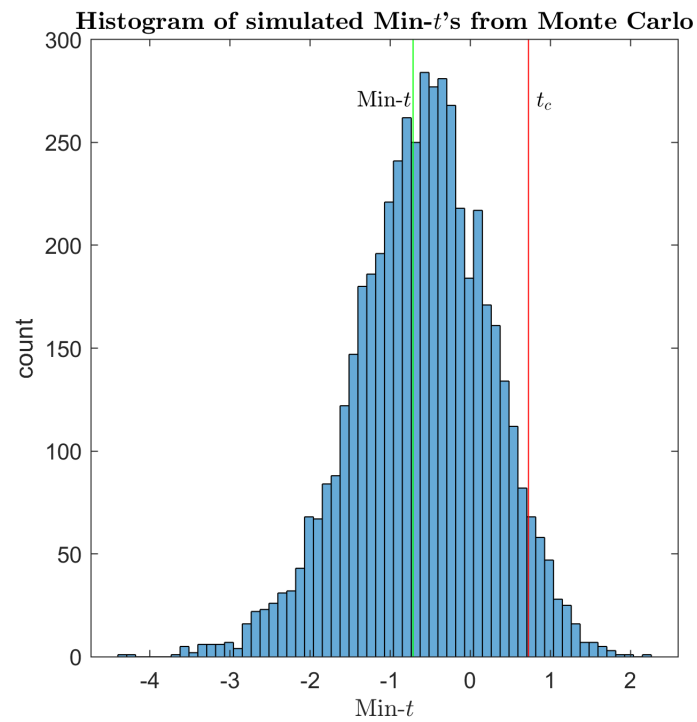
**Figure 13:** Profits and losses (PL) for overall strategy for daily trading with transaction costs.

Considering the above evidence contained in Figure 13, Figure 14(a) and Figure 14(b), the overall strategy does not survive historical back tests in terms of profitability when transaction costs are considered and hence may not be well suited for an investor utilising daily data whom has a limited time to make adequate profits. This is in agreement with Schulmeister [62] in that there is a strong possibility that stock price and volume trends have shifted to higher frequencies than the daily time scale and resultantly, trading strategies' profits have over time diminished on such time scales.

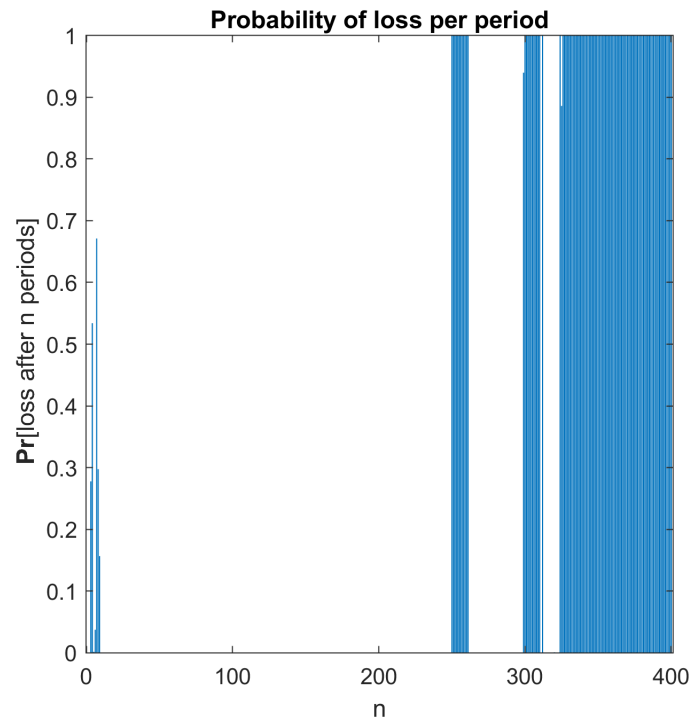
## 8.2 Intraday-Daily Data

Below we report the results of the algorithm implementation for a combination of intraday and daily JSE data as discussed in Section 5.3. We run the algorithm on the OHLCV data of 15 most liquid stocks from a set of 30 of the JSE Top 40<sup>43</sup>. Liquidity is calculated in terms of average daily trade volume for the first 4 days of the period 02-01-2018 to 09-03-2018. The set of 15 stocks, ranked from most to least liquid over the specified period, is as follows: FSR:SJ, GRT:SJ, SLM:SJ, BGA:SJ, SBK:SJ, WHL:SJ, CFR:SJ, MTN:SJ, DSY:SJ, IMP:SJ, APN:SJ, RMH:SJ, AGL:SJ, VOD:SJ and BIL:SJ. The remaining 40 days' data for the aforementioned period is utilised to run the learning algorithm on. As in the daily data implementation, we again analyse the two cases of trading with and without transaction costs which we report in the following two subsections below.

<sup>43</sup>See Appendix A.2 for the list of the 30 stocks along with their Bloomberg ticker symbols



(a)



(b)

**Figure 14:** Figure 14(a) illustrates the histogram of the 5000 simulated Min- $t$  statistics resulting from the CM model and the incremental process given in Eq. (58) along with the Min- $t$  statistic (green) for the overall strategy's daily trading profit and loss sequence over the 400-day period stretching from the 30<sup>th</sup> trading day until the 430<sup>th</sup> trading day with transactions costs incorporated. Also illustrated is the critical value at the 5% significance level (red). Figure 14(b) shows the probability of the overall trading strategy generating a loss for each of the first 400 trading days.

### 8.2.1 No Transaction Costs

Without transaction costs, the cumulative wealth achieved by the overall strategy, illustrated in Figure 15(a) evolves similarly to an exponential function over time. The associated profits and losses are displayed in Figure 15(b). Incremental profits and losses are obviously a lot smaller compared to those achieved by the daily data case which results in a much smoother function in comparison to the daily data case (Figure 15(b)).

