Tiako P.F. (ed) *Intelligent Computing and Consumer Support Applications*. Chronicle of Computing. OkIP
 https://doi.org/10.55432/978-1-6692-0003-1_2

# Robustness Analysis of Deep Reinforcement Learning for Online Portfolio Selection

Marc Velay[1,2], Bich-Liên Doan[1], Arpad Rimmel[1], Fabrice Popineau[1], Fabrice Daniel[2]

[1] *Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91190, Gif-sur-Yvette, France*
[2]*Lusis, 5 cité Rougemont, 75009 Paris, France*

*ABSTRACT*

*Online Portfolio Selection (OLPS) requires a careful mix of assets to minimize risk and maximize rewards over a trading episode. The stochastic, non-stationary aspect of the market makes decision-making very complex. Heuristic methods relying on historical returns were traditionally used to select assets that found a balance of risk and reward. However, improvements in modeling time series from Neural Networks led to new solutions. Deep Reinforcement Learning (DRL) has become a popular approach to solve this problem, but its methods rarely reach a consensus among publications. In other fields, solutions using non-Markovian state representations are frequent. Crafting rewards to improve agent learning is common but has effects on the resulting behaviors. The resulting processes are rarely compared to other recent State-of-the-Art solutions but to heuristic algorithms. The proliferation of approaches motivated us to benchmark them using traditional financial metrics, and evaluate their robustness over time and across market conditions. We aim to evaluate the contributions to measured performance from each method in market representation, policy learning and value estimation.*

*Keywords: Online Portfolio Selection, Deep Reinforcement Learning, Resource Allocation, Robustness Analysis*

## I. INTRODUCTION

Traditional Portfolio Selection involves allocating funds across predefined financial assets. Several existing methods can be used to solve this problem. These usually rely on historical standard deviation of returns, the volatility, correlations between assets and their expected change. A balance between maximizing rewards and minimizing the risk of losing value can be found through statistical means and Dynamic Programming (Markowitz, 1952).

The main difficulties that arise from this field are linked to the nature of the market. It is stochastic with a large proportion of aleatoric uncertainty, and is non-stationary, where the parameters of underlying models evolve with time. Properly estimating dynamics is complex and rarely accurate. Instead, we seek to find positions, which minimize the effect of unexpected shifts. This requires anticipation and planning. Furthermore, epistemic uncertainty leads to unstable models, with limited usable timeframes. Therefore, the objective is for our strategies to perform well now, but also to maintain a similar performance over time.

*Corresponding author E-mail: marc.velay@centralesupelec.fr

Online Portfolio Selection (OLPS) is a problem that requires solutions from Finance, Optimization and Dynamic Control, based on Reinforcement Learning (Pigorsch, 2021; Ye, 2020; Zhang, 2020). Similar to traditional Portfolio Selection, it seeks to find the best compromise of returns and risk, but also evolves the positions at intervals. OLPS solutions rely on Deep Learning to find the optimal positions. Few components of published solutions have reached a consensus. Multiple methods are employed for representing the market, choosing reward functions, evaluating performance or which learning algorithms to use. Each individual choice has theoretical differences which modify how the problem is viewed and solved. Separating contributions from each component and the variability from using different datasets at different periods is complex and prevents establishing a state of the art methodology for OLPS.

In our work, we focus on DRL approaches within three blocks: State representations, learning algorithms and reward functions. These are defined in the Preliminaries section. We have excluded comparing action shaping as we rely on continuous actions using softmaxed allocation weights vectors, as one of the few components with a wide consensus (Benhamou, 2020; Liang, 2018; Ye, 2020; Zhang, 2020). Our experiment training and backtesting processes are done with domain constraints in mind, to ensure proper results. Within this framework, we measure overall performance, performance over time and robustness to changes in market conditions and to worse-case scenarios.

## II. PRELIMINARIES

### A. Reinforcement Learning

Deep Reinforcement Learning consists in learning a Policy, a choice of actions, that maximizes a reward function given observations of an environment. This problem is often formulated within the framework of a Markov Decision Process (MDP) as a 4-tuple $(\boldsymbol{\mathcal{S}},\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{P}},\boldsymbol{\mathcal{R}})$.

