

# Deep Reinforcement Learning and Convex Mean-Variance Optimisation for Portfolio Management

---

Ruan Pretorius

*Supervisor:*  
Prof. Terence van Zyl



**WITS**  
UNIVERSITY

A research report submitted in partial fulfilment of the requirements for the degree  
of Master of Science in the field of e-Science

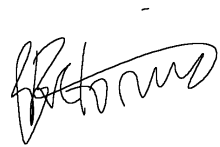
in the

School of Computer Science and Applied Mathematics  
University of the Witwatersrand, Johannesburg

29 May 2022

# Declaration

I, Ruan Pretorius, declare that this research report is my own, unaided work. It is being submitted for the degree of Master of Science in the field of e-Science at the University of the Witwatersrand, Johannesburg. It has not been submitted for any degree or examination at any other university.

A handwritten signature in black ink, appearing to read 'Ruan Pretorius', written in a cursive style.

Ruan Pretorius

29 May 2022

## *Abstract*

Traditional portfolio management methods can incorporate specific investor preferences but rely on accurate forecasts of asset returns and covariances. Reinforcement learning (RL) methods do not rely on these explicit forecasts and are better suited for multi-stage decision processes. To address limitations of the evaluated research, experiments were conducted on three markets in different economies with different overall trends. By incorporating specific investor preferences into the proposed RL models' reward functions, a more comprehensive comparison could be made to traditional methods in risk-return space. Transaction costs were also modelled more realistically by including non-linear changes introduced by market volatility and trading volume. The results of this study suggest that there can be an advantage to using RL methods compared to traditional convex mean-variance optimisation methods under certain market conditions. The proposed RL models could significantly outperform traditional single-period optimisation (SPO) and multi-period optimisation (MPO) models in upward trending markets, but only up to specific risk limits. In sideways trending markets, the performance of SPO and MPO models could be closely matched by the proposed RL models for the majority of the excess risk range tested. The specific market conditions under which these models could outperform each other highlight the importance of a more comprehensive comparison of Pareto optimal frontiers in risk-return space. These frontiers give investors a more granular view of which models might provide better performance for their specific risk tolerance or return targets.

# Acknowledgements

This research would not have been possible without the guidance of my supervisor, Prof. Terence van Zyl. The support of the DSI-NICIS National e-Science Postgraduate Teaching and Training Platform (NEPTTP) towards this research is also hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NEPTTP.

# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Abbreviations</b>	<b>viii</b>
<b>List of Publications</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.1.1 Portfolio Management . . . . .	1
1.1.2 Traditional Methods of Portfolio Management . . . . .	2
1.1.3 Reinforcement Learning . . . . .	3
1.1.4 Reinforcement Learning for Portfolio Management . . . . .	3
1.2 Literature Review . . . . .	4
1.2.1 Reinforcement Learning Methods for Portfolio Management . . . . .	4
1.2.2 Current Limitations . . . . .	7
1.3 Problem Statement . . . . .	9
1.4 Research Question . . . . .	9
1.5 Research Aims and Objectives . . . . .	10
1.5.1 Research Aims . . . . .	10
1.5.2 Objectives . . . . .	10
1.6 Assumptions and Limitations . . . . .	11

1.7	Overview . . . . .	11
<b>2</b>	<b>Research Methodology</b>	<b>12</b>
2.1	Research Design . . . . .	12
2.2	Methods . . . . .	12
2.2.1	Portfolio Performance Measures . . . . .	12
2.2.2	Traditional Mean-Variance Optimisation Baselines . . . . .	14
2.2.3	State-of-the-art RL Models for Comparison . . . . .	17
2.2.4	Proposed RL Model (FRONTIER) . . . . .	18
2.3	Data . . . . .	23
2.3.1	Data Collection . . . . .	23
2.3.2	Data Processing . . . . .	23
2.4	Analysis . . . . .	25
2.5	Procedure . . . . .	26
2.6	Software, Libraries and Hardware . . . . .	26
2.7	Ethical Considerations . . . . .	27
<b>3</b>	<b>Results and Discussion</b>	<b>28</b>
3.1	Market Conditions . . . . .	28
3.2	Traditional Mean-Variance Optimisation Methods . . . . .	30
3.3	Reinforcement Learning Methods . . . . .	32
3.4	Transaction Cost Models . . . . .	34
3.5	Reinforcement Learning vs. Traditional Mean-Variance Optimisation Methods . . . . .	35
3.6	Equally Weighted Method . . . . .	38
3.7	Summary . . . . .	38
<b>4</b>	<b>Conclusions and Future Work</b>	<b>39</b>
4.1	Conclusions . . . . .	39
4.2	Future Work . . . . .	41
	<b>References</b>	<b>42</b>

# List of Figures

1.1	Agent-environment interaction diagram . . . . .	3
2.1	Log-returns policy network diagram . . . . .	19
2.2	Forecast-only policy network diagram . . . . .	20
2.3	All-inputs policy network diagram . . . . .	22
3.1	Price of market indices during train and test periods . . . . .	29
3.2	Pareto optimal frontiers of traditional mean-variance optimisation models . . . . .	31
3.3	Pareto optimal frontiers of reinforcement learning models . . . . .	33
3.4	Pareto optimal frontiers of reinforcement learning models and traditional mean-variance optimisation models . . . . .	37

# List of Tables

2.1	Market data description . . . . .	23
3.1	Model performance difference when using different transaction cost models . . . . .	35



# List of Abbreviations

<b>A2C</b>	<b>Advantage Actor Critic</b>
<b>ADX</b>	<b>Average Directional Index</b>
<b>CCI</b>	<b>Commodity Channel Index</b>
<b>CNN</b>	<b>Convolutional Neural Network</b>
<b>DDPG</b>	<b>Deep Deterministic Policy Gradient</b>
<b>DJIA</b>	<b>Dow Jones Industrial Average</b>
<b>DQN</b>	<b>Deep Q-Network</b>
<b>DRQN</b>	<b>Deep Recurrent Q-Network</b>
<b>DSRQN</b>	<b>Deep Soft Recurrent Q-Network</b>
<b>ETF</b>	<b>Exchange Traded Fund</b>
<b>FRONTIER</b>	<b>reinforcement learning portfolio manager with investor preferences</b>
<b>LSTM</b>	<b>Long Short-Term Memory</b>
<b>MACD</b>	<b>Moving Average Convergence Divergence</b>
<b>MA-FDRNN</b>	<b>Multi-Asset Fuzzy Deep Recurrent Neural Network</b>
<b>MDP</b>	<b>Markov Decision Process</b>
<b>MPO</b>	<b>Multi-Period Optimisation</b>
<b>OSBL</b>	<b>Online Stochastic Batch Learning</b>
<b>PPO</b>	<b>Proximal Policy Optimisation</b>
<b>RL</b>	<b>Reinforcement Learning</b>
<b>RNN</b>	<b>Recurrent Neural Network</b>
<b>RSI</b>	<b>Relative Strength Index</b>
<b>ReLU</b>	<b>Rectified Linear Unit</b>
<b>SPO</b>	<b>Single-Period Optimisation</b>
<b>TD</b>	<b>Temporal Difference</b>

# List of Publications

The following publication came from this study. It is currently under review for publication to the Journal of IEEE Transactions on Artificial Intelligence.

1. Ruan Pretorius and Terence van Zyl. *Deep Reinforcement Learning and Convex Mean-Variance Optimisation for Portfolio Management*. TechRxiv (2022), Preprint. URL: <https://doi.org/10.36227/techrxiv.19165745.v1>

# Chapter 1

## Introduction

This introductory chapter provides the definitions and descriptions of core concepts necessary to frame and motivate this study. This chapter also includes the problem statement, research question, scope, and limitations of this study.

### 1.1 Background

#### 1.1.1 Portfolio Management

The term *portfolio management* (also called portfolio optimisation or asset allocation) refers to the process of allocating portions of some total amount of wealth to different assets in an asset universe. Modern portfolio theory and the concept of portfolio management was first introduced by Harry Markowitz in the 1950s [1], [2]. There are typically more than one time-step considered within an investment period where the allocation of assets can be adjusted or rebalanced as more recent information becomes available. This rebalancing is done in order to keep the portfolio performance in line with the investor's preferences in terms of expected returns or risk [3].

The main reason for an investor to distribute their wealth between a variety of assets in a portfolio, as opposed to investing only in a single asset, is to mitigate risk through diversification [1], [2]. Zivot (2017) has proven that the volatility (risk) of a long-only portfolio of assets is always lower than that of a single asset, given the assets in the portfolio are not perfectly correlated. Here, a long-only portfolio means that no assets are borrowed, i.e, there are only positive positions in assets [4].

The most common assets considered in portfolio management are stocks, foreign exchange, cryptocurrencies, or exchange traded funds (ETFs) [5], [6], [7], [8]. In this study, the only assets considered were stocks and cash.

### 1.1.2 Traditional Methods of Portfolio Management

Harry Markowitz's framework of *mean-variance* portfolio optimisation is widely used in industry and academia. It allows an investor to optimally allocate their wealth between assets in order to balance the risk-reward trade-off according to their risk appetite. For example, an investor might decide on a maximum amount of risk that they are willing to tolerate. Markowitz's mean-variance method allows them to choose the optimal weighting of assets in a portfolio to maximise their expected returns without that level of risk being exceeded. When these optimal portfolios are computed for a range of different risk values and plotted in risk-return space, they form a curve called the *efficient frontier*. This efficient frontier can be used to select the optimal portfolio with maximum expected returns for a given risk value [2], [4].

One of the main limitations of Markowitz's mean-variance method is that it only considers one time-step into the future [3]. In other words, the allocation of assets is done in a way that only takes a single portfolio rebalancing period into account. Ideally, the impact on future decisions should also be taken into account.

Another limitation of traditional mean-variance methods is that they rely on accurate forecasts of returns and covariances between the assets that make up the portfolio. Unrealistic assumptions of normally distributed returns are also sometimes made which can lead to large drawdowns (losses) that often cannot be tolerated by investors [8].

In 2017, a study by Boyd et al. showed how this single-period optimisation (SPO) approach can be extended to a multi-period optimisation (MPO) implementation [9]. Both the SPO and MPO versions of Boyd et al. (2017) are used as traditional mean-variance benchmark methods in this study. These methods are convex optimisation problems that aim to maximise expected returns in the presence of transaction costs and risk. These methods are explained in more detail in section [2.2.2](#).

### 1.1.3 Reinforcement Learning

Sutton and Barto (2018) [10] gave the following description of reinforcement learning (RL). It describes both a type of problem and a class of solutions that work well to solve that problem. It applies to problems and solutions that can be formulated in terms of an agent that interacts with and receives feedback from its environment. This interaction is often framed as a Markov decision process (MDP) which has four essential components. These are sensations in the form of observable states of the environment, actions that can be executed by the agent, a transition function that determines the next state based on the action taken in the current state, and reward signals that guide the agent towards a goal related to an ideal state of the environment. The agent is never given explicit instructions of which actions to take but instead learns which actions are best to take in different circumstances through exploration of its environment. Figure 1.1 shows a visual representation of this agent-environment interaction.

