NORTHWESTERN
UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2023-10

May 2023

Cryptocurrencies Portfolio Trading with Deep Reinforcement Learning

Yujia Xie

Abstract

This research paper investigates the performance of deep reinforcement learning (DRL) algorithms in cryptocurrencies portfolio trading, which includes BTC, ETH, LTC, AAVE, UNI, and SOL. Cryptocurrency market is known for its volatility and unpredictability because many are not backed by substantial elements, despite it has been a profitable and low-entry level market that attracts the public attentions. In recent years, researchers have been exploring and demonstrating that deep reinforcement learning could learn beat-the-market low-frequency trading strategies in some stock market conditions. Therefore, this paper intends to evaluate the effectiveness of common DRL frameworks on the cryptocurrency trading market, considering both the cumulative returns and the risk level measured by Sharpe ratio. We show that, with the proper network architecture, DRL algorithms can learn a trading strategy that gains 30\% more return than a baseline equal-weight strategy.

Keywords

Deep Reinforcement Learning, Cryptocurrencies, PPO, A2C, DDPG, LSTM

Cryptocurrencies Portfolio Trading with Deep Reinforcement Learning

Yujia Xie [1]

Abstract

This research paper investigates the performance of deep reinforcement learning (DRL) algorithms in cryptocurrencies portfolio trading, which includes BTC, ETH, LTC, AAVE, UNI, and SOL. Cryptocurrency market is known for its volatility and unpredictability because many are not backed by substantial elements, despite it has been a profitable and low-entry level market that attracts the public attentions. In recent years, researchers have been exploring and demonstrating that deep reinforcement learning could learn beat-the-market low-frequency trading strategies in some stock market conditions. Therefore, this paper intends to evaluate the effectiveness of common DRL frameworks on the cryptocurrency trading market, considering both the cumulative returns and the risk level measured by Sharpe ratio. We shown that, with the proper network architecture, DRL algorithms are able to learn a trading strategy that gains 30% more return than a baseline equal-weight strategy.

1. Introduction

The emergence of blockchain technology has introduced a new form of digital currency – Cryptocurrency – into the financial system, and its market has experienced explosive growth in recent years (Mukhopadhyay et al., 2016). Bitcoin (BTC) itself has achieved a peak market capitalization of over $1 trillion in 2021, and other cryptocurrencies like Ethereum (ETH), Binance Coin (BNB), Litecoin (LTC), and Ripple (XRP) have achieved a combined market size of over $500 billion. The unique blockchain properties enable a decentralized peer-to-peer network structure that provides crypto transactions with a more transparent, secure, permanent, and efficient space, which gives a better solution to the problems that traditional financial institutions have. Thus, more and more investors are attracted by this new market, and have proposed a range of statistical models and machine learning algorithms that are commonly used in stock trading market to better predict the cryptocurrency trading market. Nevertheless, due to its highly volatile nature, impact from political and economic issues, unpredictable social sentiments, and unbounded trading time period and uncapped daily price variation, the cryptocurrency market is subject to significant fluctuations, and thus making the prediction and profitable trading a complicated challenge.

Moreover, even though several researches claimed that they could achieve a cryptocurrency price prediction with a Mean Squared Error (MSE) of as low as 0.0011 (Parekh et al., 2022), single stock closed price prediction or trend prediction is barely helpful in real-life trading task because the decisions like whether to buy or sell, the corresponding amount, and the action time are still left unknown. Therefore, the final objective of a trading prediction task should be portfolio return-driven rather than price-driven.

Fortunately, a growing body of studies have demonstrated the effectiveness of deep reinforcement learning algorithms on these return-driven tasks. Through simulating a trading environment, an agent could learn a policy that maximize cumulative return to perform the optimal buy and sell actions at the calculated time, based on user-input information like initial capital amount, trading-platform specific rules, and individual trading preference.

In this paper, we endeavor to investigate the domain of cryptocurrency portfolio management using deep reinforcement learning framework on six cryptocurrencies' historical data and their common technical indicators. We will use equal-weight investing strategy as our baseline cumulative return measurement.

