Equity Portfolio Optimization Using Reinforcement Learning: Emerging Market Case

 # Abstract Summary: The abstract outlines a study on portfolio optimization using reinforcement learning agents in financial markets. The study compares the performance of these agents against traditional market index funds, specifically targeting the BIST30 market constituents. The agents, trained with price indicators and fundamental company information, aim to generate higher returns and a better Sharpe ratio than the market index. The study emphasizes the agents' ability to learn from trial and error without prior knowledge of the market environment and their encouragement to diversify portfolios for increased real-life applicability.

#Objective: The challenges of working with financial time series data, emphasize behaviors like heteroscedasticity and non-linearity that make forecasting difficult. It's a trade-off between risk and return in portfolio optimization and aims to develop a portfolio weighting mechanism that outperforms the BIST30 market index. The focus is on achieving better performance in both risk management and returns within the same asset universe. The modeling of portfolio optimization as a Markov decision process, where the next action depends solely on the current state. State, action, and reward functions are defined, creating a reinforcement learning framework to learn the best investment policy. The use of reinforcement learning enables the exploration of novel investment models and potentially addresses behavioral biases, ultimately aiming for better portfolio decisions.

#Background:

Time Series: A company's price **p** is influenced by numerous factors, resulting in high uncertainty. However, due to the non-linear and non-stationary nature of price series, they are not directly utilized in analysis. Instead, they are transformed into return series, **r** to meet certain assumptions necessary for methodology development. In contrast to price, returns show stationary behavior with a mean close to zero. They also tend to follow the Gaussian distribution.

$$\mathbf{r}_t = \frac{p_t - p_{t-1}}{p_{t-1}} \in \mathbb{R}^{\tau}_{(-\infty,\infty)}$$

Portfolio: Each asset's worth relative to the total net worth of the portfolio constitutes its weight vector, denoted by w. The weight vector is normalized, ensuring that the sum of weights equals 1. To balance between timely incorporation of new market information and minimizing transaction costs, the rebalancing period is set at one month for simplicity, aligning with common industry practices.

Markov Decision Process: The Markov decision process is a mathematical framework to model trial-and-error processes. An MDP consists of 4 components; agent, action, reward, and state. The process starts with an initial state S0, and depending on its value, an action A0 is taken. Afterward, a reward R0 is received, and the agent passes to state S1. This is the effect of that taken action A0. For a τ –step MDP, this procedure continues until reaching Sτ.

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \qquad v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right]$$

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right]$$

Existing RL Approaches:
The paper explains two approaches in detail which will be used for final optimization:

- **DDPG (Deep Deterministic Policy Gradient):** DDPG agents utilize deep neural networks to approximate the policy and value functions in continuous action spaces.

- **PPO (Proximal Policy Optimization)**: PPO is an algorithm designed for optimizing large-scale policy functions, commonly used in reinforcement learning tasks.

#Modeling and Analysis:

Simplification: To start with, we made the following assumptions: no transaction costs, no slippage, no market impact, and the existence of fractional shares. One important restriction for the models is the short sale. As our main benchmark is a stock market index, we only allowed long-only portfolios