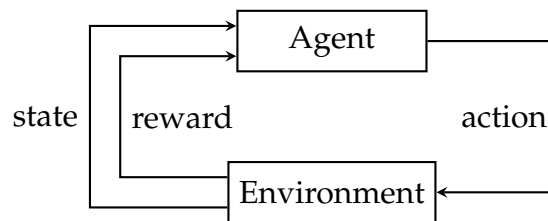


FIGURE 1.1: Interaction between agent and environment formulated as a Markov decision process.

By formulating sequential decision processes in terms of MDPs, future states, actions, and rewards depend on past ones. Therefore, the MDP formulation captures the need for a trade-off between immediate and delayed rewards. RL methods are able to solve these types of problems because the goal of the agent is to maximise expected future rewards [10].

### 1.1.4 Reinforcement Learning for Portfolio Management

To emphasise the applicability of RL methods to the portfolio management task, the relevant MDP components are identified here. The role of the agent is that of a decision-maker that has to choose the optimal weighting between assets in a

portfolio (action). The environment in this case is the market, which supplies the observable states (such as historical prices) and reward signals (such as realised returns after transaction costs) as feedback.

When transaction costs are considered, the portfolio management problem becomes a multi-stage decision-making process where future states and decisions are impacted by past decisions [11]. In this setting, immediate rewards are not the only important aspect to consider, but also the possible negative impact of current decisions on the ability to receive rewards in the future. RL methods are particularly well suited for this type of problem since they aim to maximise the accumulation of rewards in the long term even if it means acting sub-optimally in the short term [10]. This capacity to make long-term decisions is the main reason for choosing RL methods as the subject of investigation in the portfolio management task.

## 1.2 Literature Review

This section contains a summary of the methods and main results of related previous research on RL methods for portfolio management. This review is given to highlight the four main limitations identified in previous research. Note that some portfolio performance measures like returns, volatility, drawdowns, and Sharpe ratio are mentioned in this section. These terms are defined later in section 2.2.1.

### 1.2.1 Reinforcement Learning Methods for Portfolio Management

Meng and Khushi (2019) conducted a survey of 29 studies on RL used for stock and forex trading [12]. This survey included studies where many different methods of RL were used, including on-policy and off-policy methods. Most of the studies from this survey used historical prices and returns exclusively as the observable states. As for the reward signal, most studies either used the Sharpe ratio or returns based on historical data. One of the major limitations pointed out in this survey, was that the majority of studies did not include transaction costs [12].

Jiang et al. (2017) conducted a study on using deep RL for portfolio management on a 12-asset cryptocurrency portfolio [13]. They considered a rebalancing period of 30 minutes and included transaction costs with a commission rate of 0.25% (proportional to the portfolio weight changes). The states of their models included the

current portfolio weighting and the open, high, low, and close prices of each 30-minute trading window for the past 50 windows (one day and one hour). The actions of their models were continuous portfolio weights between zero and one (long-only). Reward signals were chosen to be the natural logarithm of the realised returns (after transaction cost) between rebalancing periods. These rewards were averaged between all time-steps in an episode before updating the policy. This update method equates to having a discount factor of one. All models in this study were model-free policy gradient methods, where the policy was approximated by using three different neural network architectures. These architectures included a convolutional neural network (CNN), a basic recurrent neural network (RNN), and a long short-term memory (LSTM) neural network. Even though the task was framed in an episodic manner, online learning was achieved through a proposed online stochastic batch learning (OSBL) scheme. This involved selecting a time-window starting from some starting time in the test set. The starting time was randomly selected from historical values with a geometric distribution so that more recent events were more likely to be selected. Whenever a new data point became available, it was added to the training set, making it an online learning method. Three backtests were conducted between 2016 and 2017 by training the models on approximately two years of data and testing their performance on approximately two months of data. The best model was the CNN which produced a Sharpe ratio of 0.087, a maximum drawdown of 22%, and 400% returns in a 50-day period [13].

Another study, conducted by Filos (2019), looked at RL for portfolio management [14]. They used different types of model-free RL methods on a 12-asset portfolio consisting of cash and stocks from the S&P 500 universe. The states of their models also included the current portfolio weights and the natural logarithm of realised returns (after transaction costs) for varying time-window sizes. Transaction costs in the form of broker commission and spreads were considered together as 0.2% of the change in portfolio weights. Similar to the study of Jiang et al. (2017), the model actions were long-only portfolio weights bounded between zero and one, and the reward signal was either the logarithm of the realised returns or the Sharpe ratio. In this study, the models were trained on a data set spanning five years between 2000 and 2005. Thereafter, testing was done on a data set spanning a 13-year period from 2005 to 2018. Since the models were configured to use online learning, they were able to still update their policies during the testing phase. However,

these policy updates were dependent on the model architecture. Their temporal difference (TD) method called Deep Soft Recurrent Q-network (DSRQN) was an adaptation of both the Deep Q-network (DQN) of Mnih et al. (2015) [15] and the deep recurrent Q-network (DRQN) of Hausknecht and Stone (2015) [16], which were only able to handle discrete action spaces. DSRQN used a combination of a CNN and RNN to approximate the action-value function and was extended to be able to handle pseudo-continuous action spaces by introducing a softmax output layer. Since this was a TD-method, its policy was updated after every time-step. This DSRQN method was able to produce 256% returns with a Sharpe ratio of 2.4 and a maximum drawdown of 85% over the 13-year test period [14].

Another method used in the study by Filos (2019), was a Monte Carlo policy gradient method called REINFORCE [14]. It used the same states, actions, and reward signals as the DSRQN method, but only updated its policy at the end of each episode by averaging the rewards over the time-steps, using different discount factors between zero and one. In this method, the policy was approximated directly using a combination of both a CNN and a RNN, similar to the DSRQN method. The REINFORCE method was able to produce returns of 325% with a Sharpe ratio of 3.02 and a maximum drawdown of 63.5% over the 13-year test period [14].

A recent study by Yang et al. (2020) looked at an ensemble of three different model-free deep reinforcement actor-critic methods for portfolio management [17]. The three methods considered were Proximal Policy Optimisation (PPO), Advantage Actor Critic (A2C), and Deep Deterministic Policy Gradient (DDPG). The portfolio consisted of the 30 stocks of the Dow Jones Industrial Average (DJIA) index. The action-space for these models was slightly different to that of the studies mentioned before. In this case, actions were integer amounts of shares to buy or sell for each stock. The state-space of these models was also slightly different, consisting of the latest close prices, the amount of stocks owned, the current balance, and some technical indicators based on a window of historical stock prices. The technical indicators included the Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Commodity Channel Index (CCI), and Average Directional Index (ADX). During the training phase, each model was trained on daily data of nearly seven years between 2009 and 2015. The last three months of the data from 2015 was used as a validation set to determine model hyperparameter values. The testing phase was done in an online fashion on four and a half years of



data between 2016 and mid-2020. During this period, the Sharpe ratio was recorded for each model based on a trailing window of three months. The top performing model was then selected for execution of trades. This way, the active model was dynamically selected based on recent performance. This ensemble method was able to produce 13% annual returns with an annual volatility of 9.7%, an annual Sharpe ratio of 1.3, and a maximum drawdown of 9.7% over a period of four and a half years [17].

Skeepers et al. (2021) developed a novel model-free RL model for portfolio management called Multi-Asset Fuzzy Deep Recurrent Neural Network (MA-FDRNN) [18]. Their implementation used historical prices passed through a fuzzy layer where each asset was mapped to a fuzzy representation cluster. The output of the fuzzy layer was then passed to a RNN layer that performed feature learning. Finally, the output of the RNN layer represented the allocation changes to the assets in the portfolio. The state space of MA-FDRNN was continuous and included historical stock prices and the amount of cash in the portfolio. The reward function was set up to be the portfolio log-returns. The action space was also continuous and consisted of the changes in the amount of wealth allocated to each asset of the portfolio. Contrary to the other studies mentioned thus far, MA-FDRNN had three major differences. Firstly, it did not incorporate any transaction costs. Secondly, it allowed negative positions in assets that represented short trades. Thirdly, it was tested on five different markets that included upward, sideways, and downward trending conditions. Because of its ability to perform short trades, MA-FDRNN was able to outperform other RL methods (including DDPG and PPO) in downward and sideways trending markets. However, it was not able to outperform the other RL models in upward trending markets. The average daily log-returns achieved by MA-FDRNN was between 5.51% and 7.84% for upward trending markets; 4.91% for the sideways trending market; and 0.65% for the downward trending market [18].

### 1.2.2 Current Limitations

The preceding literature demonstrates that researchers have previously investigated numerous approaches to portfolio management with RL methods. These previous studies include different combinations of on-policy and off-policy learning with

TD methods and Monte Carlo methods for value-function estimation, policy estimation, and actor-critic methods. However, this study identified the following four limitations in the evaluated research.

Firstly, most of the RL methods were not risk-aware due to the reward signal being related exclusively to returns (except for those methods that used the Sharpe ratio as a reward function). The large drawdowns during test periods confirm the risk-ignorant nature of some of these models. Most of the evaluated research aimed to outperform the market or traditional methods only in terms of returns. This maximum return aim does not cater to the needs of different investors with different risk tolerances and return targets. For example, more risk-averse investors that are not necessarily aiming for maximising returns irrespective of risk might want to incorporate some limit to the risk they are willing to assume by investing in these risky assets.

The second and closely related limitation of the evaluated research was that they only compared single portfolio outcomes. Apart from only being of interest to investors with particular risk and return goals, this gives a limited view of the model's performance in the risk-return space. In other words, the evaluated studies produced only a single performance point in the risk-return space instead of an entire efficient frontier like traditional mean-variance optimisation methods do.

Thirdly, in the aforementioned research, the transaction cost was limited in that it was a linear function of bid-ask spreads and broker commission. This limitation neglects the inclusion of a second, non-linear term that changes as a function of market volatility and trading volume, which is a more realistic characterisation of transaction costs [9].

Finally, most of the above research only assessed the performance of models on a single market. Therefore, they were limited in that their results might not apply to markets in different economies or markets with different characteristics and conditions as far as overall market behaviour is concerned. For example, if experiments were only conducted on markets with an overall upward trend in value, models that invested heavily in risky assets would have performed very well. However, these same models could have performed very poorly in markets with an overall downward trend in value where investing more in a risk-free asset would yield better performance.

### 1.3 Problem Statement

Many different methods have been used in the past for portfolio management. Traditional mean-variance optimisation methods are able to incorporate specific investor preferences and can produce an efficient frontier in risk-returns space. However, they can under-perform due to their reliance on accurate forecasts of asset returns and covariances.

Other approaches to portfolio management like RL methods do not rely on accurate forecasts or assumptions on return distributions. Moreover, they are suited for multi-stage decision processes by considering the implications of their actions on their ability to produce future rewards.

RL methods have been successfully applied to portfolio management in previous research. However, these studies were limited in four ways. Firstly, they aimed at outperforming the market or traditional methods only in terms of returns without taking specific investor preferences into account. Secondly, model performance was described by single points in risk-return space, not by many points forming an efficient frontier. Thirdly, transaction costs were modelled in a limited way, neglecting non-linear changes introduced by market volatility and trading volume. Finally, mostly single markets were used to assess model performance, leading to limited applicability to other markets and market conditions.

### 1.4 Research Question

The research question of this study is as follows. To what extent can traditional mean-variance optimisation methods of portfolio management be out-performed by using RL methods that take specific investor preferences into account in different market conditions?