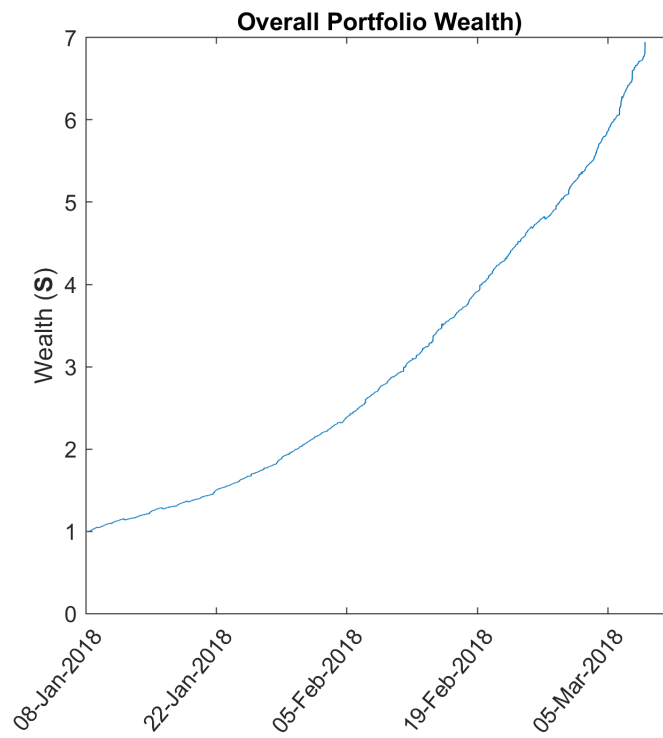
Table 20 is the intraday-daily analogue of Table 18. In this case, the exponential moving crossover strategy (EMA X-over) produces the expert with the greatest wealth and acceleration (ACC) the expert with the least terminal wealth. Exponential moving crossover also produces experts with the highest variation in terminal wealth's. Price rate of change (PROC) is by far the strategy with the best mean ranking experts among all experts however Z-BCRP produces experts with highest mean terminal wealth.

Strategy	Mean (mean rank)	St. Dev.	Min	Max
EMA X-over	1.0024 (662.7639)	0.0094	0.9801	1.0375
Ichimoku Kijun Sen	0.9989 (710.3750)	0.0085	0.9663	1.0303
MACD	0.9995 (684.8704)	0.0067	0.9720	1.0202
Moving Ave X-over	1.0012 (708.7824)	0.0058	0.9766	1.0204
ACC	0.9953 (831.3333)	0.0079	0.9646	1.0048
BOLL	0.9974 (712.9722)	0.0069	0.9787	1.0089
Fast Stochastic	0.9991 (711.4167)	0.0040	0.9871	1.0085
MARSI	0.9973 (736.2500)	0.0062	0.9824	1.0094
MOM	0.9982 (723.1389)	0.0087	0.9700	1.0082
Online Anti-Z-BCRP	0.9980 (597.3056)	0.0062	0.9828	1.0103
Online Z-Anticor	1.0015 (655.7778)	0.0058	0.9896	1.0180
Online Z-BCRP	1.0031 (566.5833)	0.0069	0.9898	1.0149
PROC	0.9980 (445.1389)	0.0064	0.9814	1.0140
RSI	0.9997 (535.5833)	0.0065	0.9861	1.0171
SAR	0.9945 (499.7222)	0.0053	0.9790	1.0005
Slow Stochastic	1.0007 (508.5278)	0.0048	0.9927	1.0173
Williams %R	1.0020 (536)	0.0034	0.9957	1.0133

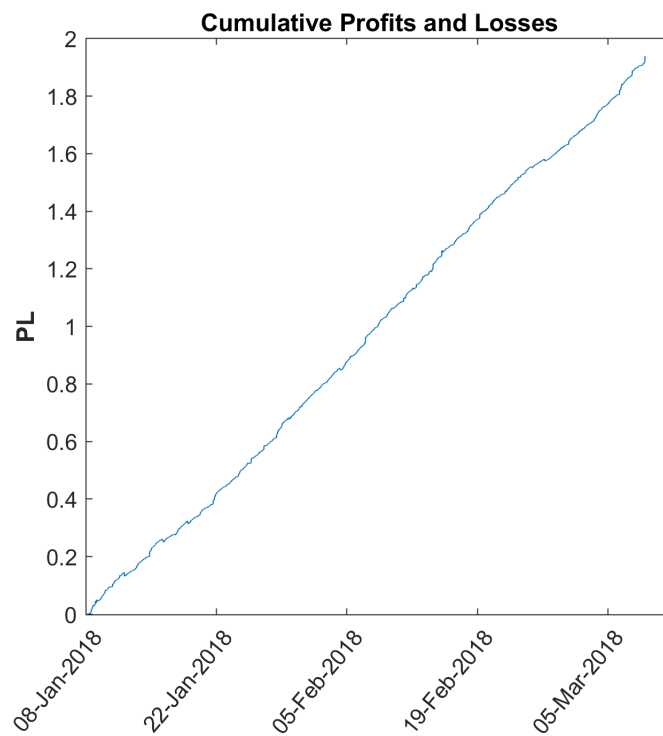
**Table 20:** Group summary statistics of the overall rankings of experts grouped by their underlying strategy ( $\omega(i)$  where  $i = 1, \dots, 17$ ) for intraday-daily trading. In brackets are the mean overall ranking of experts utilising each strategy.

Again, as for the daily data case, we implement a test for statistical arbitrage for intraday-daily trading without transaction costs for 400 trading periods starting from the 6<sup>th</sup> time bar of the 2nd trading day<sup>44</sup> using the intraday-daily profit and loss sequence (PL). Figure 17 illustrates the histogram of simulated Min- $t$  values with the 0.95-percentile of the simulated distribution representing the critical value  $t_c$  (red) and the Min- $t$  (green) resulting from the incremental profits and losses of the overall strategy resulting from the learning algorithm. The resulting critical value is 0.7234

<sup>44</sup>This corresponds to the trading period within which the very first trading decisions are made.

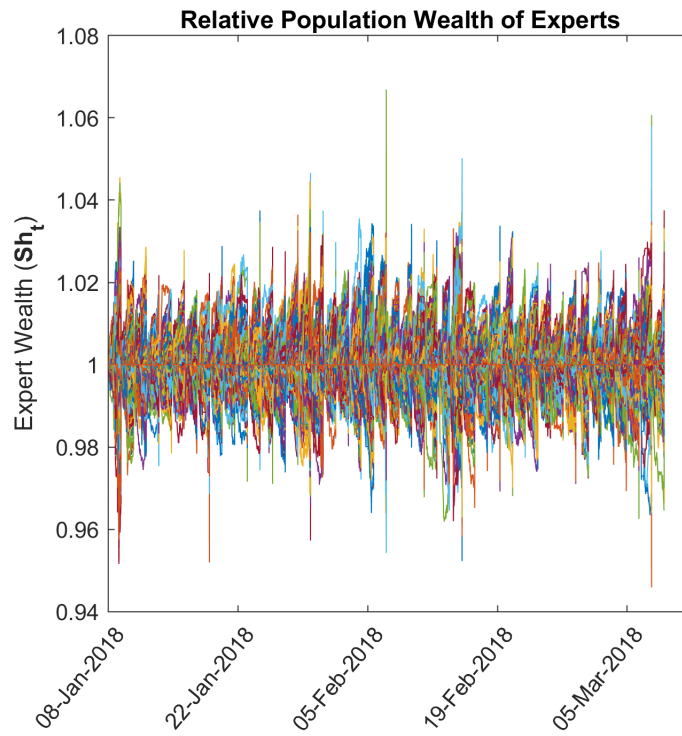


(a)