2. Literature Review

2.1. Artificial Intelligence in Quantitative Finance

In recent years, researchers have been adopting and exploiting machine learning (ML) algorithms to perform market analysis and price predictions to quantitatively understand the stock market dynamic, and some of these techniques have been applied to the newly emerged cryptocurrency market. Among them, advanced deep learning network LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit) combining with Twitter sentiment analysis achieved the highest performance in BTC price prediction task (Serafini et al., 2020) (Parekh et al., 2022). The common trading strategies based on ML price prediction is to buy if predicted price is higher than the current price, and sell otherwise (Ji

et al., 2019).

On the other hand, Reinforcement Learning could derive next-period actions based on the data in simulated environment, which allows the trader to develop an automated strategy across multiple assets. (Necchi, 2016) demonstrated that state-of-the-art RL algorithms like NPGPE learned a long-short trading strategy that significantly outperforms the buy-and-hold strategy on simulated stock data. (Liu et al., 2022) proposed an open-source deep reinforcement learning library that provides structured and standardized tools for financial tasks like performing automated stock trading or providing future contract strategies. (Yang et al., 2021) proposed an ensemble model with three commonly-used RL algorithms – PPO, A2C, and DDPG – that outperforms the Dow Jones Industrial Average index in terms of risk-adjusted return measured by the Sharpe ratio. Finally, (Gort et al., 2022) proposed a new training pipeline that reduces overfitting in the trained agent, and concluded that the corrected PPO agent achieved a 15% increase in cumulative returns in the period where the cryptocurrency market has crashed twice.

3. Contributions

Following are the major contributions of this paper:

- We compared the performance of the previously proposed DRL agents on the most recent market condition through backtesting.
- Since LSTM has been proved to be a better choice in dealing with long-term time series data than other artificial neural networks, we combined the PPO agent with LSTM neural network to account for the serial correlation and historical memories in the cryptocurrency data.

4. Technical Methods

4.1. Deep Reinforcement Learning Framework

Since the crypto trading is a essentially making multiple decisions under a discrete, stochastic, and sequential environment, we could model this task as a Markov Decision Process (**MDP**), where an agent could learn a policy in a pre-defined environment by performing a set of actions, from which the agent will receive rewards based on the chosen action. In our setting, we defined these elements as the following:

- State: $s_t = [c_t, \mathbf{p_t}, \mathbf{h_t}, \mathbf{f_t}]$, where $c_t \in \mathbb{R}$ is the cash amount at time t, $\mathbf{p_t} \in \mathbb{R}_+^D$ is the market price vector for $D = 6$ cryptocurrencies, $\mathbf{h_t} \in \mathbb{R}_+^D$ is the share holding vector, and $\mathbf{f_t} \in \mathbb{R}^{I \times D}$ is a feature vector holding the calculated information for a list of $I = 6$ chosen technical indicators.
- Action: $a_t \in [-1, 1]^D$. A negative action means sell, a positive action means buy, and 0 means hold. The action is also adjusted to represent the amount of shares to buy/sell for corresponding cryptocurrency type.
- Reward: $r_t \in \mathbb{R}$. We define the reward as the difference in total asset between the current time t and the previous time $t-1$, where the total asset is

$$m_t = c_t + \mathbf{p_t}^T \mathbf{h_t}$$

- Expected Return: $Q_t(s_t, a_t)$: the expected return from state s_t until the terminal state.

$$Q_t(s_t, a_t) = \mathbf{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

- Policy: π_t. At time t, the policy will determine an action to take based on the current state s_t. It gives the trader a trading strategy optimized for the rewards defined above.

4.2. Technical Indicators

We incorporated six commonly used technical indicators suggested by (Yang et al., 2021) and (Gort et al., 2022), and we performed an exploratory data analysis to ensure there are no highly correlated variables that could cause multicollinearity issue. Technical indicators are calculated by TA-Lib python library (Benediktsson), which include the following:

- RSI: Relative Strength Index, a price momentum indicator which measures the speed and change of price movement.
- DX: Directional Movement Index, a price momentum indicator which identifies in which direction the price of an asset is moving.
- ULTOSC: Ultimate Ocsillator, a price momentum indicator which measures the price momentum of an asset across multiple timeframes.
- OBV: On Balance Volume, a volume indicator which suggests the flowing momentum of an assets' volume and thus make predictions about its price
- Volume: the market volume of the asset.
- HT_DCPHASE: Hilbert Transform - Dominant Cycle Phase. The moving averages after removing the dominant cycle calculated through the Hilbert Transformation.