The portfolio management task was framed as a multi-stage decision making process by incorporating non-linear transaction costs and allowing multiple rebalancing opportunities during a given investment period. Investor preferences were considered by introducing risk-aversion and trade-aversion parameters to the RL methods' reward functions in order to suit the preferences and risk appetite of a range of different investors. The portfolio management performance of all methods

in this study was assessed on three markets with different overall market trends using commonly used metrics such as returns, volatility, and Sharpe ratio.

## 1.5 Research Aims and Objectives

### 1.5.1 Research Aims

The main aim of this study was to assess the extent to which traditional mean-variance optimisation methods of portfolio management can be out-performed by using RL methods that take specific investor preferences into account in different market conditions. This was done by addressing the four limitations identified in previous research so that a more comprehensive comparison could be made between traditional mean-variance optimisation methods and RL methods in risk-return space and to assess the extent to which performance is affected by different market conditions.

### 1.5.2 Objectives

To answer the research question, the following objectives were considered:

1. Identify three markets where the overall price has an upward, downward, and sideways trend. For each of these markets, select a set of stocks to be included in the portfolio.
2. Build the SPO and MPO versions of traditional mean-variance optimisation models for a baseline comparison.
3. Build the proposed RL models with reward functions that allow for incorporation of different investor preferences.
4. Build state-of-the-art RL models (PPO, DDPG, and A2C) to benchmark the performance of the proposed RL models against.
5. Train and backtest (simulate) all models with non-linear transaction costs on the three different markets which will produce efficient frontiers in risk-return space.
6. Assess the portfolio management performance of all models in terms of returns, volatility (risk), and Sharpe ratio.

## 1.6 Assumptions and Limitations

One limitation of this study is that the portfolio management performance of different methods was assessed by conducting backtests (simulations) only, i.e., no live-trading was executed. This limitation involves three necessary assumptions as in the studies by Filos (2019) [14] and Jiang et al. (2017) [13]. These assumptions are *sufficient liquidity*, *no market impact*, and *no slippage*. These assumptions are all valid if the volume of the assets traded in the portfolio is high enough [19].

The sufficient liquidity assumption means it is assumed that every stock can be converted into cash almost instantly with no loss in value. The assumption of no market impact assumes that when stocks are traded, it does not impact their value. Finally, the no slippage assumption assumes that trades are executed immediately, allowing no time for their value to change between the time a trade is requested and executed [14], [13].

Although the experiments of this study will be repeated on three markets with different overall price trends, it does not mean that the results will hold for all such markets with similar trends. In other words, this study contributes a step towards generality but might not be fully general yet.

## 1.7 Overview

The rest of this document is structured as follows. Chapter 2 describes the methodology that was followed in order to gather evidence used to support answers to the research question. Chapter 3 contains an exposition of the results obtained as well as a critical discussion of them. Finally, Chapter 4 concludes with a summary of this study and proposed future work.

## Chapter 2

# Research Methodology

### 2.1 Research Design

This study made use of experimental and statistical methods to assess the extent to which traditional mean-variance optimisation methods of portfolio management can be out-performed using RL methods. The proposed RL models took specific investor preferences into account and were assessed in different market conditions. The simulated portfolio management performance of all models was measured using common metrics such as returns, volatility, and Sharpe ratio.

### 2.2 Methods

#### 2.2.1 Portfolio Performance Measures

Portfolios are composed of a collection of assets whose prices change over time. Therefore, the portfolio's value also changes over time. The changes in returns and risk exposure can be quantified and depend on the prices of the underlying assets in the portfolio as well as the weighting assigned to each one. In order to understand and appreciate the methods and results of this study, this section presents an overview of portfolio performance measures to provide context.

The portfolio performance measures used in this study were returns, volatility, and Sharpe ratio. These measures are commonly found in literature and are described here using notation similar to that of Boyd et al. (2017) [9].

For a portfolio of  $n$  assets (stocks) and cash, the price vector at any time-step  $t$  is denoted  $p_t \in \mathbb{R}_+^{n+1}$  (the subscript indicates non-negative values). From this price vector, a vector of returns  $r_t \in \mathbb{R}^{n+1}$  can be constructed. The return of asset  $i$  is the

percentage price change between two successive time-steps:

$$(r_t)_i = \frac{(p_t)_i - (p_{t-1})_i}{(p_{t-1})_i}, \quad i = 1, \dots, n+1 \quad (2.1)$$

Alternatively, the log-return is also sometimes used and is calculated as follows:

$$\log \left( \frac{(p_t)_i}{(p_{t-1})_i} \right) = \log (1 + (r_t)_i), \quad i = 1, \dots, n+1 \quad (2.2)$$

At any time-step  $t$ , the proportion of the total portfolio assigned to each asset is captured by the portfolio weight vector  $w_t \in \mathbb{R}^{n+1}$ . It is useful to also define the change in weight  $z_t \in \mathbb{R}^{n+1}$  between successive time-steps for each asset  $i$  as follows:

$$(z_t)_i = (w_t)_i - (w_{t-1})_i, \quad i = 1, \dots, n+1 \quad (2.3)$$

In this study, as in Boyd et al. (2017) [9], transaction costs were modelled as a unit-less, non-linear function of bid-ask spread, broker commissions, trading volume and market volatility. The total transaction cost  $\phi_t^{\text{trade}}$  at time  $t$  is the sum of transaction costs resulting from trading  $n$  individual assets:

$$\phi_t^{\text{trade}} = \sum_{i=1}^n \left[ a|z_{t,i}| + b\sigma_{t,i} \frac{|z_{t,i}|^{3/2}}{\sqrt{V_{t,i}/v_t}} + cz_{t,i} \right] \quad (2.4)$$

where a double subscript is used to indicate time-step and asset (e.g.  $z_{t,i} = (z_t)_i$ ). Here,  $a \in \mathbb{R}$  is used to capture half of the bid-ask spread expressed as a fraction of asset price. Any broker commission can also be incorporated in  $a$ . Here,  $b \in \mathbb{R}$  has units of inverse-dollars and is used to scale the second term. The recent volatility (standard deviation of returns) of asset  $i$  at time  $t$  is captured by  $\sigma_{t,i} \in \mathbb{R}$  in dollars.  $V_{t,i} \in \mathbb{R}$  is the volume traded in dollars of asset  $i$  at time  $t$ , which is scaled by the total portfolio value  $v_t$  in order to keep the denominator unit-less. Finally,  $c \in \mathbb{R}$  can be used to create asymmetry in the transaction cost when buying and selling do not cost the same.

The realised return  $R_t^p$  (after transaction costs) of the entire portfolio of  $n+1$  assets at time  $t$  can then be calculated as follows:

$$R_t^p = r_t^T w_t + r_t^T z_t - \phi_t^{\text{trade}} \quad (2.5)$$

When considering some investment period from  $t = 1, \dots, T$ , it is common to calculate the average realised returns  $\overline{R^p}$  for that period as follows:

$$\overline{R^p} = \frac{1}{T} \sum_{t=1}^T R_t^p \quad (2.6)$$

The risk associated with holding a portfolio can be quantified by the standard deviation of portfolio returns  $\sigma^p$ . This value is commonly referred to as volatility and can be calculated using the following equation:

$$\sigma^p = \left[ \frac{1}{T} \sum_{t=1}^T (R_t^p - \overline{R^p})^2 \right]^{1/2} \quad (2.7)$$

It is often useful to compare the portfolio risk and return to some benchmark. This benchmark can either be another portfolio or a single asset. The terms *excess return* and *excess risk* are used to refer to the risk and return obtained in excess of a risk-free asset like cash. The excess return  $R_t^e$  of a portfolio is defined as:

$$R_t^e = R_t^p - (r_t)_{n+1} \quad (2.8)$$

where  $(r_t)_{n+1}$  refers to the return of the  $(n + 1)^{\text{th}}$  asset of the portfolio (the risk-free or cash asset). The excess risk  $\sigma^e$  can be calculated as the standard deviation of excess returns.

The Sharpe ratio  $SR$  is used to quantify the risk-adjusted excess returns of a portfolio as follows:

$$SR = \frac{\overline{R^e}}{\sigma^e} \quad (2.9)$$

With these portfolio performance metrics established, some traditional methods of portfolio management can be considered.

## 2.2.2 Traditional Mean-Variance Optimisation Baselines

As mentioned in the introduction, SPO and MPO extensions of Markowitz's mean-variance optimisation, developed by Boyd et al. (2017) [9], were used as baseline portfolio management methods in this study. These methods are convex optimisation problems formulated to enable single and multi-period optimisation.



The SPO version arrives at the optimal portfolio weight vector  $w_{t+1} = w_t + z_t$  for the next time-step by solving for  $z_t$  in the optimisation problem:

$$\begin{aligned} \max \quad & \hat{r}_t^T(w_t + z_t) - \gamma^{\text{trade}} \hat{\phi}_t^{\text{trade}}(z_t) - \gamma^{\text{risk}} \psi_t(w_t + z_t) \\ \text{s.t.} \quad & z_t \in \mathcal{Z}_t, \quad w_t + z_t \in \mathcal{W}_t, \quad \mathbf{1}^T z_t = 0 \end{aligned} \quad (2.10)$$

where  $\psi_t(w_t + z_t)$  describes the risk function which is an estimate of the variance of portfolio returns. In this study, the constraints  $\mathcal{Z}_t \in \mathbb{R}$  and  $\mathcal{W}_t \in [0, 1]$  are set to ensure long-only trades are made. A caret is placed over some variables to emphasise that they are estimates (since they are not known at time  $t$ ). Here,  $\gamma^{\text{trade}}$  and  $\gamma^{\text{risk}}$  scale the trading and risk aversion respectively and can be changed to capture the preferences of different investors. As the trading aversion increases, trading will be discouraged and transaction costs will decrease. The risk aversion parameter is used to discourage holding portfolios with high volatility. In this study the quadratic risk function is used and is described as follows:

$$\psi_t(w_t + z_t) = (w_t + z_t)^T \hat{\Sigma}_t (w_t + z_t) \quad (2.11)$$

where  $\hat{\Sigma}_t \in \mathbb{R}^{(n+1) \times (n+1)}$  is an estimate of the return covariance matrix of all assets in the portfolio.

Similarly, the objective of the MPO version is to choose the change in portfolio vector that maximises expected realised returns while minimising risk and transaction costs. However, the MPO version extends the SPO framework to take multiple time-steps into account. This MPO constitutes a trading plan for  $H$  time-steps into the future, producing a sequence of portfolio vector changes,  $z_t, z_{t+1}, \dots, z_{t+H-1}$  by solving:

$$\begin{aligned} \max \quad & \sum_{\tau=t}^{t+H-1} \left[ \hat{r}_{\tau|t}^T(w_\tau + z_\tau) - \gamma^{\text{trade}} \hat{\phi}_\tau^{\text{trade}}(z_\tau) \right. \\ & \left. - \gamma^{\text{risk}} \psi_\tau(w_\tau + z_\tau) \right] \\ \text{s.t.} \quad & z_t \in \mathcal{Z}_t, \quad w_t + z_t \in \mathcal{W}_t, \quad \mathbf{1}^T z_t = 0, \\ & w_{\tau+1} = w_\tau + z_\tau, \quad \tau = t, \dots, t+H-1 \end{aligned} \quad (2.12)$$

where  $\hat{r}_{\tau|t}$  is used to denote the return forecast of  $r_\tau$  made at time  $t$ , using only

information available at time  $t$ . In both the SPO and MPO versions,  $w_t$  is known since it is the current portfolio weight vector.