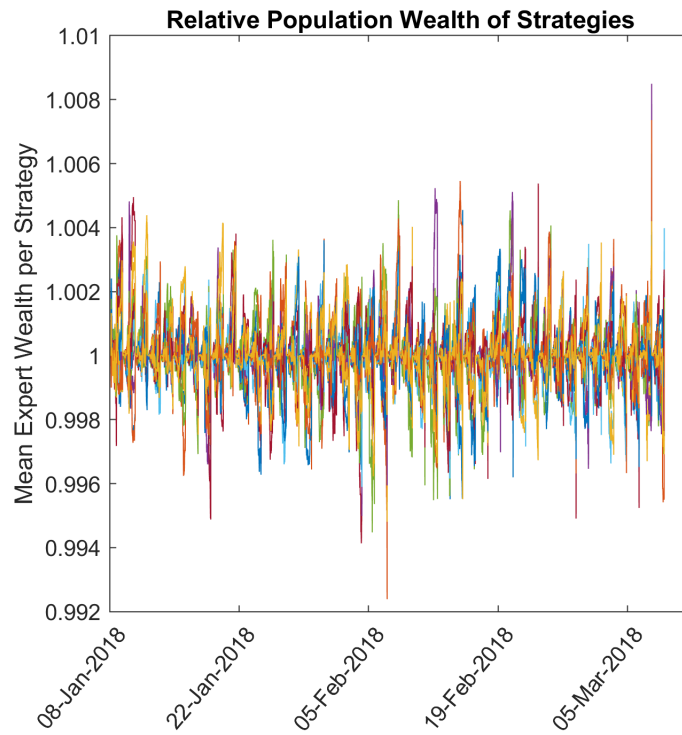


(b)

**Figure 15:** Fig. 15(a) shows the overall cumulative portfolio wealth ( $S$ ) for intraday-daily trading with no transaction costs. Fig. 15(b) illustrates the profits and losses ( $PL$ ) for overall strategy for intraday-daily trading with no transaction costs.



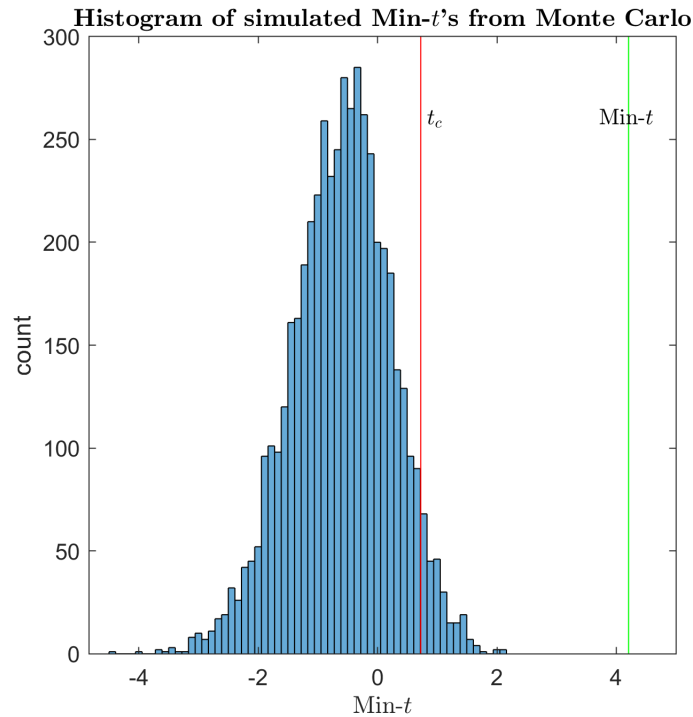
(a)



(b)

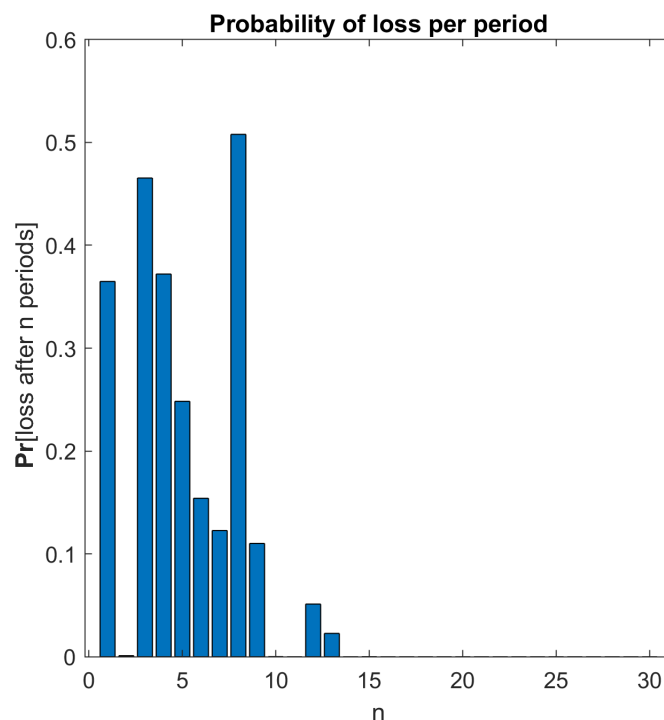
**Figure 16:** Fig. 16(a) illustrates the expert wealth ( $S_{h_t}$ ) for all  $\Omega$  experts for intraday-daily trading with no transaction costs. Fig. 16(b) illustrates the mean expert wealth of all experts for each trading strategy ( $\omega(i)$ ) for intraday-daily trading with no transaction costs.

and the Min- $t$  value is 3.8558. Thus, there is strong evidence to reject the null hypothesis of no statistical arbitrage as the resulting p-value is identical to zero.



**Figure 17:** Histogram of the 5000 simulated Min- $t$  statistics resulting from the profit and loss process without transaction costs taken into account along with the Min- $t$  statistic for the overall strategy (green) and the critical value at the 5% significance level (red). The profit and loss process is extracted from the 6<sup>th</sup> time bar on the second day, which corresponds to the period when active trading begins, until the 400<sup>th</sup> time bar hence. This corresponds to roughly four and a half days worth of trading profits.