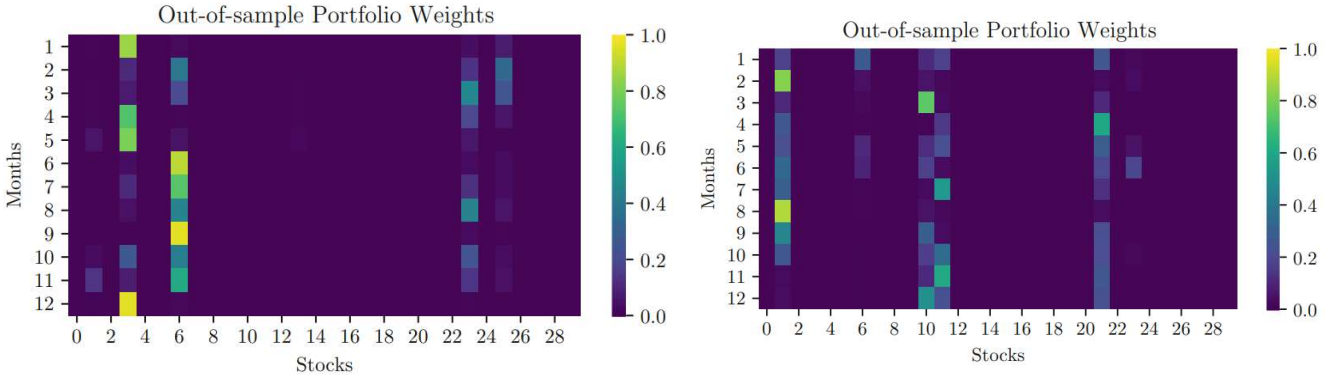
Setup: Divided dataset into two parts, which correspond to the training set from 2010-01-01 to 2019-12-31 and the test set from 2020-06-01 to 2021-06-01. We use return and volatility, in addition, we use technical indicators such as RSI, CCI, ADX, ATR, SAR, and EMA divergence to help provide information for short-term price characteristics. Fundamental features such as price-to-book ratio and price-to-sales ratio to help understand the long-term behavior. Before training, transform each sample to 21 × 30 × 12 tensor and divide by their standard deviation which is estimated using the training set to prevent any look-ahead bias.

Calculation: First, we calculate the return and compare with the past returns, second we measure portfolio diversification to assess risk.
where Φ is Gaussian cdf. Parameters μ and σ are estimated from 1 year's worth of return samples.

$$\mathfrak{r}_{\mu,\sigma} = \Phi(\mathfrak{r}; \mu, \sigma) - 0.5 \in \mathbb{R}_{[-0.5,0.5]}$$

$$R(a, \mathfrak{r}; \mu, \sigma) = \mathfrak{r}_{\mu,\sigma} H_n(a) \in \mathbb{R}_{[-0.5,0.5]} \qquad H_n(p) = -\sum_{i=1}^{n} \frac{p_i ln(p_i)}{ln(n)} \in \mathbb{R}_{[0,1]}$$

DDPG, a reinforcement learning algorithm, utilizes target networks for stable learning by gradually updating them alongside main networks. Experience replay stores transitions to break correlations and stabilize learning, while Ornstein-Uhlenbeck noise promotes exploration. Its actor-critic architecture enables action selection and evaluation. Additionally, DDPG includes a diversification term in the reward function to discourage hard-weighting, promote diversified portfolio weights, and mitigate risks associated with asset concentration.



PPO, a reinforcement learning algorithm, employs a surrogate objective to limit policy changes and ensure stable training. Experience replay isn't typically used in PPO, which directly optimizes the policy without storing transitions. PPO encourages exploration through its exploration policy. It follows an actor-critic architecture where the actor proposes actions and the critic evaluates them. Designed the networks of PPO in a similar way to the DDPG agent. Additionally, maximum pooling layers are used to reduce overall model complexity.

#Results:

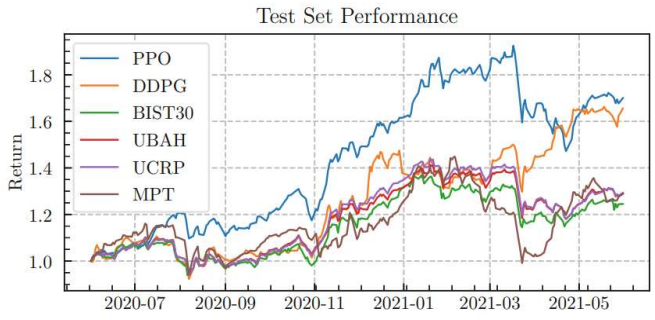


Fig. 3. Portfolio values.

The two agents have resulted in similar overall returns with different dynamics. The overall performance of the PPO agent is superior, however, it experiences a large drawdown which reduces its final value significantly. DDPG has performed better in the last two months. It is also highly correlated with the market index in the first 5 months.

#Conclusion:

We can conclude that RL agents could provide a better weighting model to capitalization-weighted stock market indices in emerging markets under certain conditions.