Following the method of Boyd et al. (2017), the MPO version used in this study was a two-period optimisation where  $H = 2$ . The same values were also used for  $\gamma^{\text{risk}}$  and  $\gamma^{\text{trade}}$  (with some extension on both extreme ends) so that all 504 pairwise combinations of the following sets were tested to capture a wide range of investor preferences:

$$\begin{aligned}\gamma^{\text{risk}} &\in \{0.1, 0.178, 0.316, 0.562, 1, 2, 3, 6, 10, 18, 32, 56, \\ &\quad 100, 178, 316, 562, 1000, 2000, 5000, 10000, 20000\} \\ \gamma^{\text{trade}} &\in \{0.1, 0.5, 1, 2, 3, 4, 5, 5.5, 6, 6.5, 7, 7.5, 8, 9, 10, 11, \\ &\quad 12, 15, 20, 30, 45, 60, 100, 200\}\end{aligned}$$

As in Boyd et al. (2017), the asset returns covariance matrix  $\hat{\Sigma}$  was estimated using a factor model [9]. This involved, firstly, calculating the actual returns covariance matrix of the trailing two-year period  $\Sigma^{\text{past}}$ . Secondly, an eigendecomposition of this covariance matrix of past returns was performed as follows:

$$\Sigma^{\text{past}} = \sum_{i=1}^n \lambda_i q_i q_i^T \quad (2.13)$$

where the eigenvalues  $\lambda_i$  were in descending order. Thirdly, these were used to construct the covariance matrix of factor returns  $\Sigma^f = \text{diag}(\lambda_1, \dots, \lambda_k)$ , the factor loading matrix  $F = [q_1, \dots, q_k]$ , and the idiosyncratic risk matrix:

$$D = \sum_{i=k+1}^n \lambda_i \text{diag}(q_i) \text{diag}(q_i) \quad (2.14)$$

Finally, these components were used to construct a factor model to estimate the asset returns covariance matrix as follows:

$$\hat{\Sigma} = F \Sigma^f F^T + D \quad (2.15)$$

where  $\hat{\Sigma} \in \mathbb{R}^{(n+1) \times (n+1)}$ ,  $F \in \mathbb{R}^{(n+1) \times k}$ ,  $\Sigma^f \in \mathbb{R}^{k \times k}$ , and  $D \in \mathbb{R}^{(n+1) \times (n+1)}$ . In this study,  $k = 15$  factors were used. Using this factor model enabled faster simulation times on the order of  $\mathcal{O}(nk^2)$  as opposed to  $\mathcal{O}(n^3)$  when not using a factor model [9].

Another model used in this study as a benchmark was the *equally weighted* model. This model did not rely on any return forecasts or other estimations of the underlying assets in the portfolio but instead applied a single, simple rule to make allocation decisions. The equally weighted model started off fully invested in non-cash assets, where each asset was assigned an equal weight, i.e.,  $w_t = [1/n, 1/n, \dots, 0]$ . At the end of each day, the equally weighted model rebalanced its holdings to re-establish this equal weighting.

### 2.2.3 State-of-the-art RL Models for Comparison

This study used three existing state-of-the-art actor-critic RL models as a baseline comparison for the proposed RL models. These were Advantage Actor-Critic (A2C), Proximal Policy Optimisation (PPO), and Deep Deterministic Policy Gradient (DDPG). These are the same three models implemented in the study by Yang et al. (2020) [17] in their ensemble model. The only modification made to these models were to change the linear transaction cost function so that it captures the non-linear dynamics described in Equation (2.4). This modification was made to allow for a valid comparison between all the models in this study. Apart from this modification, all other aspects of the original models remained unchanged. This replication of the research ensured that these models were as close to their original forms as possible.

The critic network in A2C approximates what is called an advantage function in addition to the usual value function. This advantage function enables A2C to assess both the quality of actions and how good they can be, which leads to a more robust policy with lower variance. The experiments of Yang et al. (2020) suggest that A2C performs better in down-trending markets with high volatility compared to both PPO and DDPG [17].

DDPG combines Q-learning and policy gradient methods and has a policy network that deterministically maps states to actions. DDPG has a replay buffer that stores the state transitions and their corresponding actions and rewards during training. Based on batches of these transitions drawn from the replay buffer, the model parameters are updated [17].

Finally, PPO introduces a clipping term to its loss function that discourages large policy changes when model parameters are updated, leading to more stable policy

learning. According to the experimental results of Yang et al. (2020), both DDPG and PPO perform better in sideways and upward-trending market conditions compared to A2C, with PPO slightly outperforming DDPG [17].

The continuous state space for DDPG, PPO, and A2C included technical indicators containing the historical price, trend, volume information, and the current portfolio holdings to allow them to account for transaction costs. The action space was also continuous for all three models, consisting of a normalised vector that specifies the number of shares to buy or sell for each asset in the portfolio. As in the original study, the reward function of these models was set to maximise realised portfolio returns [17]. All hyperparameter values for these models were set to the default values used in the code library accompanying the original study [20].

#### 2.2.4 Proposed RL Model (FRONTIER)

The proposed RL model was named FRONTIER (reinforcement learning portfolio manager with investor preferences) due to its capacity to take different investor preferences into account and its output creating a Pareto optimal frontier in risk-return space (explained below). FRONTIER is a Monte Carlo policy gradient method based on the REINFORCE algorithm [10]. The policy of this model was represented with a neural network (policy network). Three different policy networks were examined in this study in order to determine the limitations and advantages of each one. For all policy networks, the observable state input variables or features were supplied as an input at the beginning of each time step. Although different intermediate layers were used for the different policy networks, they all had an output layer with a softmax activation function and one neuron for each asset in the portfolio. This final output layer thus produced the portfolio weight vector  $w_{t+1}$  to be used in the next time step. Therefore, FRONTIER had a continuous action space in  $\mathbb{R}_+^{n+1}$  (the subscript indicates non-negative values, which implies long-only policies).

The first variation of the policy network, seen in Figure 2.1, was called the *log-returns* policy. For this policy network, a window of historical log-returns was passed as input along with some additional features. The log-returns window was made up of the daily log-returns for each asset in the portfolio, spanning a length of  $L$  time-steps, calculated using Equation (2.2). A convolutional filter of

size  $(n + 1) \times \tau$  was then passed over this log-returns window to produce  $k$  feature maps. This convolutional neural network (CNN) block automatically detects patterns in individual assets and between assets (such as covariance).

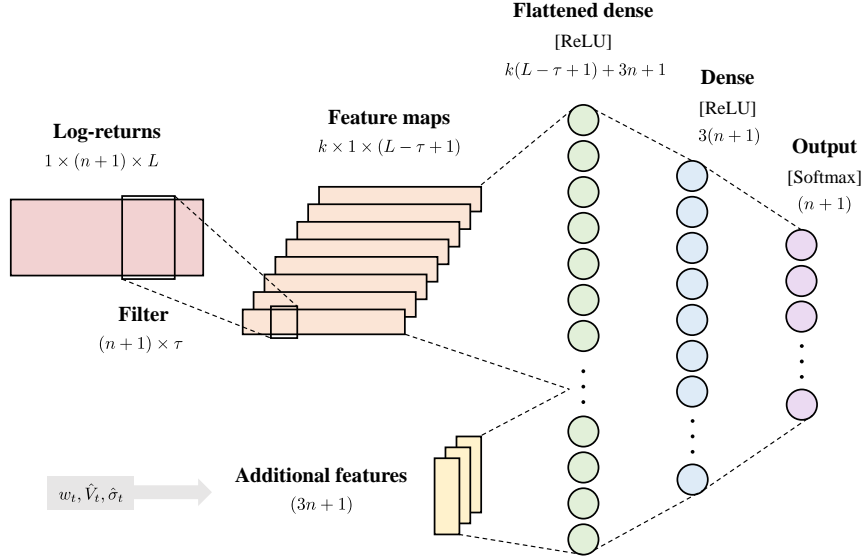


FIGURE 2.1: The log-returns policy network of the FRONTIER model with only historical log-returns and additional features as state inputs. A convolutional filter was passed over the log-returns window to produce several feature maps. The additional feature vectors gave the model the capacity to take transaction costs into account. The feature maps and additional features were flattened and connected with fully-connected layers to produce the next portfolio weight vector as an output.

In addition to the log-returns, three additional feature vectors were also given as state inputs to give the model the capacity to take transaction costs into account (note these features correspond to factors and terms in the transaction cost Equation (2.4)). These three features were the current portfolio weight vector  $w_t \in \mathbb{R}_+^{n+1}$ , the estimated volume traded for each non-cash asset  $\hat{V}_t \in \mathbb{R}_+^n$ , and the estimated volatility for each non-cash asset  $\hat{\sigma}_t \in \mathbb{R}_+^n$ . Therefore, the state space of FRONTIER was also continuous.

As seen in the policy network diagrams, the feature maps from the CNN block were flattened along with the additional three feature vectors to produce the next fully connected layer consisting of  $k(L - \tau + 1) + 3n + 1$  neurons. This layer was

followed by another fully connected layer consisting of  $3(n + 1)$  neurons. These fully connected layers had the rectified linear unit (ReLU) activation function and led to the final fully connected output layer as described earlier.

The hyperparameters of the policy network were selected somewhat arbitrarily with  $L = 20$  to represent 20 working days or one month and  $\tau = 5$  to represent a single working week. The amount of feature maps produced by the CNN block was chosen to be  $k = (n + 1)$ . These and other policy network hyperparameters were not fine-tuned or optimised further in any way, partially due to time/computation constraints and partially to avoid over-fitting to any specific markets or asset portfolios. As part of future work, these hyperparameters can be fine-tuned to determine the effect they have on overall performance. The values given here can be considered as a starting point or baseline.

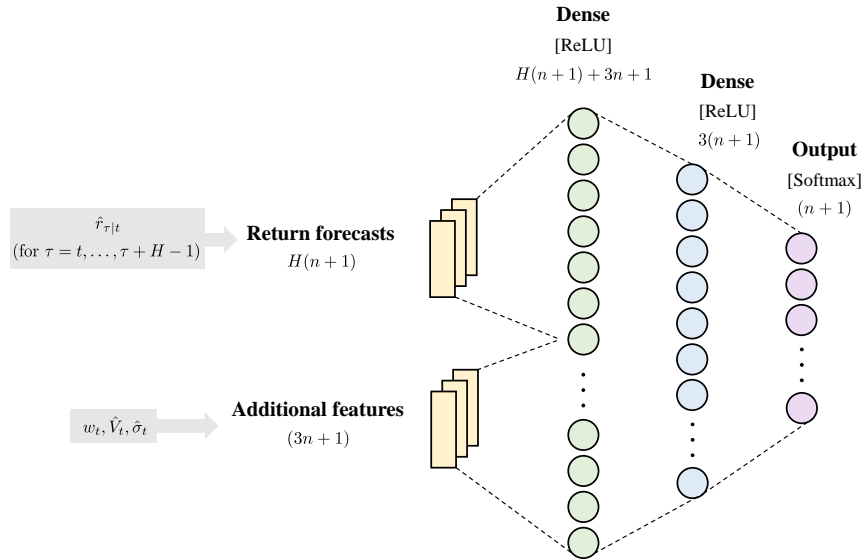


FIGURE 2.2: The forecast-only policy network of the FRONTIER model with only return forecasts and additional features as state inputs. The forecasts were explicit returns forecasts for all assets in the portfolio for  $H$  steps into the future. The additional feature vectors gave the model the capacity to take transaction costs into account. The return forecasts and additional features were flattened and connected with fully-connected layers to produce the next portfolio weight vector as an output.