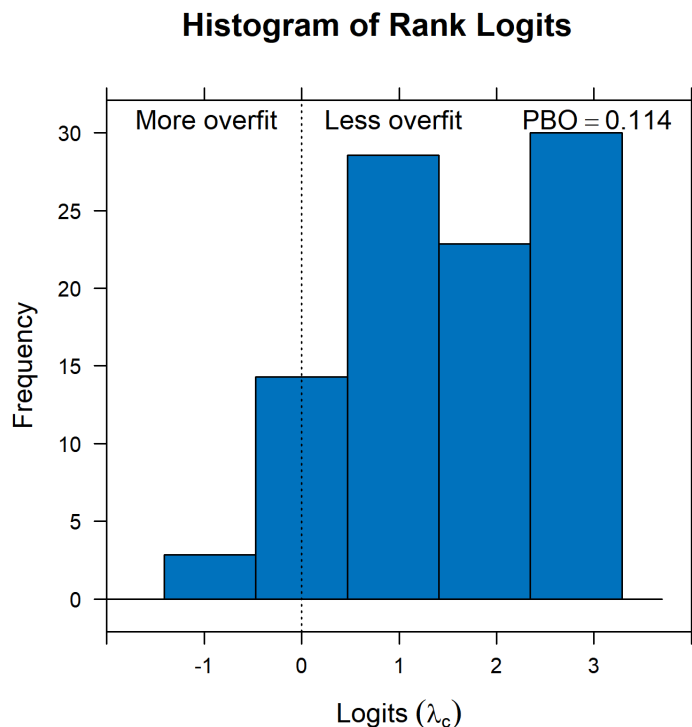
Figure 18 illustrates the probability of loss for each of the 30 periods starting from the 6<sup>th</sup> time bar of the 2nd trading day using the wealth process ( $\mathbf{S}$ ) from the intraday-daily trading algorithm without transaction costs. It takes roughly an hour (13 periods) for the probability of loss to converge to zero.



**Figure 18:** Probability of the overall trading strategy generating a loss after  $n$  periods for each  $n = 1, \dots, 30$  of the intraday-daily profit and loss process (**PL**) without transaction costs. The profits and losses are taken starting from the  $5^{th}$  time bar of the second day when active trading commences as the first day is required as a look-back for the daily trading component.

Finally, we implement the CSCV procedure on the historic intraday-daily data. Rather than set  $T_{BL} = 60$ , as in the daily data implementation,  $T_{BL}$  is chosen to be 3 days. This results in a total of  $N = 22$  simulations of length  $T_{BL} = 178$ .<sup>45</sup> Again construct  $M_{178 \times 22}$  by taking the profits and losses resulting from the simulations. A PBO of approximately 11% is recovered from the CSCV procedure which is more than the 1% resulting from the daily data implementation (Figure 12). A reasonably low PBO is recovered from both implementations which is definitely favourable as it indicates the algorithm's ability to perform desirably on unseen data (low generalization error).

<sup>45</sup>89 periods per day (88 intraday time bars plus a daily data time bar) but the first day (of the 3 day period) is required to compute the volatility estimates for the beginning of the day's transaction cost estimates (see Section 5.5.1 for more details) which results in  $T_{BL} = 89 * 2 = 178$



**Figure 19:** Probability of back-test overfitting for 20 simulations of the online learning algorithm on intraday-daily historic data.

## 8.2.2 Transaction Costs

We now report the results of the algorithm run on the same intraday-daily data as in the subsection above but this time with transaction costs incorporated as outlined in Section 5.5. Figure 20(a) and the figure inset illustrate the overall cumulative portfolio wealth ( $S$ ) and profits and losses ( $PL$ ) respectively for intraday-daily trading with transaction costs. For comparative reasons, the axes are set to be equivalent to those in the case of no transaction costs (Figure 20(a) and the inset of the figure). Surprisingly, even with a total daily trading cost (direct and indirect) of roughly 130bps, which is a fairly aggressive approach, the algorithm is able to make satisfactory returns which is in contrast to the daily trading case (Figure 13). Furthermore, Figure 21(a) provides significant evidence that the no statistical arbitrage null hypothesis can be rejected and has an almost identical Min- $t$  statistic (3.86 compared to 3.95) to that of the case of no transaction costs (Figure 17). What is even more comforting is the fact that even when transaction costs are considered, the probability of loss per trading period converges to zero albeit slightly slower (26 trading periods as illustrated in Figure 21(b)) than the case of no transaction costs (13 trading periods as illustrated in Figure 18).

The above results for intraday-daily trading are in complete contrast to the case of daily trading with transaction costs whereby the no statistical arbitrage null could not be rejected, the probability of loss did not converge to zero and remain there, and trading profits steadily declined over the trading horizon. This suggests that the proposed algorithm may be much better suited to trading at higher frequencies. This

is not surprising and is in complete agreement with Schulmeister [62] who argues that the profitability of technical trading strategies had declined over from 1960 before becoming unprofitable from the 1990s. The same set of technical trading strategies are then implemented on 30-minute data and the evidence suggests that such strategies returned adequate profits between 1983 and 2007 however, such profits declined slightly between 2000 and 2007 in comparison to the 1980's and 1990's. This suggests that markets may have become more efficient and even the possibility that stock price and volume trends have shifted to even higher frequencies than 30 minutes [62]. This supports the choice to trade the algorithm proposed in this paper on at least 5-minute OHLCV data and reinforces the suggestion provided in Section 5.4 that ultimately, the most desirable implementation of the algorithm would be in volume-time which is best suited for high frequency trading.

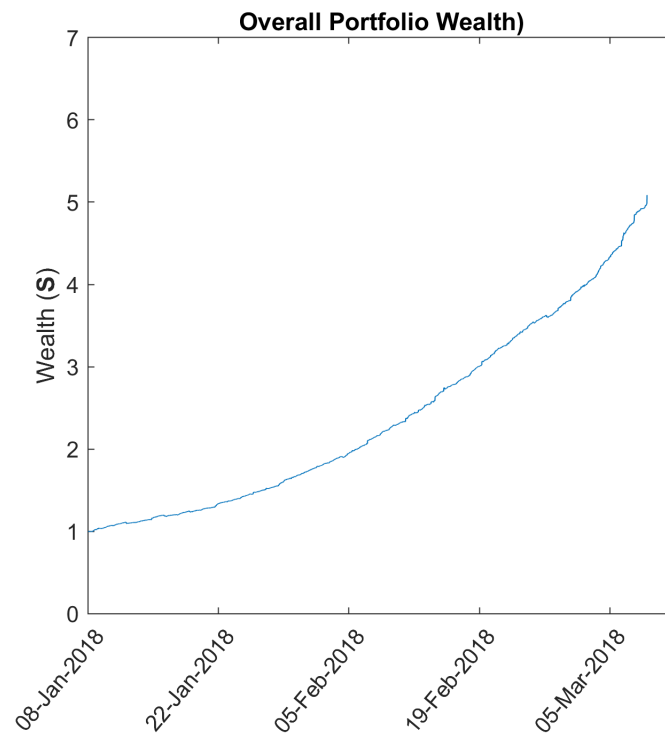
## 9 Conclusion

We have developed a learning algorithm built from a base of technical trading strategies for the purpose of trading equities on the JSE that is able to provide favourable returns when ignoring transaction costs, under both daily and intraday trading conditions. The returns are reduced when transaction costs are considered in the daily setting, however there is sufficient evidence to suggest that the proposed algorithm is really well suited to intraday trading.