The observations, which MDP assumes are enough to inform the agent in making optimal decisions, are denoted $s_t \in \boldsymbol{\mathcal{S}}$. These transitions are sampled from the distribution of probabilities $\boldsymbol{\mathcal{P}}$: $\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}} \times \boldsymbol{\mathcal{S}} \rightarrow [0,1]$, where $\boldsymbol{\mathcal{A}}$ is the set of possible actions. From these transitions, we can observe a new state $s_{t+1}$ and an immediate reward $r \in \boldsymbol{\mathcal{R}}$.



Presented at the 2022 OkIP International Conference
on Automated and Intelligent Systems (CAIS)
in Oklahoma City, OK, USA, and Online, on October 3-6, 2022

By interacting with the environment, the agents learn estimations of value functions. When explicit, these are mappings of expected rewards from a given state for each valid action. Instead, implicit methods predict actions directly from the observations, using mechanisms such as the Actor-Critic algorithm. The learning algorithms we explore in our benchmark are DDPG, ADDPG, PPO, and SAC (Benhamou, 2020; Li, 2019b; Liang, 2018).

### *B. Market Interaction*

Market observations fall under three categories. The first is strictly Markovian, which uses only current prices and allocation (Li, 2019b). This includes financial indicators, such as movement direction, momentum, and volatility. The last two are non-Markovian, using multiple observations containing only historical asset information (Liang, 2018) or asset and contextual historical information (Benhamou, 2020). Each category aims to reduce the market noise and extract some information using various levels of domain knowledge.

The actions $a \in \mathcal{A}$ are the outputs of an agent's policy. They express the portfolio allocation weights for each asset, including liquidity. The result is a [k+1] continuous-valued softmax vector. An allocation is expressed at each iteration, at each start of a business day, based on information from previous observations.

### *C. Metrics*

We evaluate Portfolio Selection strategies over a fixed period. The first metrics will compute an overall performance across the whole period. We use Net Returns, corresponding to the normalized difference in value between the start and T, the trading horizon. The metric describes how well the agents behaved but gives little information about generalization.

The second type evaluates risk-to-return ratios, comparing a portfolio's returns to fluctuations (Benhamou, 2020). The Sharpe ratio compares the portfolio's returns over a predefined threshold to the returns' standard deviation. The Sortino ratio has a similar approach but replaces the total standard deviation with the downward deviation, only looking at negative returns. These metrics give an estimate of the stability of strategies. Investors may be inclined to less total gains but smaller, guaranteed returns.

The third type evaluates the robustness of the algorithms (Moos, 2022). For this, we first focus on CVaR, the 5% worse observed returns, corresponding to the worse expected losses by the strategy. Then, we look at performance over time by computing the return and risk metrics over shorter windows. Plotting these results gives a qualitative value but can be quantitatively compared using the maximum difference and standard deviation in measurements.

## III. EXPERIMENTS

The experiment relies on market data composed of stocks from the CAC40, commodities futures, and Forex pairs, totaling 18 assets. Previously cited papers use between 4 and 10 assets. Capturing a broader picture of the market can provide more diverse, stabler portfolios. We used data from 2002 to 2019 for training, validation, and testing for the experiment. The last years, 2020-2022, are reserved for backtesting our strategies. The raw data comprises Open, High, Low, and Closing prices and Volume (OHLCV). The other information in observations is computed based on sliding windows of OHLC prices. To avoid leaking information to other subsets, we include the sliding window length required to compute indicators in the following subset. These subsets are randomly sampled from the 17 years and distributed over the whole period.

We follow an identical training process for each learned model. First, we run a hyperparameter (HP) search using Optuna on pre-defined spaces. Second, we initiate each algorithm from a set of seeds for reproducibility and the best sampled HP. Third, we train the agents on the equivalent of 1000 iterations of the training set, which we sample randomly. Adding iterations has proven to decrease training performance due to the low volume of data available. The agents generally reach a plateau before this threshold and perform worse afterward. Finally, we select the agent with the best validation score or the latest trained iteration based on the test set performance. Their performance may vary, and the test set gives a better impression of their generalization capabilities than validation scores.