The second policy network version was called the *forecast-only* policy network

(see diagram in Figure 2.2). This network had the same inputs as the log-returns policy network, with the log-returns window replaced by explicit returns forecasts of all assets in the portfolio. These forecasts took the form of  $H$  vectors of  $\hat{r}_{\tau|t} \in \mathbb{R}^{n+1}$ , where each vector represented the returns forecast of a separate time-step. In this study, whenever return forecasts were given as state inputs to the FRONTIER model, a value of  $H = 2$  was used to ensure a valid comparison to the MPO could be made (MPO also used  $H = 2$ ). The forecast-only policy network was introduced to isolate the part of the policy network that produced forecasts. This way, the performance of the forecast-only policy could be compared to the log-returns policy to assess the efficacy of the CNN block in producing implicit returns forecasts.

The third and final version, called the *all-inputs* policy network (see diagram in Figure 2.3), was a combination of the previous two policy networks such that it had access to all available state inputs. This version was introduced as a final check to see if any relative performance differences between the log-returns policy and the forecast-only policy were due to independent factors or if extra performance gains could be achieved by allowing access to all state input variables.

The FRONTIER model was trained in an episodic fashion where an episode of fixed length (30 days) was drawn from the training data according to a uniform distribution. The episode length was arbitrarily selected and can be fine-tuned as part of future work. The policy network parameters were then updated based on that episode's expected discounted future rewards. The discounted future rewards  $G_t$  for each time step  $t$  of the episode were calculated as follows [10]:

$$G_t = \sum_{k=t+1}^T \gamma^{k-t-1} R_k \quad (2.16)$$

where  $\gamma = 0.99$  is the future reward discount rate and  $R_t$  is the immediate reward given at time  $t$ . This immediate reward was chosen to take the same form as the quantity maximised by the SPO and MPO mean-variance optimisation algorithms of Boyd et al. (2017) [9] and is given in Equation (2.17). This expression was included and given the same form for two main reasons. Firstly, because no other existing RL methods take investor preferences into account and secondly, so that a valid comparison of FRONTIER could be made to SPO and MPO models for a

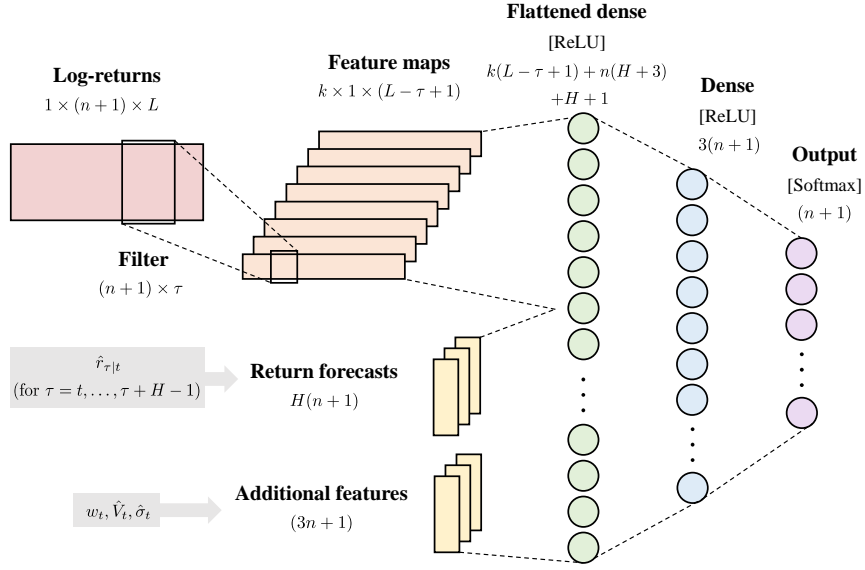


FIGURE 2.3: The all-inputs policy network of the FRONTIER model with all state inputs (historical log-returns, explicit return forecasts, and additional features). A convolutional filter was passed over the log-returns window to produce several feature maps. Explicit returns forecasts were also given for all assets in the portfolio for  $H$  steps into the future. The additional feature vectors gave the model the capacity to take transaction costs into account. The feature maps, explicit return forecasts, and additional features were flattened and connected with fully-connected layers to produce the next portfolio weight vector as an output.

range of investor preferences.

$$R_t = r_t^T w_{t+1} - \gamma^{\text{trade}} \phi_t^{\text{trade}} (w_{t+1} - w_t) - \gamma^{\text{risk}} \psi_t (w_{t+1}) \quad (2.17)$$

Finally, to get the expected discounted future rewards (from which the model parameters are updated during training), the average discounted future rewards were taken for each episode.



## 2.3 Data

### 2.3.1 Data Collection

All data used in this study came from Yahoo Finance [21] and Qunadl [22]. Data downloaded from Yahoo Finance included the daily open, high, low, and close prices of all assets, including their daily volume traded. Quandl was used to obtain the returns of what was considered the cash or risk-free asset in all portfolios. The US Federal three-month treasury bill rate was selected to be the risk-free asset.

The above-mentioned data were obtained for three different markets so that all models could be tested on different market conditions. The main goal was to select three markets: one where the overall market trend was upward; one where the overall trend was downward; and one where the overall trend was stable or sideways. A secondary goal was to select markets where these trend conditions were present for sufficiently long periods so that they could span both training and testing periods. A summary of the three selected markets is given in Table 2.1. The price change and overall trends of these markets can also be seen in Figure 3.1.

TABLE 2.1: Description of data for each market, including the overall trend, training period, testing period, and number of assets used.

Market	Trend	Training	Testing	Assets <sup>a</sup>
Dow 30	Upward	2010-01-01 – 2018-01-01	2018-01-01 – 2020-01-01	30
Nikkei 225	Sideways	2013-05-01 – 2018-01-01	2018-01-01 – 2020-01-01	24
Latin America 40	Downward	2010-03-01 – 2014-12-01	2014-12-01 – 2016-01-01	24

<sup>a</sup>Final number of assets selected after filtering and processing.

### 2.3.2 Data Processing

For consistency and valid comparison, whenever models relied on estimates in this study, the same method was used in estimating values. This section describes these estimation methods and other data processing methods used in this study.

Whenever asset return forecasts were made, the method of Boyd et al. (2017) [9] was followed. This forecasting method entailed perturbing realised future returns

by adding noise with zero mean to simulate return forecasts. This forecasting method was employed in an attempt to keep the focus of this study on what is possible once return forecasts were already obtained. More specifically, the return forecasts were obtained for all non-cash assets using:

$$\hat{r}_t = \alpha (r_t + \epsilon_t) \quad (2.18)$$

where  $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$  was zero-mean normally distributed noise and  $\alpha$  was selected to minimise the mean squared error between the realised returns  $r_t$  and the noisy “forecast”  $\hat{r}_t$ , which equates to a scaling value of  $\alpha = \sigma_r^2 / (\sigma_r^2 + \sigma_\epsilon^2)$ , where  $\sigma_r^2$  is the variance of  $r_t$ . A noise value of  $\sigma_\epsilon^2 = 0.02$  was used along with a typical value of  $\sigma_r^2 = 0.005$ . This noise addition relates to a standard deviation in the forecast of 10 times that of the returns, which in turn relates to a return forecast accuracy on the higher end of what is expected in practice [9]. It is important to note that these simulated forecasts were used only for SPO and MPO and did not affect the implicit forecasts of FRONTIER from the log-returns state input. This allowed for a realistic and hard benchmark for FRONTIER to be compared to.

In order to estimate the remaining values used in SPO and MPO for estimating the transaction cost and risk, the following steps were followed, again using the same method as Boyd et al. (2017) [9]. Estimates for return volatility  $\hat{\sigma}_t$  and volume traded  $\hat{V}_t$  were calculated for each asset by taking a trailing 10-day moving average using the following equations:

$$\hat{V}_t = \frac{1}{10} \sum_{\tau=1}^{10} V_{t-\tau} \quad (2.19)$$

$$\hat{\sigma}_t = \frac{1}{10} \sum_{\tau=1}^{10} \sigma_{t-\tau} \quad (2.20)$$

For the additional features used as state inputs to the RL models, the same methods were used to calculate estimates for volume traded  $\hat{V}_t$  and returns volatility  $\hat{\sigma}_t$  as in Equation (2.19) and Equation (2.20), respectively. Before these additional state inputs were passed to the policy network, they were normalised to be on the same order of magnitude as  $w_t$  (between 0.0 and 1.0). This normalisation was done for each asset by dividing all  $\hat{V}_t$  and  $\hat{\sigma}_t$  values by their respective averages over the 30 days preceding the start of the training period specified in Table 2.1.

In order to calculate the realised transaction costs, the realised volume traded  $V_t$  was used along with the daily asset returns volatility  $\sigma_t$ . Since the collected data was on a daily frequency,  $\sigma_t$  was approximated as in Boyd et al. (2017) [9] using:

$$\sigma_t = \left| \log \left( p_t^{\text{open}} \right) - \log \left( p_t^{\text{close}} \right) \right| \quad (2.21)$$

where  $p_t^{\text{open}}$  and  $p_t^{\text{close}}$  are the daily open and close prices of the asset in question, respectively.

## 2.4 Analysis

In multi-objective optimisation problems with several conflicting objectives, a set of viable solutions can be found where none of the solutions are dominated by others. These non-dominated solutions are all optimal solutions with trade-offs in at least one objective. Together, these non-dominated solutions form a multi-dimensional *Pareto optimal frontier* [23]. In this study, the experimentally derived set of optimal portfolios was referred to as the Pareto optimal frontier which is similar to the efficient frontier described by Markowitz [1], [2].

To compare the performance of all models against each other, they were all back-tested on the testing portion of the data set for each market as specified in Table 2.1. This test portion of the data set was kept from all models during the training phase to assess the out-of-sample performance of all models.

The FRONTIER models were trained and tested, along with the SPO and MPO models for the entire investor preference spectrum spanned by the 504 combinations of risk-aversion  $\gamma^{\text{risk}}$  and trade-aversion  $\gamma^{\text{trade}}$  parameters. In addition to this, due to the stochastic nature of the RL model training process, all FRONTIER models were trained and tested on the same data set 10 times using different seed values for their pseudo-random number generating processes. This repetition was done for two reasons. Firstly, to assess the average performance of each model and secondly, to quantify the variance of the experimental performances obtained.

Each model's performance was assessed in terms of excess return  $\bar{R}^e$  and excess risk  $\sigma^e$  for a specific investor preference parameterised by  $\gamma^{\text{risk}}$  and  $\gamma^{\text{trade}}$ . The excess risk and return were obtained using Equation (2.8) and Equation (2.9). This computation was done for all 504 parameter combinations expressing a range of

investor preferences. Once all these points in risk-return space were obtained, the non-dominated set was determined by choosing all points for which the maximum excess return was obtained without increasing excess risk. This non-dominated set constituted the Pareto optimal frontier, which was a set of optimal portfolios to hold during the test period.

