Deep Reinforcement Learning Models Performance for Crypto-currency Trading

Harold Popluhar
School of Engineering and Technology
Asian Institute of Technology
Klong Luang,
Pathum Thani 12120
Email: st122556@ait.asia

Muhammad Omer Farooq Bhatti
School of Engineering and Technology
Asian Institute of Technology
Klong Luang,
Pathum Thani 12120
Email: st122498@ait.asia

Cedric Le Mercier
School of Engineering and Technology
Asian Institute of Technology
Klong Luang,
Pathum Thani 12120
Email: st122004@ait.asia

***Abstract*—Deep reinforcement learning (DRL) has proven to be a viable way for automatic portfolio management in many financial investment products, including the top cryptocurrencies in market capitalisation. However, implementing a practical and beginner-friendly DRL agent can be difficult because of the different goals, strategies, and libraries available. In this paper, we are implementing the FinRL library for automated cryptocurrency trading and comparing the different agents across several librairies on different trading strategies. Furthermore, we are exploring the viability of the models by validating during periods or turmoil and bear trend, allowing an assessment of whether they are viable even in bad market conditions. Our models were trained and tested on the top 10 cryptocurrencies by market capitalization at the start of training period.**

***Index Terms*—cryptocurrency, bitcoin, deep reinforcement learning, trading, investment**

I. Introduction

In spot trading, the success of an investor will depend directly on three actions: Buying, holding and selling. Seasoned investors use their specific knowledge of macro-economics and indicators to increase their odds of success, whether it is for short term or long term benefit. The application of machine learning algorithms have gained popularity in recent years for cryptocurrency trading because of its high volatility nature. However, implementing DRL or RL trading strategy can be an arduous task for beginners, between different training environments, managing intermediate trading states or standardizing training data and outputs for evaluation.

To answer this demand, FinRL library was developed by Liu et al. [1] as a pratical tool to help analysts get a hands-on approach on implementing DRL algorithms to develop their own trading strategies, by providing various environment and a choice of stock markets including the NASDAQ-100 and SP 500. The FinRL library has a three-layer architecture: Stock market environment, DRL trading agent, and stock trading applications. Each layer operating independently, with lower layer operating as an API for the upper layer. Although complete step-by-step tutorials and comparison of models have been done on the agents available, there are less on cryptocurrency, with only one tutorial available.

In this paper, we explore and implement the libraries and agents available in FinRL for automated cryptocurrency selection and trading. We use the top 10 cryptocurrencies by market capitalization and both on-policy and off-policy models such as DDPG, PPO, TD3, A2C and evaluate their performance in good and bad market conditions. Finally, we explore the usage of an LSTM-based architecture and transfer learning strategies from one cryptocurrency to another.

II. Literature Review

A. DRL on cryptocurrency

Many previous work on traditional market has been done using deep machine learning to predict future price movements and trends. Using price history data as well as some technical indicators, the approach was usually treating it as a regression problem [3]. Instead of predicting future prices, a novel field is the usage of DRL for automated portfolio allocation and trading. There are two major reasons for this approach: First is that high accuracy is usually difficult to achieve; and secondly, price movements still need human judgement and action, which still need to be translated into machine actions [2] . In recent years, there has been several approaches in using a DRL agent. In 2017, a CNN [2] was proposed to predict a price vector and a policy gradient network outputs the probability of action, with a goal to maximize portfolio value. Another approach is using a DRL agent to find the best trading strategy to maximize short-term profits, such as swing trading, day trading, scalping or position trading [4]. In our paper, we implement the FinRL library [1] for automated cryptocurrency selection and trading to maximize profits over any period of time. The library comes with different DRL agents and we compare the performance of each.

B. The FinRL library

The FinRL library [1] allows a more flexible approach to implementing DRL algorithms on various financial structures and environments. It consists of three layers: applications, agents, and market environments. The agent layer interacts with the environment layer with exploration-exploitation. Lower layers provides APIs for upper layers, making the lower layer transparent to the upper layer. As of today, the library has extensive implementation examples and detailed

tutorials on using agents for the stock market and Forex environments, but the current implementation on cryptocurrencies is uncompleted. We propose to fill the gap and implement our cryptocurrency environment, as well as a comparison of the performance of each agent from three different DRL librairies. Eventually, we test our agents in different market conditions and backtest the results.