We removed indicators like daily high, daily open, and daily low indicators, and only used the daily close as the interested price. The resulting correlation heatmap for the final variables is shown below.

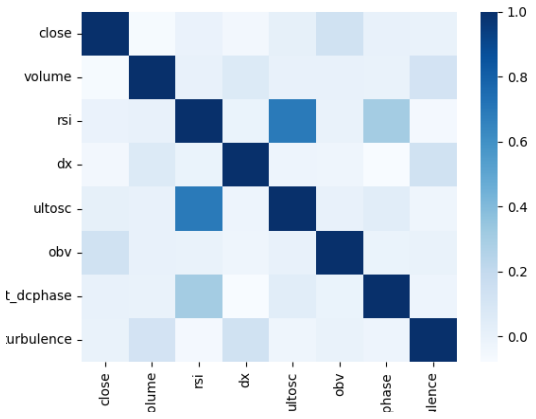

Figure 1. Heatmap for Features Correlation

4.3. Cryptocurrency Trading Environment

We will be deploying the learned strategies on the Alpaca trading platform for paper-trading experiments, so we will simulate the trading environment following the rules and assumptions for the Alpaca platform.

Specifically, it enforces a 0.25% buy fees and 0.15% sell fees as the transaction cost, and there is a transaction limit for different assets. For BTC, the minimum transaction amount is 0.0001 shares; for LTC, UNI, and AAVE, the minimum transaction amount if 0.01; for ETH, it is 0.001; and for SOL, it is 0.05. We will also ensure a non-negative balance on the account, and sell action is only allowed if the number of shares of that asset is non-negative. For all six assets, Alpaca allows for fractional trades up to 9 digits, and they are all non-shortable, which corresponds with our non-negative balance here. Lastly, we set a discount factor of $\gamma = 0.95$ for rewards gained in the long term, representing a depreciation of money value against market inflation.

With these restrictions, we define our reward function as follow:

$$m_t = c_t + \mathbf{p_t}^T \mathbf{h_t} - tc_t^B - tc_t^S$$

$$r_t = \gamma(r_{t-1}) + m_t - m_{t-1}$$

where tc_t^B is the transaction cost for buy actions at time t, and tc_t^S is the transaction cost for sell actions at time t.

4.4. Deep Reinforcement Learning Algorithms

There are many DRL algorithms that are suitable for cryptocurrency price prediction, and we will use the Stable Baseline 3's implementation of actor-critic based algorithms PPO, and A2C as our agents (Raffin et al., 2021b). Based on the literature review, actor-critic algorithms like A2C and PPO are generally more robust than off-policy methods like DDPG and DQN in terms of trading task. In actor-critic algorithms, the agent simultaneously learns a policy function (**actor**) and a value function (**critic**), where the value function is our reward, or the cumulative return over the trading period. In our trading task, we defined our observation space as our state space, where the values could range from negative infinity to positive infinity. Therefore, the stochastic nature of reinforcement learning might randomly pick on a wild value to experience, and thus could not learn much knowledge about this space without an enormous number of training steps to take on. Fortunately, A2C and PPO are more advantageous under this scenario.

4.4.1. Advantage Actor Critic – A2C

The A2C algorithm (Mnih et al., 2016) compares the current actions taken with a baseline value, which is the value generated by taking an average action at the current state, and the resulting difference is called **Advantage Value**.

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

By optimizing on the advantage value, the algorithm not only evaluates how good an action is, but also how much better it could be. Even better, A2C allows for multiprocessing by assigning different batches to different workers at the same training period, and update the policy with the averaged return value, meaning that it allows for fast exploration while maintaining stability at the same time. Therefore, it is suitable for a portfolio trading task.

4.4.2. Proximal Policy Optimization – PPO

The PPO algorithm also follows the actor-critic structure and has an advantage value, except it adds a clipped range for policy updates (Schulman et al., 2017). Advantage values calculated outside of the clipped range are not considered as an update. In essence, PPO searches for new policies that are close to the old policy, which make sense in exploring trading strategies because the buy/sell amount for a low-frequency non-risk-prone trader is usually fixed within a range.