Since the experiment was repeated 10 times for each FRONTIER model, the calculation of the Pareto frontier was done 10 times as well. The mean Pareto frontier was then calculated along with a t-test to determine the 95% confidence interval of this mean Pareto frontier.

## 2.5 Procedure

The following procedure was followed in order to obtain the results of this study:

1. Compile data set (download, preprocess, split for training and testing) for each market.
2. Build the SPO and MPO versions of traditional mean-variance optimisation models for a baseline comparison.
3. Build proposed RL models with reward functions that allow for incorporation of different investor preferences.
4. Build state-of-the-art RL models (PPO, DDPG, and A2C) to benchmark the performance of proposed RL models against.
5. Train and backtest (simulate) all models with non-linear transaction costs on the three different markets to produce Pareto optimal frontiers in risk-return space.
6. Assess the portfolio management performance of all models in terms of returns, volatility (risk), and Sharpe ratio.

## 2.6 Software, Libraries and Hardware

All models used in this study, including traditional mean-variance optimisation models and RL models, were programmed in Python 3. These models were trained

and tested on a computer with 64 CPU cores and 252GB of RAM using the Ubuntu 21.04 (x86-64) operating system.

The proposed RL (FRONTIER) models were programmed in Python 3.7, using the Tensorflow 2.4 library [24] for creating the different policy networks.

The SPO and MPO models were created in Python 3.6 and used the *cvxpy* library [25] for convex optimisation. These models were implemented using the *cvx-portfolio* library developed by Boyd et al. [26]. In particular, an updated version, modified by Razvan Oprisor, was used [27]. The equally weighted model was also implemented using this updated *cvxportfolio* library.

The state-of-the-art RL models (A2C, PPO, and DDPG) were implemented using the *FinRL* library developed by Liu et al. [28]. Minor modifications were made to simulate more realistic non-linear transaction costs. This library was created using Python 3.6 and used RL algorithms from the *Stable-Baselines3* package [29].

## 2.7 Ethical Considerations

No human or animal participants were used during the course of this study for the purpose of gathering information. All necessary information was gathered through literature survey, computer simulation, and quantitative databases of stock/index price data that was in the public domain. All the collected data were stored on a password protected computer for the duration of the study. For these reasons, no ethical risk was posed to any person as a result of this study being conducted.

## Chapter 3

# Results and Discussion

To fully appreciate the results obtained by all the models in this study, it is helpful to consider the overall market trends during the training and testing periods of the data sets used. Figure 3.1 shows the daily closing price of the market indices from which the stocks were selected. Note that all prices were normalised and made dimensionless for easy comparison; this was done through dividing by the initial price. The shaded grey area on each chart indicates the testing period, whereas the non-shaded area indicates the training period.

### 3.1 Market Conditions

The upward trending market (Dow 30) shows a strong upward trend during the training and testing periods, with a price increase of just over 141% during the training period and just over 15% during the testing period. The sideways trending market (Nikkei 225) shows a slight upward trend during the training period with a price change of just over 58% with a minor increase of under 1% during the testing period. Finally, the downward trending market (Latin America 40) shows a strong downward trend with a change of just over 25% during the training period, continuing downward with a price decrease of just over 43% during the test period.

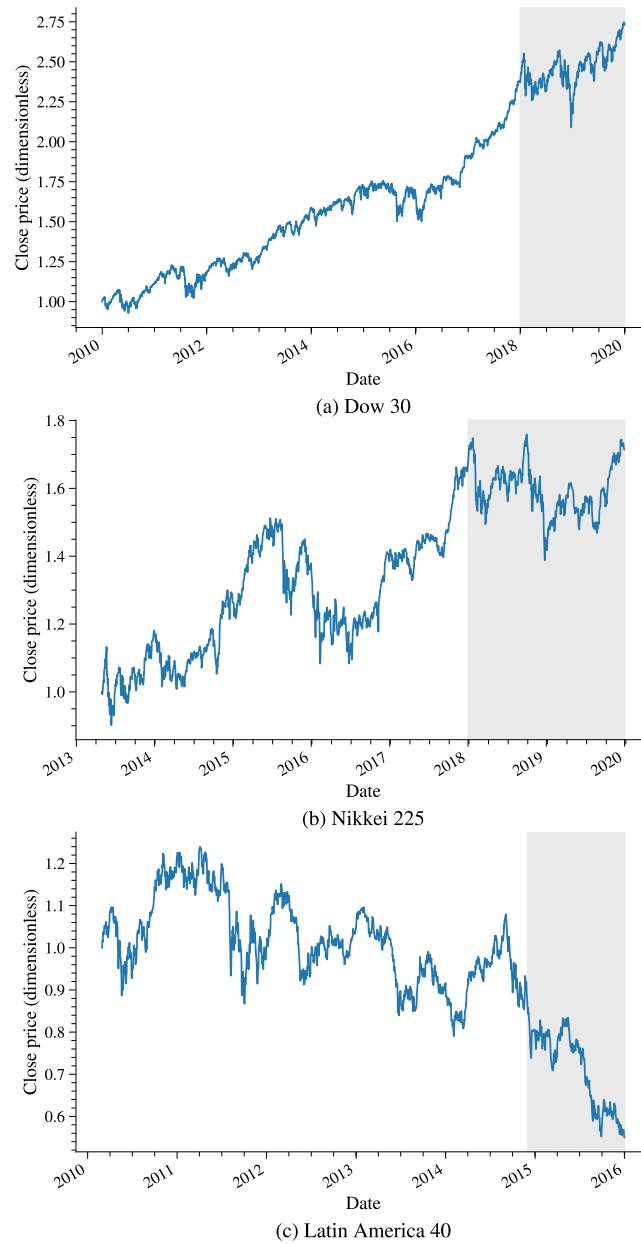


FIGURE 3.1: Dimensionless price of all three markets used in this study. The shaded grey area indicates the test period of the data and the non-shaded white area indicates the training period of the data. The price changes are as follows for each period: Dow 30: +141.06% (train) and +15.33% (test); Nikkei 225: +58.48% (train) and +0.49% (test); Latin America 40: -25.36% (train) and -43.11% (test).

## 3.2 Traditional Mean-Variance Optimisation Methods

Figure 3.2 shows the performance of SPO and MPO on each of the three markets in isolation. These are the Pareto optimal frontiers obtained by simulating back-tests over the test period for all 504 pairwise combinations of risk-aversion  $\gamma^{\text{risk}}$  and trade-aversion  $\gamma^{\text{trade}}$ . This figure shows, as expected that MPO slightly outperforms SPO on average in all three markets. This outperformance is likely due to the extra time-step taken into account by the MPO model during its multi-period optimisation. This result is also found in the study of Boyd et al.(2017) on the S&P 500 market.

It is important to note the difference in scale in Figure 3.2(c) of the plot for the Latin America 40 market. Both SPO and MPO produced only negative excess returns over a very small excess risk range. Upon closer inspection of the portfolio weight vectors  $w_t$  these models produced, it is clear that both SPO and MPO almost exclusively invested in the risk-free asset, only to shift to small positions in riskier stocks for very short periods as their risk-aversion decreased. This shift explains the slight variation of excess risk and return that produced these Pareto frontiers. However, both SPO and MPO behaved differently in the Dow 30 and Nikkei 225 markets. In these two markets, a more gradual portfolio weight change was made, spanning the whole spectrum from fully invested in the risk-free asset to large positions in risky stocks as the risk-aversion parameter was lowered. Larger trades (changes in  $w_t$ ) were also observed as the trade-aversion parameter was reduced.



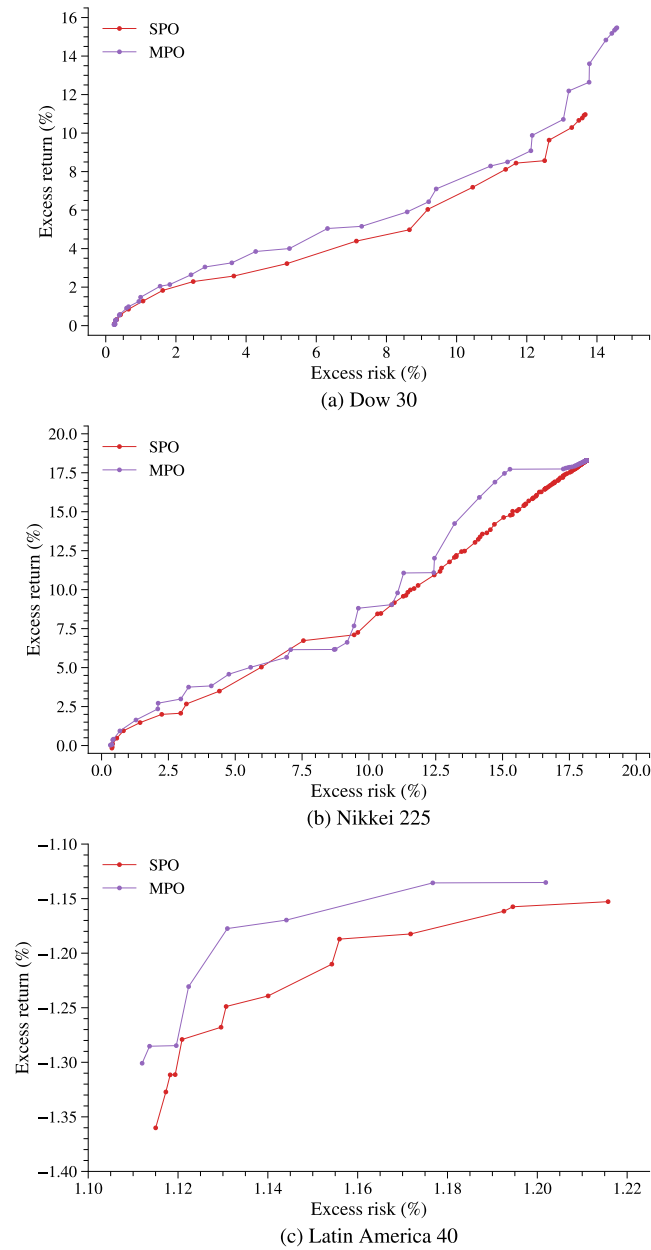


FIGURE 3.2: Pareto optimal frontiers in risk-return space of SPO and MPO models on all three markets produced by parameter sweep of all 504 pairwise combinations of risk and trade-aversion. MPO outperforms SPO on average in all three markets. Note: significant scale difference on (c) Latin America 40.

### 3.3 Reinforcement Learning Methods

Figure 3.3 shows the mean Pareto frontiers (along with their 95% confidence intervals) produced by FRONTIER when using different policy network architectures. For the Dow 30 market, all three policy networks performed very similarly for the entire excess risk and return ranges. On the Nikkei 225 market, the performance of all three policy networks was also similar, with the mean frontiers of the log-return and forecast-only networks slightly outperforming the all-inputs network for the most part and the log-return network achieving the most excess returns towards the high-risk end. On the Latin America 40 market, all three policy networks were also very closely matched with the all-inputs version producing the highest excess returns towards the low-risk end (see Figure 3.4 for closer inspection on the low-risk end).

However, considering the overlapping confidence intervals for the vast majority of the frontiers in all three markets, none of these policy networks could significantly outperform any of the others consistently. This result suggests that the all-inputs policy network did not have an added advantage even with all state inputs at its disposal. It also suggests that the log-returns policy network implicitly produced asset return forecasts with the same degree of accuracy as the perturbed realised return forecasts.