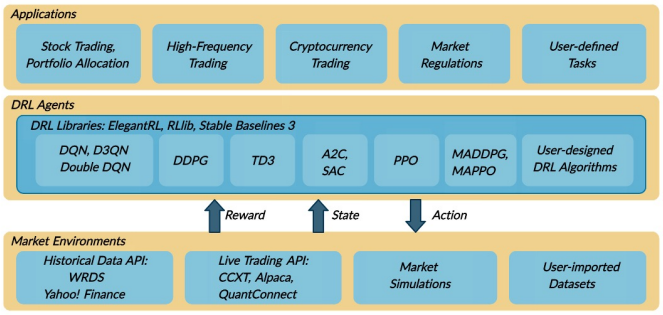


Fig. 1. The three-layer architecture

C. Actor Critic

The Actor Critic model is a 'hybrid' method of value-based and policy-based methods that makes an update at each time step instead of waiting for the end of an episode and calculate total rewards. The policy updates and at each step:

$$\Delta\theta = \alpha * \nabla_\theta * (log\pi(S_t, A_t, \theta)) * Q(S_t, A_t)$$

It uses two neural networks:a Critic that measures how good the action taken is (value-based), and an Actor updates policy distribution suggested by the critic (policy-based)

D. PPO

The Proximal Policy Optimization (PPO) is based on Advantage Actor Critic architecture and particularly solves the issue of having a too large policy update using a regularization method, hence improving the stability of the actor training by limiting the policy update at each step. With π being the policy (actor), we calculate the impact of actions using the probability of actions under the previous policy over that probability under the current policy $r = \pi_{new}(s)(a)/\pi_{old}(s)(a)$. In our case, we expect the PPO to fit well for high dimensional problems such as multiple stock or cryptocurrency data and its ability to mimic a trader by optimising cumulative long term reward.

E. DDPG

The Deep Deterministic Policy Gradient (DDPG) algorithm [5] is a model-free off-policy algorithm using the actor-critic framework. It combines Deep Q Network (DQN) and Policy Gradient. Unlike DQN, it is able to deal with a continuous action space instead of discrete, which allows for trading over a large set of cryptocurrencies, while "deterministic" in means that the actor computes the action directly instead of a probability distribution over actions, which is more desirable in production. Because we will mainly work on trading across multiple cryptocurrencies, or "portfolio allocation", this is a continuous decision problem. The other advantage of DDPG is that it is simpler to implement than other state-of-the-art algorithms.

III. Problem Definition

In order to use reinforcement learning algorithms, first a clear definition of the agent environment and objective to maximize has to be done. The definition of the problem is inspired from this work [8]

A. The model

- State To define the state of the agent at each time step t, we define s_t as a vector of size $2 \times 4 \times lookback$ where $lookback$ is the number of previous days included in the definition of the state. The components of the states are:
  - b_t : the available amount of money available at time t
  - $stock_t$: number of currency owned by the agent at time t
  - concatenation of four technical indicators for every $lookback$ previous days. The four technical indicators used are the Moving Average Convergence Divergence (MACD), the Relative Strength Index (RSI), and the Commodity Channel Index (CCI), Directional Movement Index (DMI).
- Action The agent can make three main actions: Buy, Sell or Hold. Each of this action are represented with a discrete value: -1 to sell, 0 to hold and 1 to buy. Since we let the agent buy or sell one more than one currency at each time for each time step t, $a_t \in -k, -(k-1), ... -1, 0, 1, ..., k-1, k$ where k represents the maximum number of currency exchanged allowed in one trade. Since, the Bitcoin is particularly expensive more than 34k$ in 2022, we extend the definition above to the continuous space $[-1, 1]$.
- Reward The reward for each time step $r_t(s_t, a_t, s_{t+1})$ is defined as $v_t - v_{t+1}$. Where v_t represents the total value of the portfolio that means the sum of the value of the currencies bought plus the available balance. The policy π objectif of the MDP is to maximize the discounted cumulative returns written as:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...$$

B. Training Dataset

The dataset used to train the agent is pulled from the www.binance.com website. The training dataset can be from any fixed period and with any given time-interval between successive readings. The dataset used to test the trained agent is taken from a separate time duration to assess the generalizability of the learned policy. The data provided by binance.com consists of the price and technical indicators of all currencies at a given time.