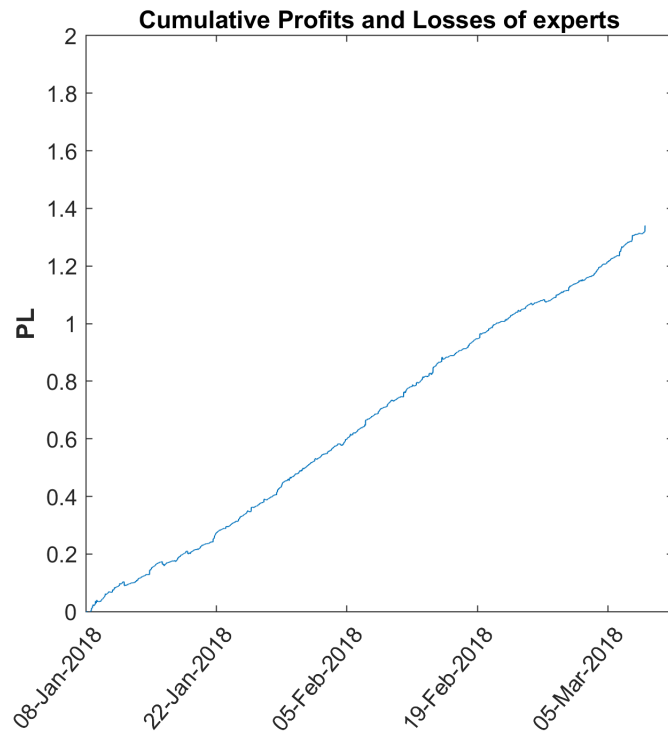
This is reinforced by the fact that there exists meaningful evidence to reject a carefully defined null hypothesis of no statistical arbitrage in the overall trading strategy even when a reasonably aggressive view is taken on intraday trading costs. We are also able to show that it in both the daily and intraday-daily data implementations that the probability of loss declines below 5% relatively quickly which strongly suggests that the algorithm is well suited for a trader whose preference or requirement is to make adequate returns in the short-run. Furthermore, we recover a low probability of back-test overfitting in both the daily and intraday-daily implementations, which was expected to be the case prior to analysing the results, as the algorithm in fact makes all trading decisions out-of-sample. This indicates the algorithms ability to perform desirably on unseen data, which is an attractive feature for any quantitative trader.

The superior performance of the algorithm for intraday trading is in agreement with Schulmeister [62], who concluded that while the daily profitability of a large set of technical trading strategies has steadily declined since 1960 and has been unprofitable since the onset of the 1990's, trading the same strategies on 30-minute (intraday) data between 1983 and 2007 has produced decent average gross returns. However, such returns have slowly declined since the early 2000's. In conclusion, the proposed algorithm is much better suited to trading at higher frequencies.

We are however cognisant of the fact that intraday trading will also typically require a large component of accumulated trading profits to finance frictions, concretely to fund direct, indirect and business model costs [6]. For this reason, we are careful to remain sceptical with this class of algorithms long-run performance when trading with real money in a live trading environment. The current design of the algorithm is not yet ready to be traded on live market data, however with some effort it is easily transferable to such use cases given the sequential nature of the algorithm and its inherent ability to receive and adapt to new incoming data while making appropriate trading decisions based on the new data. Concretely, the algorithm should be deployed

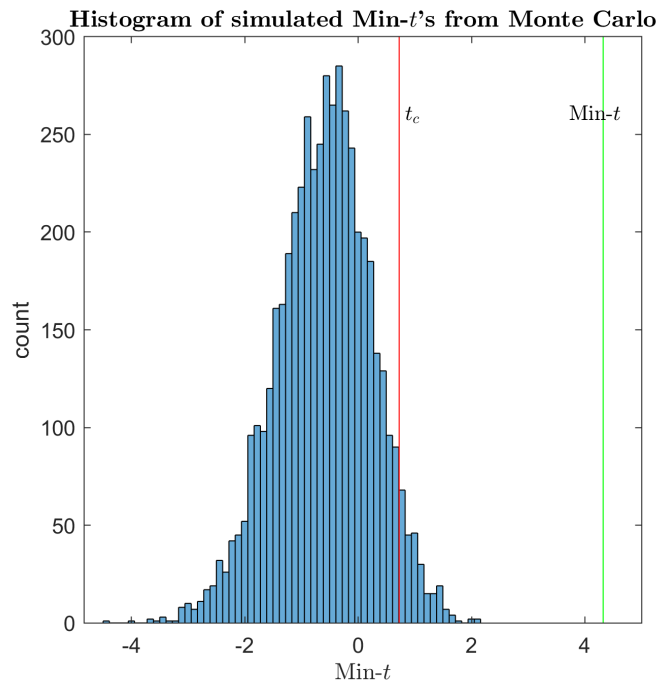


(a)

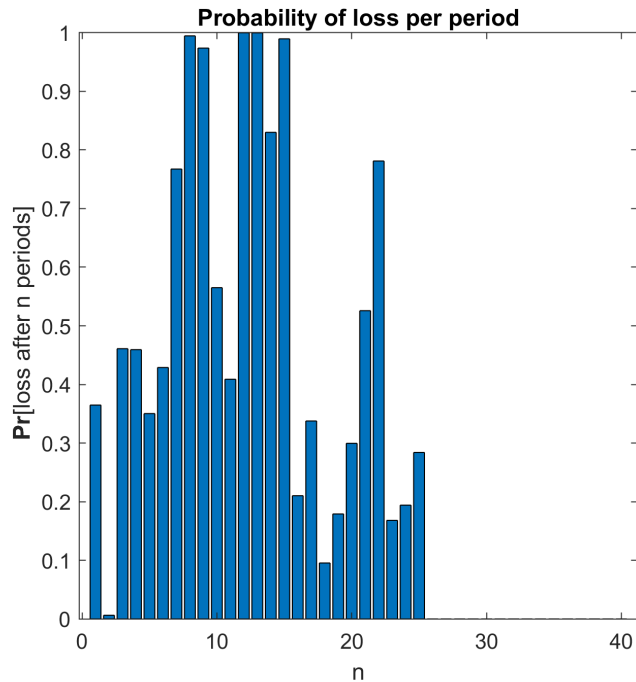


(b)

**Figure 20:** Figure 20(a) and Figure 20(b) illustrate the overall cumulative portfolio wealth ( $S$ ) and profits and losses ( $PL$ ) respectively for intraday-daily trading with transaction costs.



(a)



(b)

**Figure 21:** Figure 21(a) shows the histogram of the 5000 simulated Min- $t$  statistics resulting from the CM model and the incremental process given in Eq. (58). Also illustrated is the Min- $t$  statistic (green) for the first 400 trading periods from commencement of active trading ( $5^{\text{th}}$  time bar of the second day) for intraday-daily profit and losses with transaction costs incorporated along with the critical value at the 5% significance level (red). Figure 21(b) illustrates the probability of the overall trading strategy generating a loss after  $n$  periods for each  $n = 1, \dots, 30$  of the intraday-daily profit and loss process (PL) with transaction costs incorporated commencing from the first period of active trading ( $5^{\text{th}}$  time bar of the second day).

in the context of volume-time trading rather than the calendar time context considered in this work.