4.4.3. Recurrent PPO

Lastly, instead of using a Multi-layer Perception (MLP) neural network as the actor and critic network in PPO, we use a LSTM neural network following the implementation here (Raffin et al., 2021a). Many researches have shown that for time-series financial data, LSTM is one of the most outperforming neural networks, and therefore it should be a good choice in the trading task as well. Right now, we have 1 LSTM layer with 256 hidden units for both actor and critic network, and used $tanh$ as the activation function.

5. Experimental Data

We use 5-minute interval historical data of six cryptocurrencies AAVE/USDT, BTC/USDT, ETH/USDT, LTC/USDT, UNI/USDT, SOL/USDT from 10/18/2021 to 03/23/2023 downloaded through Binance API as training and validation dataset, and data from 03/24/2023 to 04/24/2023 as testing dataset. In total, there are 899,436 training/validation data points, and 53,562 trading data points for backtesting.

6. Backtesting Results & Analysis

We evaluate the performance of each agent based on cumulative return, which is defined as

$$cumulative\_return = \frac{r_T}{c_0}$$

and Sharpe Ratio

$$Sharpe = \sqrt{factor}\frac{\bar{\mathbf{r}}}{\sigma(\mathbf{r})}$$

where $factor = 60/5 * 24 * 365 = 105,120$, representing the total data points per year. $\mathbf{r}$ is the rewards array calculated from each step, and r_T is the final rewards we get, and c_0 is the intial capital amount.

Even though the overall cumulative return is negative, We could see that the Recurrent PPO outperforms other agents by over 20%. The general market trend in the trading period is downward trending around early April, and since we do not allow our algorithm to short selling, it is hard to gain positive profit with normal buy and sell actions. However, the both Recurrent PPO and PPO performs better than the equal-weights trading strategy, meaning that the deep reinforcement learning algorithms are learning a working policy from the past data.

7. Limitations

There are several limitations in our trading strategies.

First of all, on-policy methods generally suffer from insufficient sampling problem, meaning that the agent needs to be trained on a large amount of data to learn a decent policy. A typical working reinforcement learning algorithm is trained over hundreds of episodes, yet we only trained over one episode due to computational limitations. Parallel training on GPU with more episodes might generate better results.

Secondly, reinforcement learning algorithms are significantly impacted by hyper-parameters, including batch size, learning rate, gamma, entropy coefficient, etc. Further investigations are needed to generate the optimal set of hyper-parameters for each agent.

Thirdly, our backtest range did not test for the effectiveness of risk control. According to (Gort et al., 2022), the Crypto Volatility Tokens (CVIX) are ERC-20 tokens that aim to track the implied volatility of crypto markets, and a CVIX over 90.1 is considered as too volatile for traders to complete any transactions. In our testing period, the CVIX varies around 58 - 70, meaning that we are not in a risky period.

In addition, the current strategy only includes a tiny set of trading options from the investment market. It is atypical for general public trader to only trade cryptocurrency assets, and therefore a model that allows for a wider range of assets should be further investigated. The current researches focus on developing strategies in a single simulated environment, and future researches should consider scenarios with multiple trading environments.

Lastly, learning patterns from historical data is not effective in cryptocurrency market. Unlike other investment assets where they are backed by established banks and real money, many cryptocurrencies (like DogeCoin) are only backed by public sentiments. Therefore, it is also important to perform public sentiment analysis on Twitter and Reddit, as well as other financial journals, in order to capture this complicated market trend.

8. Conclusion

In this paper, we exploited the Deep Reinforcement Learning Algorithms in developing a trading strategy from historical data using Actor-Critic methods, and have shown that the Recurrent PPO agent is able to outperform on the cumulative returns than other algorithms.

References

Benediktsson, J. ta-lib-python.

Gort, B. J. D., Liu, X.-Y., Sun, X., Gao, J., Chen, S., and Wang, C. D. Deep reinforcement learning for cryptocurrency trading: Practical approach to address backtest overfitting. 2022. doi: 10.48550/ARXIV.2209.05559. URL https://arxiv.org/abs/2209.05559.