IV. Experiment and Results

A. Policy Evaluation using different Hyperparameters

1) Policy Evaluation using different DRL models: Three different DRL agents were used to learn an optimized policy

on the dataset obtained from the binance.com website for the year 2021 with a time-step of 1 hour. The trained agents were then subsequently used to trade on the test dataset. The test dataset used was for the duration of the first four months of 2022.

The three models chosen were Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradients (DDPG) and Soft Actor-Critic (SAC). Proximal Policy optimization is an on-policy method, whereas the other two are off-policy. DDPG model is specifically chosen because of its utility on a high dimensional, continuous action space. And SAC model uses entropy regularization to find a balance between maximizing returns and including a random element in the policy for possible exploration.

For the given time duration, the Soft-A2C agent performed the best. The returns are less than one because the current market is down-trending. The agent cannot make a profit if the inherent trading value of the crypto-currency, as determined by the market, is going down. We decided to test the agent on the latest data to have a better idea of its potential. However, we see that the agent achieved returns of upto 12 (profit upto 11 times the investment), on the training data.

Another consideration is the generalizability of the policy learned on the training dataset to the test dataset. The agent should be able to utilize the technical indicators to formulate a good policy, but the technical indicators alone are not enough to model the complexity and uncertainties of the currency market.

Nevertheless, we see that the model performed better than the two simplistic alternative strategies of investing everything in BTC or investing equally in all currencies. Given an up-trending crypto market, we will see greater returns on investment. The returns obtained using the trained models on the test data from Jan-April, 2022 is presented below.

TABLE I
POLICY EVALUATION: COMPARISON OF DIFFERENT DRL MODELS

Sr.	Model	Returns
1	PPO	0.7195
2	DDPG	0.6994
3	SAC	0.7435
4	Equal Weight Portfolio	0.6621
5	BTC	0.6074

2) Policy Evaluation using Different States: The FinRL environment provides us with a state which is based on the following metrics in the current time-step:

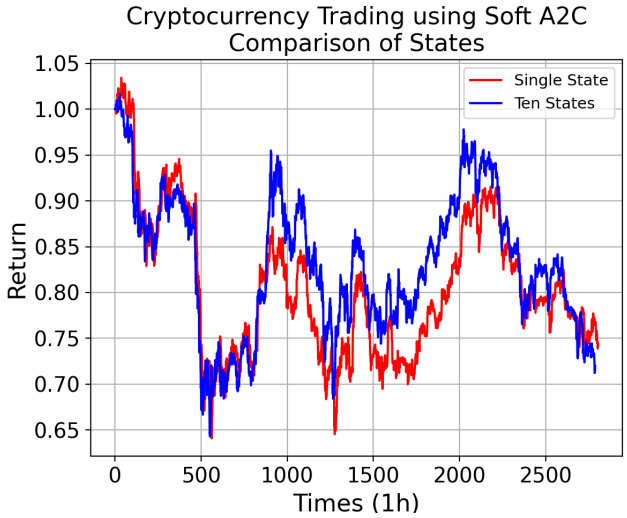
- Current assets: in terms of USD cash and crypto-currency
- Technical indicators

The Reinforcement Learning algorithm takes the state as an input and outputs an action. We make a naive assumption that if we give the agent states at multiple time-steps as an input, it will be able to learn a better policy. To this end, we modify the FinRL trading environment to provide the assets and technical indicators of previous ten time-steps. We use SAC agent and use the same training and test data for comparison. The results are presented in the following table.