To evaluate the performance of DRL algorithms in different conditions, we select periods inside the backtesting set. To qualify for either bull or bear markets, we must observe a down or upwards trend for at least three months with no reversal. Due to length requirements, we only select one of each out of the two available years. We did not use a specific time frame to evaluate performance during worse-case scenarios, instead relying on the CVaR metric over the whole backtesting period. Finally, to evaluate performance over time, we run the algorithms for one month every two months to see the evolution of their capabilities.

## IV. RESULTS & DISCUSSION

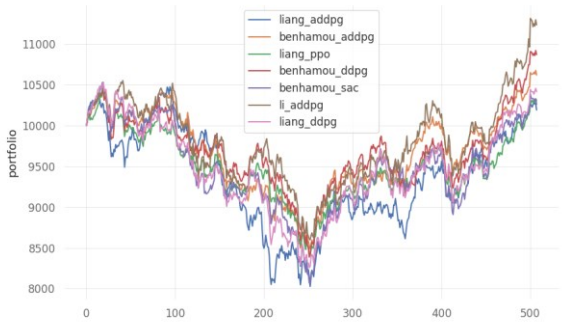


Fig 1: Portfolio Value Across Backtesting Period

The observed results from Fig 1 show that the combinations of state representations from past papers, named after their authors, and associated learning algorithms converge to similar behaviors, closely following market movements. During a financial crisis, the best algorithm yields a 15% return over two years. The following steps involve analyzing the distribution of returns and allocations. We found a surprisingly common outcome, represented by the second graph in Fig 2. Most combinations converged to constant allocations, regardless of current market conditions. These vary slightly by seed and

algorithm but remain composed of a small subset of assets with constant weights. Combinations such as the first graph in Fig 2 seem to vary allocations but converge within some iterations to the first outcome. While the net returns are strictly positive and thus could be considered welcomed from a Finance point of view, the behavior indicates poor performance in DRL agents. These results are unexpected and prevent further analysis of component contribution since agents reached the same behavior regardless of variations in approaches and market conditions.