## 9.1 Future Work

Possible future work includes implementing the algorithm in volume-time which will be best suited for dealing with a high frequency implementation of the proposed algorithm given the intermittent nature of order-flow. We also propose replacing the learning algorithm with an online (adaptive) neural network that has the ability to predict optimal holding times of stocks. Another interesting line of work that has been considered is to model the population of trading experts as competing in a predator-prey environment [63, 64]. This was an initial key motivation for the research project - to find which collections of technical trading strategies can be grouped collectively and how these would interact with each other. This includes utilising cluster analysis to group together or separate trading experts based on their similarities and dissimilarities, and hence make appropriate inferences regarding their interactions and behaviours at the level of collective and emergent dynamics.

## 10 References

- [1] L. Györfi, G. Ottucsák & H. Walk (2012). *Machine Learning for Financial Engineering*. Imperial College Press.
- [2] T. M. Cover & E. Ordentlich (1996). Universal Portfolios with Side Information, *IEEE Transactions on Information Theory*, 42(2), 348-363.
- [3] T. M. Cover (1991). Universal Portfolios. *Mathematical Finance*, 1(1), 1-29.
- [4] P. H. Algoet & T. M. Cover (1988). Asymptotic optimality and asymptotic equipartition properties of logoptimum investment. *The Annals of Probability*, 16(2), 876-898. <https://doi.org/10.1214/aop/1176991793>.
- [5] L. Györfi, F. Udina & H. Walk (2008). Nonparametric nearest neighbor based empirical portfolio selection strategies, *Statistics & Risk Modeling*, De Gruyter, 26(2), 145-15.
- [6] F. Loonat & T. Gebbie (2018). Learning zero-cost portfolio selection with pattern matching. *PLoS ONE* 13(9): e0202788. <https://doi.org/10.1371/journal.pone.0202788>.
- [7] E. Chan (2009). *Quantitative Trading: How to Build Your Own Algorithmic Trading Business*. John Wiley & Sons Inc, Wiley Trading. ISBN: 9780470466261.
- [8] J. F. Clayburg (2002). *Four Steps to Trading Success: Using Everyday Indicators to Achieve Extraordinary Profits*. John Wiley & Sons. ISBN: 978-0-471-21799-2.
- [9] G. Creamer & Y. Freund (2010). Automated trading with boosting and expert weighting. *Quantitative Finance*, 10(4), 401-420. <http://dx.doi.org/10.1080/14697680903104113>
- [10] T. S. Chande & S. Kroll (1994). *The new technical trader: boost your profit by plugging into the latest indicators*. John Wiley & Sons, Wiley Finance. ISBN: 9780471597803.
- [11] M. D. Rechenthin (2014). *Machine-learning classification techniques for the analysis and prediction of high-frequency stock direction*. PhD thesis, University of Iowa. <http://ir.uiowa.edu/etd/4732>.
- [12] D. R. Aronson (2007). *Evidence-Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signals*. John Wiley & Sons, Inc.
- [13] A. Lo, H. Mamaysky, & J. Wang (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *The Journal of Finance*, 4.
- [14] A. W. Lo & J. Hasanhodzic (2009). *The Heretics of Finance: Conversations with Leading Practitioners of Technical Analysis*. Bloomberg Press, (800) 869-1231.
- [15] L. N. Kestner (2003). *Quantitative Trading Strategies: Harnessing the Power of Quantitative Techniques to Create a Winning Trading Program*. McGraw-Hill.
- [16] C-H. Park & S. Irwin (2004). *The Profitability of Technical Analysis: A Review*. SSRN Electronic Journal.
- [17] K. Hou, C. Xue & L. Zhang (2017). *Replicating Anomalies*. NBER Working Paper Series, No. 23394. <http://www.nber.org/papers/w23394>.
- [18] D. H. Bailey, J. M. Borwein, M. M. López de Prado, & Q. J. Zhu (2014). Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance. *Notices of the AMS*, 61(5), 458-471.

- [19] D. H. Bailey, J. M. Borwein, M. L. de Prado & Q. J. Zhu (2016). The Probability of Backtest Overfitting. *Journal of Computational Finance* 20, No. 4, 39-69.
- [20] L. Teixeira & A. L. de Oliveira (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert Systems with Applications* 37, 6885-6890.
- [21] V. Vella (2016). Improving risk-adjusted performance in high-frequency trading: the role of fuzzy logic systems. PhD thesis, University of Essex.
- [22] S. Hogan, R. Jarrow, M. Teo & M. Warachka (2004). Testing market efficiency using statistical arbitrage with applications to momentum and value strategies. *Journal of Financial Economics*, 73, 525-565.
- [23] E. F. Fama (1970). Efficient capital markets: A review of theory and empirical work. Proceedings of the Twenty-Eighth Annual Meeting of the American Finance Association. *Journal of Finance*, 25(2), 383-417.
- [24] A. Shleifer (2000). *Inefficient Markets: An Introduction to Behavioral Finance*. Oxford, UK: Oxford University Press.
- [25] A. Lo (2004). The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective. *Journal of Portfolio Management*, 5(30), 15-29.
- [26] D. Farmer & A. Lo (1999). Frontiers of Finance: Evolution and Efficient Markets. *Proceedings of the National Academy of Sciences* 96, 9991-9992.
- [27] D. Farmer (2002). Market Force, Ecology and Evolution. *Industrial and Corporate Change*, 11, 895-953.
- [28] J. D. Farmer & S. Skouras (2013). An ecological perspective on the future of computer trading. *Quantitative Finance*, 13(3), 325-346.
- [29] F. Black (1986). Noise. *Journal of Finance*, 41(3), 529-543.
- [30] J-P. Bouchaud, S. Ciliberti, Y. Lemperiere, A. Majewski, P. Seager & K. S. Ronia (2017). Black Was Right: Price Is Within a Factor 2 of Value. Available at SSRN: <https://ssrn.com/abstract=3070850>.
- [31] R. Jarrow, M. Teob, Y-K Tse & M. Warachka (2012). An improved test for statistical arbitrage. *Journal of Financial Markets*, 15, 47-80.
- [32] E. F. Fama & M. E. Blume (1996). Filter rules and stock market trading. *Journal of Business*, 39, 226-241.
- [33] B. Johnson (2010). *Algorithmic trading & DMA: An introduction to direct access trading strategies*. London: 4Myeloma Press.
- [34] R. Kissell, Ph.D (2014). *The Science of Algorithmic Trading and Portfolio Management*. Elsevier Inc.
- [35] S. Albers (2003). Online algorithms: a survey. *Mathematical Programming* 97(1).
- [36] B. Li & S. C. Hoi (2016). *Online Portfolio Selection: Principles and Algorithms*. Taylor & Francis Group.
- [37] R. Dochow (2016). *Online Algorithms for the Portfolio Selection Problem*. Springer Gabler.
- [38] A. Borodin, R. El-Yaniv & V. Gogan (2004). Can we learn to beat the best stock. *Journal of Artificial Intelligence Research*, 21, 579-594.
- [39] B. Li & S. C. Hoi (2014). Online portfolio selection: A survey. *ACM Computing Survey*, 46(3), 35.
- [40] W. Lee (2000). *Theory and methodology of tactical asset allocation*. John Wiley & Sons, 65.