Looking at the state-of-the-art RL models' performance in Figure 3.3, the same qualitative performance of the study by Yang et al. (2020) is also found in the Dow 30 market in that PPO and DDPG both outperformed A2C for upward trending market conditions. In the Nikkei 225 market, these three models performed more similarly, with PPO and A2C having almost the same performance and DDPG producing slightly less excess returns for a similar risk value. These RL methods do not appear in the Latin America 40 market plot due to their large negative excess returns that are off the chart area (-28.4% for DDPG; -29.4% for PPO; and -35.5% for A2C).

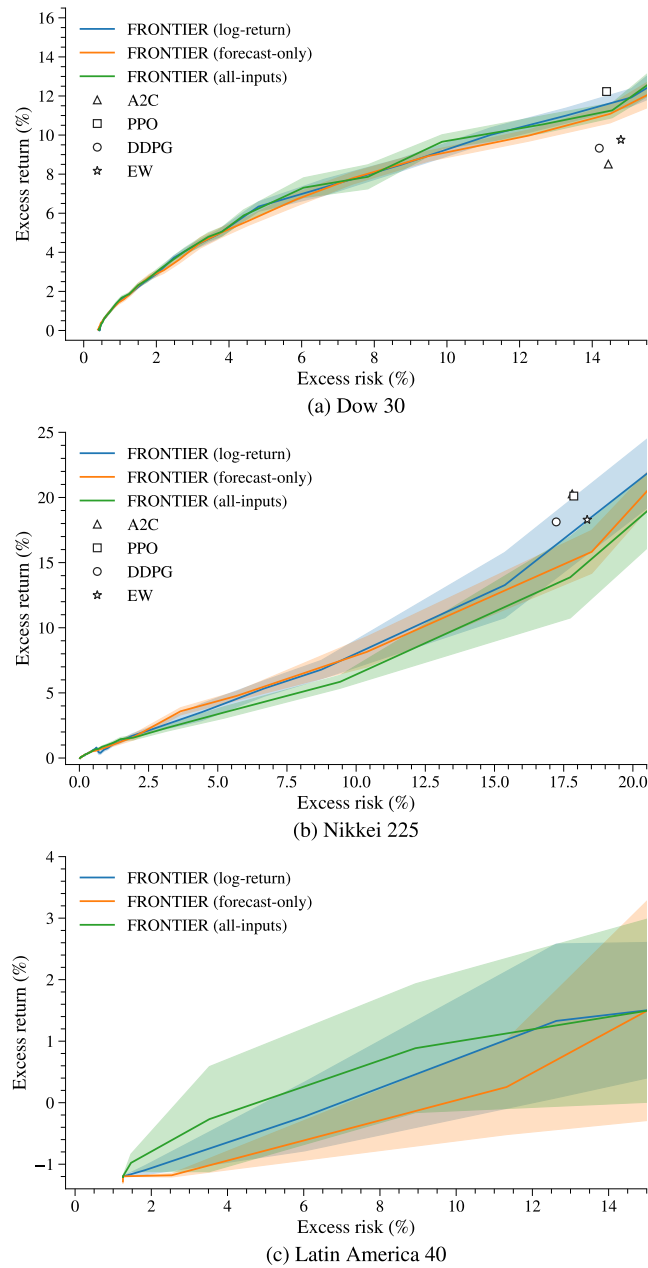


FIGURE 3.3: Pareto optimal frontiers (mean with 95% confidence interval) in risk-return space of FRONTIER models with different policy networks on all three markets. Frontiers were produced by parameter sweep of all 504 pairwise combinations of risk and trade-aversion. Also included are the performances of three state-of-the-art RL models (A2C, PPO, and DDPG) along with the equally weighted (EW) portfolio. Note: A2C, PPO, DDPG, and EW are not shown for (c) Latin America 40 since they produced large negative returns off the chart area.

Figure 3.3 also shows the performance of FRONTIER relative to A2C, PPO, and DDPG. In the Dow 30 market, FRONTIER could outperform both A2C and DDPG, with PPO producing slightly more returns than the upper confidence interval of FRONTIER fitted with a log-returns policy network. For the Nikkei 225 market, there is no significant performance difference between the proposed RL model equipped with a log-returns policy network and A2C, PPO, or DDPG. This result suggests that in markets with an upward trend, FRONTIER outperformed or at least closely matched the performance of state-of-the-art RL models seeking high returns. This result also suggests that in sideways trending markets, FRONTIER (with a log-returns policy network) matched the performance of state-of-the-art RL models seeking high returns.

### 3.4 Transaction Cost Models

In order to assess the effect that the non-linear transaction cost modification had on portfolio management performance, the DDPG, PPO, and A2C models from Yang et al. (2020) [17] were evaluated using the different transaction cost functions. These models were selected because their original versions used linear transaction cost functions. For this comparison, all three models were trained and tested on the Dow 30 market for the same periods seen in Table 2.1. The original non-linear transaction cost function for these models was equivalent to using Equation 2.4 with values of  $a = 0.0005$ ,  $b = 0$ , and  $c = 0$ . These original versions were compared to the modified versions (seen in Figure 3.3(a)) with non-linear transaction cost functions ( $a = 0.0005$ ,  $b = 1$ , and  $c = 0$ ). The excess returns, excess risk, and Sharpe ratio produced by these models can be seen in Table 3.1. For all three models, the excess risk achieved was similar when using the two different transaction cost functions (0.5% average difference). However, there was a slight difference in the excess returns (1.4% on average). PPO managed to produce slightly more excess returns using the non-linear transaction cost function, whereas DDPG and A2C both produced higher excess returns with the linear transaction cost function. PPO also achieved a slightly higher Sharpe ratio with the non-linear transaction cost function whereas DDPG and A2C produced higher values with the linear transaction cost function. This result suggests that the linear transaction cost function might overestimate risk-adjusted returns (Sharpe ratio) for some models like DDPG and

A2C while slightly underestimating them for other models like PPO. Therefore, using the nonlinear transaction cost function (Equation 2.4) can give a more realistic estimation of true performance in terms of transaction costs.

TABLE 3.1: Change in excess returns, excess risk, and Sharpe ratio obtained by DDPG, PPO, and A2C model on the Dow 30 market when using linear and non-linear transaction cost functions.

Model	Transaction cost	Excess return (%)	Excess risk (%)	Sharpe ratio
DDPG	Linear	10.801	14.908	0.724
	Non-linear	9.328	14.194	0.657
PPO	Linear	11.733	14.996	0.782
	Non-linear	12.227	14.395	0.849
A2C	Linear	10.819	14.600	0.741
	Non-linear	8.516	14.442	0.590

### 3.5 Reinforcement Learning vs. Traditional Mean-Variance Optimisation Methods

Thus far, the results address the four limitations identified in the evaluated previous research looking at portfolio management using RL methods. These results provide insight into the performance of RL methods when a wide variety of investor preferences are considered in terms of risk and trade-aversion. These results produce an entire Pareto optimal frontier from which investors can choose their risk and trade-aversion parameters to suit their particular risk and return objectives. Moreover, these model performances are more realistic compared to the aforementioned prior research from a transaction cost perspective. This improvement comes from the inclusion of non-linear changes in the transaction cost introduced by market volatility and trading volume which was taken into account in addition to the linear changes related to bid-ask spreads and broker commission. Finally, the limitation of testing on a single market was also addressed by conducting tests on three markets from different economies with different overall price trends. With these limits addressed, a more comprehensive comparison of traditional mean-variance optimisation methods could be made with RL methods and is considered next.

The performance of FRONTIER models is directly compared to that of the traditional mean-variance optimisation methods in Figure 3.4 for all three markets. In the Dow 30 market, FRONTIER could significantly outperform both SPO and MPO for excess risk values between around 1% and 13%. When taking on excess risk above around 13%, however, MPO produced significantly higher returns. In the Nikkei 225 market, all FRONTIER models produced similar excess returns for the majority of the risk range. In this market, MPO produced significantly higher excess returns compared to FRONTIER models for excess risk values between around 13% and 16%. A direct comparison could not be made in the Latin America 40 market since there was no overlap in the FRONTIER models' Pareto frontiers and those of SPO or MPO. It might be possible to extend the Pareto frontiers of the SPO and MPO models to produce an overlapping area by testing a wider range of risk and trade-aversion parameters. In the parameter sweep tested, lower risk-aversion parameters did lead to points further to the right in this risk-return space. However, they all produced very low (and often negative) returns and were not Pareto optimal. These results suggest that FRONTIER is able to significantly outperform traditional mean-variance optimisation methods like SPO and MPO in upward trending markets up to some excess risk limit (in the case of the Dow 30 market, this limit was around 13%). The results also suggest that in sideways trending markets, the performance of SPO and MPO can be closely matched by FRONTIER for the majority of the excess risk range tested. No conclusions could be drawn on the out-performance of traditional mean-variance optimisation models and FRONTIER in downward trending markets.

Given that the FRONTIER model with forecast-only policy network had the same state inputs as SPO and MPO, the main difference in these models were the temporal aspects of their optimisation algorithms. FRONTIER optimised its reward signal for expected future rewards over a period of 30 days (one episode length), where SPO and MPO only optimised their rewards over a period of one or two days. These results suggest that there is some advantage in using RL methods for portfolio management because of the way they optimise for expected future rewards over more extended periods of time (at least under certain market conditions).

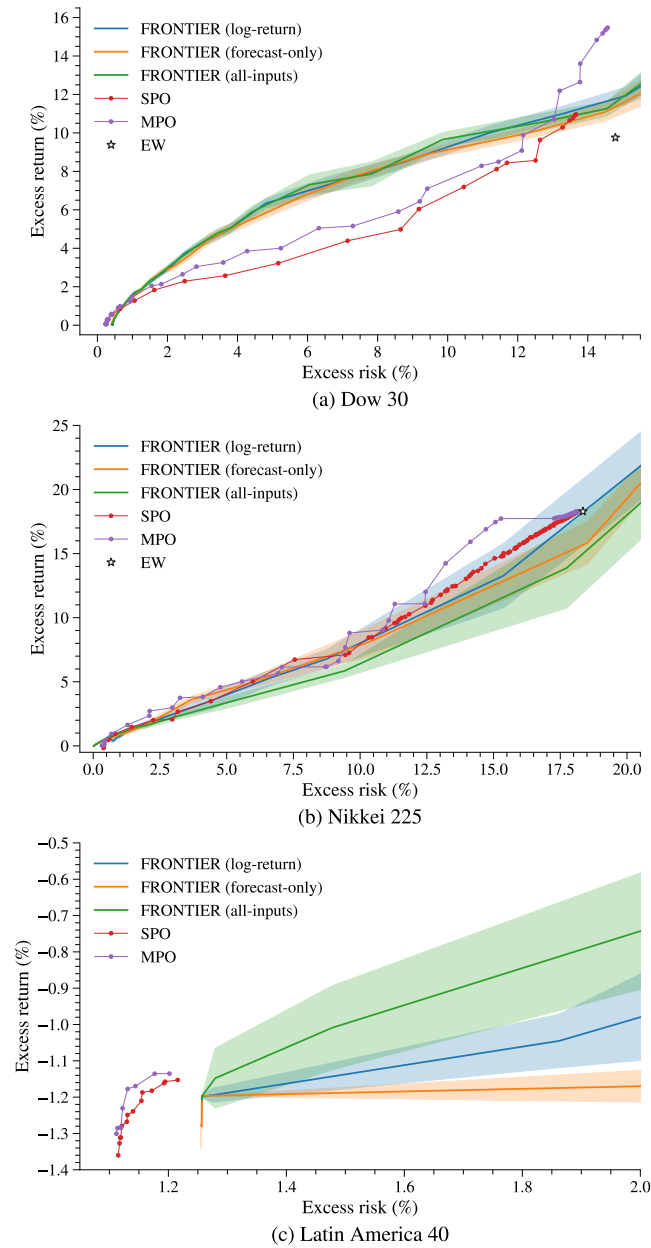


FIGURE 3.4: Direct comparison of Pareto optimal frontiers in risk-return space of FRONTIER models (mean with 95% confidence interval) and convex mean-variance optimisation models (SPO and MPO). All frontiers were produced by the same parameter sweep of all 504 pairwise combinations of risk and trade-aversion. Also included is the equally weighted (EW) portfolio performance for reference.