Ji, S., Kim, J., and Im, H. A comparative study of bitcoin price prediction using deep learning. *Mathematics*, 7(10), 2019. ISSN 2227-7390. doi: 10.3390/math7100898. URL https://www.mdpi.com/2227-7390/7/10/898.

Liu, X.-Y., Yang, H., Chen, Q., Zhang, R., Yang, L., Xiao, B., and Wang, C. D. Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance, 2022.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning, 2016.

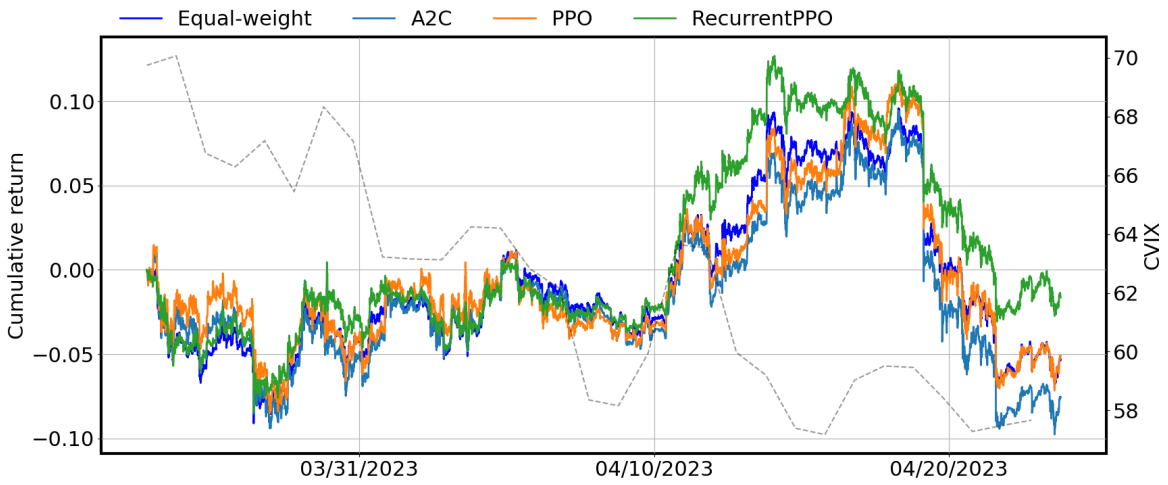

Figure 2. Backtesting Results

Metrics	A2C	PPO	Recurrent PPO	Equal-Weight
Annualized Return	-0.16%	-0.2%	-0.05%	-2.23%
Cumulative Return	-5.73%	-6.95%	-1.63%	-5.31%
Annual Volatility	59.67%	73.63%	51.67%	0.0046
Sharpe Ratio	0.59	0.61	1.40	0.004

Table 1. Performance Across Three Agents

Mukhopadhyay, U., Skjellum, A., Hambolu, O., Oakley, J., Yu, L., and Brooks, R. R. A brief survey of cryptocurrency systems. *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pp. 745–752, 2016.

Necchi, P. G. Reinforcement learning for automated trading. 2016.

Parekh, R., Patel, N. P., Thakkar, N., Gupta, R., Tanwar, S., Sharma, G., Davidson, I. E., and Sharma, R. Dl-guess: Deep learning and sentiment analysis-based cryptocurrency price prediction. *IEEE Access*, 10:35398–35409, 2022. doi: 10.1109/ACCESS.2022.3163305.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021a. URL http://jmlr.org/papers/v22/20-1364.html.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021b. URL http://jmlr.org/papers/v22/20-1364.html.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017.

Serafini, G., Yi, P., Zhang, Q., Brambilla, M., Wang, J., Hu, Y., and Li, B. Sentiment-driven price prediction of the bitcoin based on statistical and deep learning approaches. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020. doi: 10.1109/IJCNN48605.2020.9206704.

Yang, H., Liu, X.-Y., Zhong, S., and Walid, A. Deep reinforcement learning for automated stock trading: An ensemble strategy. ICAIF '20, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450375849. doi: 10.1145/3383455.3422540. URL https://doi.org/10.1145/3383455.3422540.