- [41] olpsR - On-line Portfolio Selection with R (2015). <https://github.com/ngloe/olpsR>
- [42] J. Murphy (1999). *Technical analysis of the financial markets*. New York Institute of Finance.
- [43] [http://www.kumotrader.com/ichimoku\\_wiki/index.php?title=Main\\_Page](http://www.kumotrader.com/ichimoku_wiki/index.php?title=Main_Page)
- [44] D. Linton (2010). *Cloud Charts: Trading Success with the Ichimoku Technique*. ISBN: 9780956517104.
- [45] Ichimoku 101. IchimokuTrade.com. [https://www.ichimokutrade.com/articles/Ichimoku\\_Ebook.pdf](https://www.ichimokutrade.com/articles/Ichimoku_Ebook.pdf)
- [46] T. Gebbie (2010). MATLAB Technical Analysis toolbox (“technical”).
- [47] M. B. Fayek, H. M. El-Boghdadi, S. M. Omran (2013). Multi-objective Optimization of Technical Stock Market Indicators using GAs. *International Journal of Computer Applications*, 68(20).
- [48] J.W. Wilder (1978). *New Concepts in Technical Trading Systems*. Trend Research. ISBN: 9780894590276.
- [49] C. T. Brownlees & G. M. Gallo (2006). Financial econometric analysis at ultra-high frequency: Data handling concerns. *Computational Statistics & Data Analysis*, 51(4), 2232-2245.
- [50] M. K. Bhasin (2013). The volume clock: Insights into the high-frequency paradigm (digest summary). *CFA Digest*, 43(1).
- [51] Trade talk Blog: Constant Volume Bars vs. Time Bars (23 October 2012). Trading Technologies. <https://www.tradingtechnologies.com/blog/2012/10/23/constant-volume-bars-vs-time-bars/>
- [52] The Hidden Costs of Trading. [http://pages.stern.nyu.edu/~adamodar/New\\_Home\\_Page/invemgmt/trading.htm](http://pages.stern.nyu.edu/~adamodar/New_Home_Page/invemgmt/trading.htm)
- [53] J. Gatheral (2010). No-dynamic arbitrage and market Impact. *Quantitative Finance*, 10(7), 759.
- [54] G. Huberman & W. Stanzl (2005). *Optimal Liquidity Trading*. Springer: Review of Finance, 9, 165-200.
- [55] J-P. Bouchaud, Y. Gefen, M. Potters & M. Wyart (2004). *Fluctuations and Response in Financial Markets: The Subtle Nature of ‘Random’ Price Changes*.
- [56] T. Andersen, T. Bollerslev, F. X. Diebold, & P. Labys (2001). The Distribution of Realized Exchange Rate Volatility. *Journal of the American Statistical Association*, 96, 42-55.
- [57] S-H. Poon (2008). *Volatility Estimation*. <https://www.cmegroup.com/trading/fx/files/volEst.pdf>
- [58] I. Karatzas & S.E. Shreve (1991). *Brownian Motion and Stochastic Calculus*. Graduate Texts in Mathematics, Springer New York, ISBN: 9780387976556.
- [59] J-J. D. de Wit (2013). *Statistical Arbitrage in South Africa*. Minor dissertation submitted to the faculty of Commerce in partial fulfilment of the requirements for the Masters degree in Mathematical Finance at the University of Cape Town.
- [60] etfSA STeFI-linked Money Market Deposit (5 April 2011). Managed by Nedbank Beta Solutions.
- [61] Y-L. K. Samo & D. Hendricks (2018). What Makes An Asset Useful?. arXiv:1806.08444, <https://arxiv.org/abs/1806.08444>

- 
- [62] S. Schulmeister (2009). Profitability of technical stock trading: Has it moved from daily to intraday data?. *Review of Financial Economics*, 18(4), 190-201.
- [63] J. D. Farmer (2000). A Simple Model for the Nonequilibrium Dynamics and Evolution of a Financial Market. *International Journal of Theoretical and Applied Finance*, 3(3), 425-441.
- [64] N. Johnson, G. Zhao, E. Hunsader, H. Qi, J. Meng & Brian Tivnan (2013). Abrupt rise of new machine ecology beyond human response time. *Scientific Reports* 3(2627).
- [65] D. Farmer & J. Geanakoplos (2008). The Virtues and Vices of Equilibrium and the Future of Financial Economics. *Wiley Periodicals*, 14(3).
- [66] M. Harvey, D. Hendricks, T. Gebbie & D. Wilcox (2017). Deviations in expected price impact for small transaction volumes under fee restructuring, *Physica A: Statistical Mechanics and its Applications*, 471, 416-426. <https://www.sciencedirect.com/science/article/pii/S0378437116308470>.
- [67] F. Loonat & T. Gebbie (2016). MATLAB Pattern Matching and Learning Class ("pattern").
- [68] M. Horváth & A. Urbán (2011). Chapter 4: Growth Optimal Portfolio Selection with Short Selling and Leverage. *World Scientific Review Volume*.
- [69] K. Akcoglu, P. Drineas & M-Y. Kao (2004). Fast Universalization of Investment Strategies. *SIAM Journal on Computing* 34(1), 1-22.
- [70] N. Murphy & T. Gebbie (2019). Learning the population dynamics of technical trading strategies. <https://arxiv.org/abs/1903.02228>.
- [71] N. Murphy (2019). Learning Technical Trading. <https://github.com/NJ-Murphy/Learning-Technical-Trading.git>.

# 11 Appendices

## A Data Related Appendices

### A.1 JSE TOP 40 Sector Constituents: Daily Data

The daily data is sourced from Thomson Reuters. The companies and their associated Reuters Instrument Code (RIC) of the three major sectors in the JSE Top 40 are [66]:

#### Resources (JSE-RESI - J210)

Anglo American Platinum Ltd (AMSJ.J), Anglo American PLC (AGLJ.J), AngloGold Ashanti Ltd (ANGJ.J), BHP Billiton PLC (BILJ.J), Mondi Ltd (MNDJ.J), Mondi PLC (MNPJ.J), Sasol Ltd (SOLJ.J).