TABLE II
POLICY EVALUATION: SINGLE-STEP STATE VS. TEN-STEP STATE

Sr.	Model	Returns (Single state)	Returns (Ten states)
1	PPO	0.7195	0.7067
2	DDPG	0.6994	0.6712
3	SAC	0.7435	0.7201
4	Equal Weight Portfolio	0.6621	0.6621
5	BTC	0.6074	0.6074

We see that the model with a single state input gave better episodic return than the multiple state input. However, if we look at the returns of the policy over the preceding time-steps, we see that for most of the time-steps the multiple-state model gave better returns. Given this, more experimentation is needed to conclusively decide which of the models would generalize better over a longer time-period.



3) Policy Evaluation using Temporally Congruent Training Dataset: We train two models using two different datasets. First dataset is from the whole 2021 year crypto-currency prices and technical indicators data. The second dataset is just a subset of the first dataset, it only contains data from the first four months. Two models are trained on these two datasets and then we do policy evaluation using the same test dataset i.e the first four months of 2022. The results obtained for the first dataset have already been tabulated above and the results for the second dataset are below.

We see that all the models give better returns when they are trained on the second dataset, even though it is limited to just four months. We may hypothesize that the policy learned

TABLE III
POLICY EVALUATION: USING CONGRUENT TRAINING DATASET

Sr.	Model	Returns
1	PPO	0.7256
2	DDPG	0.7208
3	SAC	0.7526
4	Equal Weight Portfolio	0.6621
5	BTC	0.6074

using this dataset generalizes better to the similar period in year 2022. More thorough experimentation is required to come to a definite conclusion.

4) Financial Metrics for Evaluation of Trading Policies: There are several financial indicators which are used to evaluate a trading policy. Some of them are listed below.

- Sharpe Ratio: the return of an investment compared to its risk
- Calmar Ratio: It is a function of the fund's average compounded annual rate of return versus its maximum drawdown.
- Max. Drawdown: The max. difference between the peak and trough in total assets during trading
- Cumulative Returns: Running total returns on investment
- Volatility: The rate at which the price of a currency increases or decreases over a time period

The evaluation of each DRL agent's policy against the baseline statistics of Bitcoin is given in Table IV.

B. Use of a new architecture and reward design with the PPO algorithm

The next step of the project was to design and adapt our own PPO methods on the cryptocurrency environment. This step was motivated by the need of a new architecture for the critic and actor network. Indeed, the elegantrl, rllib and stable_baseline3 libraries are not flexible at this point and allow only a full fully connected architecture and a convolutional architecture (adapted to Atari games). Because the technical indicators at the time step t depends of the technical indicators of previous time steps, we add a LSTM to the new architecture of the critic and actor networks. The LSTM will treat the technical indicators as an ordered sequence. The PPO implemented is an adaptation to the crypto environment of the PPO implemented during the lab13 of this semester.

Then, we change the reward with the Sharpe Ratio of the policy. The new reward is defined as:

$$r_t = \frac{mean(R_t)}{std(R_t)}$$

It has different advantages over the simple difference of wealth between two time steps. First, this reward is risk-adjusted that means big variation of return is penalized. Next, if the price skyrocket between two time steps and our agent make a good or bad action. The consequence of the action on the loss function will be bigger than the same action in a "normal" situation. To sum up, this new reward is more stable than the previous one and it reduces the variance of the policy.

We train three agents on 20 iterations with three different architecture and in the Fig 1 is the episode reward over the time steps for three of them. The first agent have three stack fully connected layers, the second have the LSTM layer with two fully connected layers and the third one has stacked fully connected layers and the Sharpe ratio as reward. *The models are trained to trade the BTC on 10 days and tested on 1 day with a time interval of 5 minutes between trades. During the testing the return of the third agent was computed with the previous reward formulation.*

Fig. 1. Episode Reward, blue: linear architecture, orange: lstm architecture, red: linear architecture and Sharpe ratio reward

As we can see above both methods that have modifications converge to a bigger return than the classic one. That means that the modifications made help the agent to reach a bigger return. The maximum return get by the agents is around 9%. Moreover,the LSTM architecture is the one that converge the fastest to the maximum reward found.