### 3.6 Equally Weighted Method

Figure 3.4 shows that in the Nikkei 225 market, SPO, MPO, and FRONTIER models produced very similar returns to the equally weighted strategy (abbreviated as EW in plots) around the 18% excess risk mark. Indeed, after inspecting the portfolio weight vectors of both SPO and MPO models, they seem to use a strategy very close to that of the equally weighted model. In the case of FRONTIER models, this seems to be more of a coincidence as they were still predominantly invested in one to three risky assets and cash. For the Dow 30 market, all FRONTIER models and MPO could produce greater returns for a given amount of risk compared to the equally weighted portfolio. Finally, in the Latin America 40 market, even though SPO, MPO, and FRONTIER produced mostly negative excess returns, they did learn to invest almost solely in the risk-free asset for high risk-aversion values. Therefore, SPO, MPO, and FRONTIER arguably outperform the equally weighted strategy, which produced extreme negative excess returns (-29.9%). These results suggest that in upward or downward trending markets, the equally weighted strategy can be outperformed using SPO, MPO, or FRONTIER in terms of returns. It also suggests that it is not possible to significantly outperform the equally weighted strategy in a sideways trending market using either traditional mean-variance optimisation or the RL models from this study.

### 3.7 Summary

The results of this study suggest that there can be an advantage to using RL methods compared to traditional mean-variance optimisation methods for portfolio management because they optimise for expected future rewards over more extended periods (at least under certain market conditions). The most benefit could be gained from the proposed RL methods in upward trending markets as this is where they had the potential to outperform the equally weighted and traditional mean-variance optimisation methods. This result especially applies to a particular excess risk range (in the Dow 30 market, this was between around 1% and 13%). The results also suggest that in sideways trending markets, the performance of SPO and MPO can be closely matched by the proposed RL models for the majority of the excess risk range tested.



## Chapter 4

# Conclusions and Future Work

### 4.1 Conclusions

This study compared the portfolio management performance of traditional mean-variance optimisation models like SPO and MPO to that of RL methods (FRONTIER) in risk-return space. One of the main reasons for doing so was the capacity of RL models to optimise their expected rewards over more extended periods compared to the relative short-sighted optimisations of SPO and MPO. This long-term optimisation is important when considering the portfolio management problem since immediate actions can affect an agent's ability to produce optimal rewards in the future due to transaction costs. Before doing so, in order to achieve the aims and objectives of this study, four limitations of the evaluated previous research on portfolio management with RL methods were addressed. This process entailed creating the proposed RL models that could take a wide range of investor preferences into account in terms of trade-aversion and risk-aversion to suit their particular risk and return objectives. The inclusion of these investor preference parameters into the proposed RL models resulted in Pareto optimal frontiers in risk-return space that could be compared to those of traditional mean-variance optimisation models (SPO and MPO). Tests were repeated on three different markets that represented three different economies and overall market trends to assess the applicability of the results to different market conditions. All models in this study were created/modified to account for more realistic non-linear changes in transaction cost introduced by market volatility and trading volume in addition to linear changes related to bid-ask spreads and broker commission.

The results of this study were then compiled in order to answer the research question. The results suggest that there can be an advantage to using RL methods

compared to traditional mean-variance optimisation methods for portfolio management because they optimise for expected future rewards over more extended periods (at least under certain market conditions). The proposed RL models were able to significantly outperform traditional mean-variance optimisation methods like SPO and MPO in upward trending markets up to some excess risk limit (in the case of the Dow 30 market, this limit was around 13%). The results also suggest that in sideways trending markets, the performance of SPO and MPO can be closely matched by the proposed RL models for the majority of the excess risk range tested. In downward trending markets, no conclusions could be drawn on the out-performance of traditional mean-variance optimisation models and the proposed RL models. The most benefit can be gained from the proposed RL methods in upward trending markets as this is where they have the potential to outperform EW and traditional mean-variance optimisation methods. This result especially applies to a particular excess risk range (in the Dow 30 market, this was between around 1% and 13%). This range might change depending on the market or underlying assets held in the portfolio.

It is important to note that in hindsight, evaluations can be made on markets with different trends, enabling the selection of a top-performing model with ease. However, knowing in advance what trend a market will have is not necessarily possible. Therefore, for practical applications, the top-performing model can't necessarily be selected with ease as its ranking might depend on market conditions. The results of this study suggest that choosing the proposed RL model (FRONTIER) might be a good option given this uncertainty since it provides potential for outperforming traditional mean-variance optimisation models if the market trends upward while matching their performance if it happens to trend sideways and heavily investing in the risk-free asset in downward trending markets.

The caveats and specific market conditions under which these models can outperform each other highlight the importance of a more comprehensive comparison in risk-return space for a range of risk values. These Pareto optimal frontiers give investors a more granular view of which models might provide better performance for their specific risk tolerance or return targets. It also gives insight to model developers to see where the possible limitations of specific methods are so that they can be improved.

## 4.2 Future Work

In future work, the methods of this study can be repeated on different markets with similar overall price trends to see if the results hold for all markets with these characteristics. Some of the policy network hyperparameters given in this study can be fine-tuned to assess the effect they have on overall performance. The proposed RL models can also be extended to allow short positions (negative positions) to see how this affects the results. It would also be interesting to add more features like market sentiment to the state input of RL models to see whether this improves the implicit returns forecasting ability and subsequent portfolio management performance. Perhaps the most interesting development from this study would be to change the reward function of other/future state-of-the-art RL models to incorporate specific investor preferences so that they can also be compared more comprehensively in risk-return space to traditional mean-variance optimisation methods. The caveats and specific market conditions under which these models can outperform each other highlight the importance of a more comprehensive comparison in risk-return space for a range of risk values.

## References

- [1] Harry Markowitz. “Portfolio Selection”. In: *The Journal of Finance* 7.1 (1952), pp. 77–91.
- [2] Harry Markowitz. “Portfolio Selection: Efficient Diversification of Investments”. In: *The Journal of Finance* (1959).
- [3] Bin Li and Steven CH Hoi. “Online portfolio selection: A survey”. In: *ACM Computing Surveys (CSUR)* 46.3 (2014), pp. 1–36.
- [4] Eric Zivot. *Introduction to Computational Finance and Financial Econometrics with R*. Chapman & Hall CRC, 2017.
- [5] Farzan Soleymani and Eric Paquet. “Financial portfolio optimization with on-line deep reinforcement learning and restricted stacked autoencoder—DeepBreath”. In: *Expert Systems with Applications* 156 (2020), p. 113456.
- [6] Zhengyao Jiang and Jinjun Liang. “Cryptocurrency portfolio management with deep reinforcement learning”. In: *2017 Intelligent Systems Conference (IntelliSys)*. IEEE. 2017, pp. 905–913.
- [7] Ralph Neuneier. “Optimal asset allocation using adaptive dynamic programming”. In: *Advances in Neural Information Processing Systems* (1996), pp. 952–958.
- [8] Zihao Zhang, Stefan Zohren, and Stephen Roberts. “Deep learning for portfolio optimization”. In: *The Journal of Financial Data Science* 2.4 (2020), pp. 8–20.
- [9] Stephen Boyd et al. “Multi-Period Trading via Convex Optimization”. In: *Foundations and Trends in Optimization* 3.1 (2017), pp. 1–76.
- [10] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

- [11] A Rapaport. "Dynamic programming models for decision making". In: *Journal of Mathematical Psychology* 4 (1967), pp. 48–71.
- [12] Terry Lingze Meng and Matloob Khushi. "Reinforcement learning in financial markets". In: *Data* 4.3 (2019), p. 110.
- [13] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. "A deep reinforcement learning framework for the financial portfolio management problem". In: *arXiv preprint arXiv:1706.10059* (2017).
- [14] Angelos Filos. "Reinforcement learning for portfolio management". In: *arXiv preprint arXiv:1909.09571* (2019).
- [15] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533.
- [16] Matthew Hausknecht and Peter Stone. "Deep Recurrent Q-learning for Partially Observable MDPs". In: *arXiv preprint arXiv:1507.06527* (2015).
- [17] Hongyang Yang et al. "Deep reinforcement learning for automated stock trading: An ensemble strategy". In: *Available at SSRN* (2020).
- [18] Tarrin Skeepers, Terence L. van Zyl, and Andrew Paskaramoorthy. "MA-FDRNN: Multi-Asset Fuzzy Deep Recurrent Neural Network Reinforcement Learning for Portfolio Management". In: *2021 8th International Conference on Soft Computing Machine Intelligence (ISCMI)*. 2021, pp. 32–37. DOI: [10.1109/ISCMI53840.2021.9654987](https://doi.org/10.1109/ISCMI53840.2021.9654987).
- [19] Paul Wilmott. *Paul Wilmott Introduces Quantitative Finance*. John Wiley & Sons, 2007.
- [20] Xiao-Yang Liu et al. "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance". In: *arXiv preprint arXiv:2011.09607* (2020).
- [21] Verizon Media. *Yahoo Finance*. URL: <https://finance.yahoo.com/> (visited on 03/15/2021).
- [22] Quandl. *3-Month Treasury Bill: Secondary Market Rate*. URL: <https://data.nasdaq.com/data/FRED/DTB3> (visited on 03/20/2021).
- [23] Kalyanmoy Deb. *Multi-objective optimisation using evolutionary algorithms*. Springer, 2011, pp. 12–22.

- [24] Martín Abadi et al. *TensorFlow*. URL: <https://www.tensorflow.org/> (visited on 07/07/2021).
- [25] Steven Diamond and Stephen Boyd. *CVXPY*. URL: <https://www.cvxpy.org/> (visited on 12/01/2021).
- [26] Stephen Boyd et al. *cvxportfolio*. URL: <https://github.com/cvxgrp/cvxportfolio> (visited on 08/11/2021).
- [27] Razvan Oprisor. *cvxportfolio*. URL: <https://github.com/roprisor/cvxportfolio/tree/2019-refresh> (visited on 09/13/2021).
- [28] Xiao-Yang Liu et al. *FinRL*. URL: <https://github.com/AI4Finance-Foundation/FinRL> (visited on 08/25/2021).
- [29] Antonin Raffin et al. *Stable-Baselines3*. URL: <https://github.com/DLR-RM/stable-baselines3> (visited on 08/25/2021).