#### Industrials: JSE-INDI (J211)

Aspen Pharmacare Holdings Ltd (APNJ.J), Bidvest Group Ltd (BVTJ.J), British American Tobacco PLC (BTIJ.J), Compagnie Financiere Richemont SA (CFRJ.J), Capital & Counties Properties PLC (CCOJ.J), Growthpoint Properties Ltd (GRTJ.J), Intu Properties PLC (ITUJ.J), Mediclinic International Ltd (MEIJ.J), MTN Group Ltd (MTNJ.J), Naspers Ltd (NPNJ.J), Remgro Ltd (REMJ.J), Redifine Properties Ltd (RDFJ.J), SABMiller PLC (SABJ.J), Shoprite Holdings Ltd (SHPJ.J), Steinhoff International Holdings (SNHJ.J), Tiger Brands Ltd (TBSJ.J), Vodacom Group Ltd (VODJ.J), Woolworths Holdings Ltd (WHLJ.J), Mr Price Group Ltd (MRPJ.J), Netcare Ltd (NTCJ.J).

#### Financials: JSE-FINI (J212)

Discovery Holdings Ltd (DSYJ.J), Firststrand Ltd (FSRJ.J), Investec Ltd (INLJ.J), Investec PLC (INPJ.J), Nedbank Group Ltd (NEDJ.J), Old Mutual PLC (OMLJ.J), RMB Holdings Ltd (RMHJ.J), Rand Merchant Investment Holdings Ltd (RMIJ.J), Sanlam Ltd (SLMJ.J), Standard Bank Group Ltd (SBKJ.J), Brait SE (BATJ.J), Barclays Africa Group Ltd (BGAJ.J), Capitec Bank Holdings Ltd (CPIJ.J), Fortress REIT Ltd (B) (FFBJ.J), Fortress REIT Ltd (A) (FFAJ.J), Reinet Investments SCA (REIJ.J).

### A.2 JSE TOP 40 Sector Constituents: Intraday Data

Below are 30 of the JSE Top 40 stocks as of the 30 June 2018. All the tick and daily data from 01-01-2018 to 30-06-2018 is sourced from Bloomberg.

#### Resources: JSE-RESI (J210)

Anglo American PLC (AGL:SJ), AngloGold Ashanti Ltd (ANG:SJ), African Rainbow Minerals Ltd (ARI:SJ), BHP Billiton PLC (BIL:SJ), Exxaro Resources Ltd (EXX:SJ), Impala Platinum Holdings Ltd (IMP:SJ), Mondi Ltd (MND:SJ), Sasol Ltd (SOL:SJ).

#### Industrials: JSE-INDI (J211)

Aspen Pharmacare Holdings Ltd (APN:SJ), Bidvest Group Ltd (BVT:SJ), British American Tobacco PLC (BTI:SJ), Compagnie Financiere Richemont SA (CFR:SJ), Capital & Counties Properties PLC (CCO:SJ), Growthpoint Properties Ltd (GRT:SJ), Intu Properties PLC (ITU:SJ), MTN Group Ltd (MTN:SJ), Naspers Ltd (NPN:SJ), Tiger Brands Ltd (TBSJ.J), Shoprite Holdings Ltd (SHP:SJ), Vodacom Group Ltd

(VOD:SJ), Woolworths Holdings Ltd (WHL:SJ).

**Financials: JSE-FINI (J212)**

Discovery Holdings Ltd (DSY:SJ), Firststrand Ltd (FSR:SJ), Investec Ltd (INL:SJ), Nedbank Group Ltd (NED:SJ), RMB Holdings Ltd (RMH:SJ), Sanlam Ltd (SLM:SJ), Standard Bank Group Ltd (SBK:SJ), Barclays Africa Group Ltd (BGA:SJ).

**B Work flow for Learning Class**

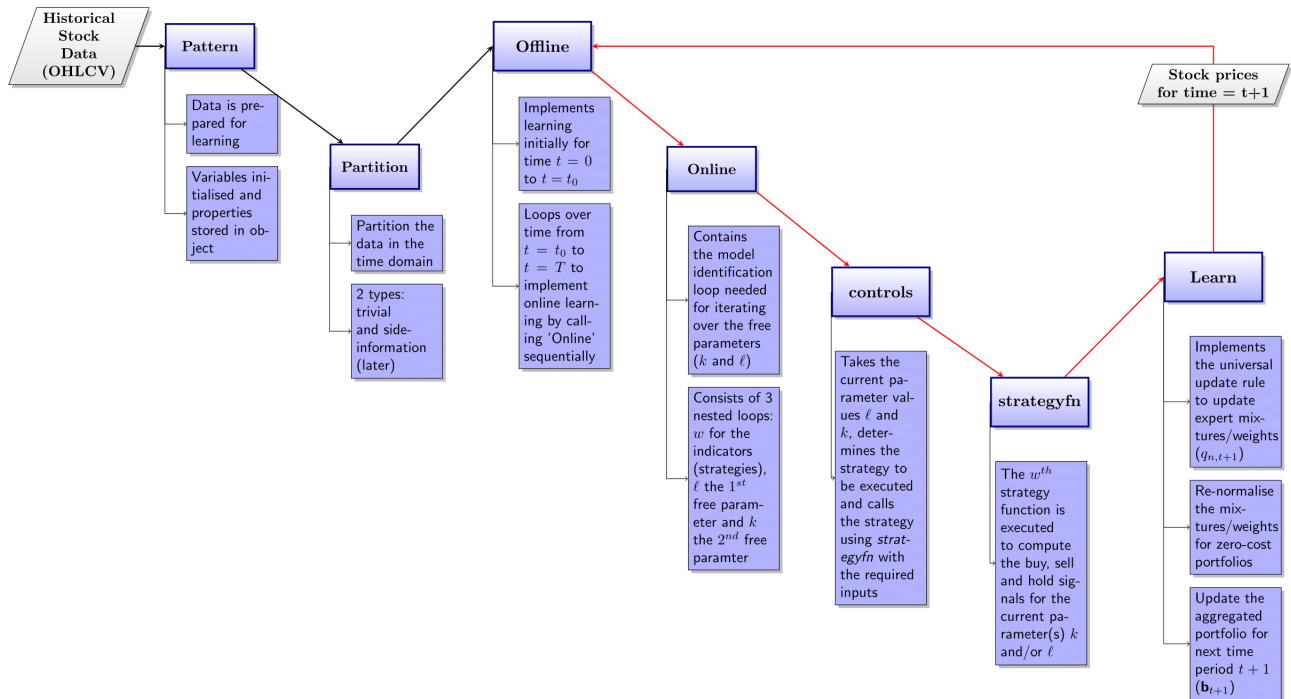


Figure 22: State flow diagram for the MATLAB learning class