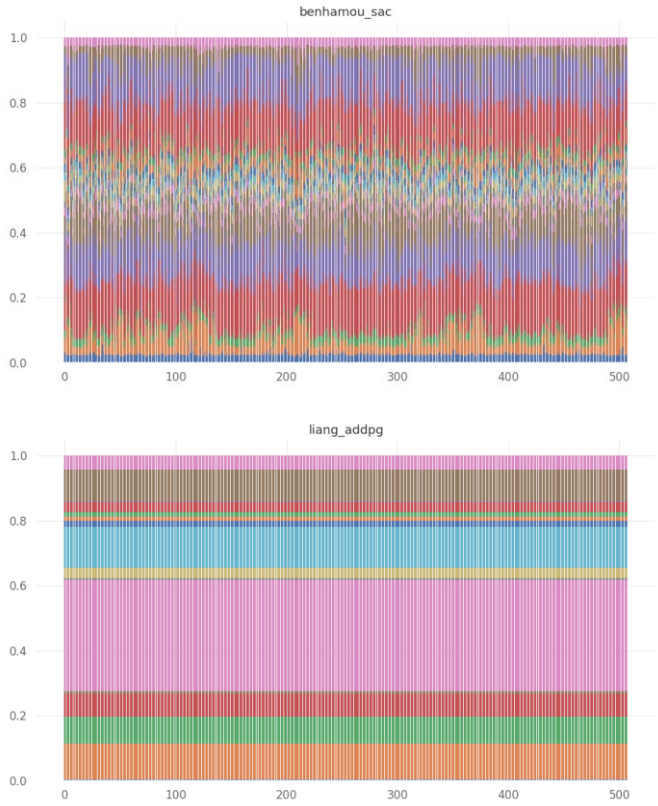


Fig 2: Rebalancing of Allocations During Backtesting

Analyzing the source of this behavior is challenging due to the nature of the environment. The data is noisy and highly stochastic, with few data points available over the training period. Our main hypothesis is that the agents do not recognize patterns and overfit a policy to predict the best historical actions. The strategy is akin to applying Markowitz Portfolio Selection once over the 14 years of training data to determine which assets are most likely to work in the future. While most learning algorithms are susceptible to hyperparameter selection, the first steps involved finding a set that seemed to learn efficiently over a short training period. The validation step would have kept models that generalized best. If the training process is not responsible for overfitting, we explore processes applied to the data. For this, we analyze approaches from other works. While few discuss such issues (Durall, 2022), we can extrapolate how others reached their results.

It is common to augment when training on insufficient data (Benhamou, 2020; Liang, 2018). We should not do so with adversarial approaches, as adding noise to an already noisy time series would further hide signals. The most promising solutions involve modifying the data we use to train the models. (Pigorsch, 2021) predicts a value score for each asset before using a softmax layer to obtain the final allocation weights. Thus, we train the policy network on individual asset histories instead of the market as a whole. An alternative is widely increasing the number of assets analyzed before sampling a limited amount to train on as a representation of the market (Li, 2019a). We did this step implicitly by selecting 18 assets manually but should adopt it more widely.

To conclude, such obstacles are frequent in Reinforcement Learning and often require experimenting to solve. Determining the origin of a problem is challenging due to the lack of interpretability and explainability of policies. It is currently challenging to distinguish if a problem originates in the selection of a model, the learning algorithm's tuning, the quality of the environment representation, or the desired behavior expressed through the rewards. Our future works aim to clarify this position and agents' learning process. We believe Deep Reinforcement Learning could provide an attractive solution to OLPS. We strive to improve current solutions in terms of explainability, learning efficiency, and robustness in future works.

## REFERENCES


Eric Benhamou et al. (2020). *Bridging the Gap Between Markowitz Planning and Deep Reinforcement Learning*. en. SSRN Scholarly Paper ID 3702112. Rochester, NY: Social Science Research Network. DOI: 10. 2139/ssrn.3702112.

Ricard Durall. (2022) *Asset Allocation: From Markowitz to Deep Reinforcement Learning*.

Xinyi Li et al. (2019a). *Optimistic Bull or Pessimistic Bear: Adaptive Deep Reinforcement Learning for Stock Portfolio Allocation*. In: arXiv:1907.01503.

Yang Li, Wanshan Zheng, and Zibin Zheng. (2019b). *Deep Robust Reinforcement Learning for Practical Algorithmic Trading*. en. In: IEEE Access 7, pp. 108014– 108022. ISSN: 2169-3536. DOI: 10 . 1109 / ACCESS . 2019.2932789.

Zhipeng Liang et al. (2018) *Adversarial Deep Reinforcement Learning in Portfolio Management*. In: arXiv:1808.09940.

Harry Markowitz. (1952). *Portfolio Selection*. In: The Journal of Finance 7.1, pp. 77–91. ISSN: 0022-1082. DOI: 10/bxrq3f.

Janosch Moos et al. (2022). *Robust Reinforcement Learning: A Review of Foundations and Recent Advances*. en. In: Machine Learning and Knowledge Extraction 4.1, pp. 276–315. ISSN: 2504-4990. DOI: 10. 3390/make4010013.

Uta Pigorsch and Sebastian Schafer. (2021) *High-Dimensional Stock Portfolio Trading with Deep Reinforcement Learning*. In: arXiv:2112.04755.

Yunan Ye et al. (2020). *Reinforcement-Learning based Portfolio Management with Augmented Asset Movement Prediction States*. In: arXiv:2002.05780.

Yifan Zhang et al. (2020). *Cost-Sensitive Portfolio Selection via Deep Reinforcement Learning*. In: IEEE Transactions on Knowledge and Data Engineering PP, pp. 1–1. DOI: 10/gj6rzg.