Add the LSTM layer comes with a computational cost, one iteration with the LSTM architecture takes almost ten times more the one required with the linear architecture.

The important observations made are: the agent can benefit of a LSTM layer inside the architecture but it will make the training slower, the Sharpe ratio is a better reward than the variation of portfolio value.

C. Transfer learning of trading strategy

Finally, we try to see if transfer learning is possible between cryptocurrency. To do so, we trained one SAC methods to trade Bitcoin on three months and test it on one week with a time interval between trade of 5 minutes. Then, we finetune the model with 100 iterations on the Ethereum on the same period.

The Figure 2, shows the return of the agent on both cryptocurrencies. We can see on it that the agent have a bigger return on the ETH than on the BTC.

TABLE IV
POLICY EVALUATION USING FINANCIAL METRICS (TEST PERIOD APR-2021)

Sr	Models	Cumulative Returns (%)	Annual Volatility (%)	Sharpe Ratio	Calmar Ratio	Stability	Max Drawdown (%)	Daily Value at Risk (%)
1	SAC	26.10	12.81	0.41	0.22	0.43	-20.90	-1.60
2	PPO	89.16	18.40	0.75	0.43	0.66	-30.20	-2.26
3	DDPG	26.54	17.54	0.34	0.15	0.06	-31.52	-2.20

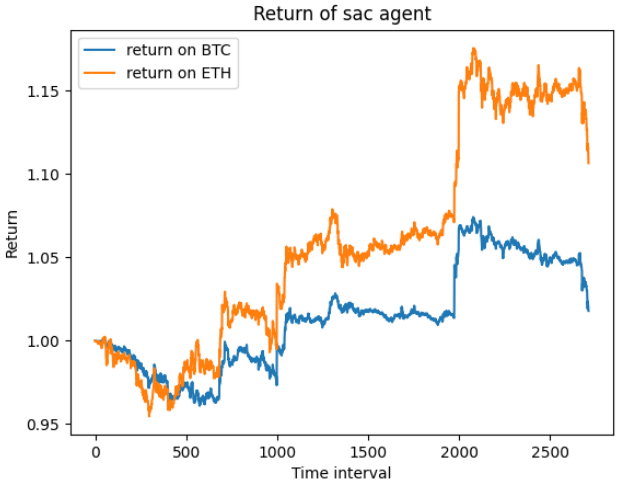


Fig. 2. Return of the agent on BTC and ETH

The Figure 3 and 4 show the position of the agent according to the price of both currency during the test period. On these figures, we can see that even if the agent have a better return while trading ETH the trading strategy on this currency is far from an optimized strategy. Indeed, the agent use the Buy and Hold strategy, although better positions can be made. Since the price of the Bitcoin is largely superior than the price of the Etherum, while finetuning on the Ethereum the agent can ”think” that the price is low and so decide to buy as much currency as he can and hold it. Also, we can notice that in a global aspect the variation of the price of the Bitcoin and the Ethereum are similar so a transfer learning of trading strategy between cryptocurrencies is probably possible.

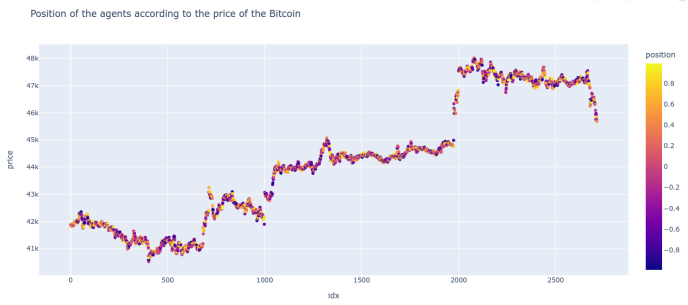


Fig. 3. Position of the agent over BTC variation

The behaviour of the agent can be due to a non-normalization of the data, but in the environment every quantities are normalized arbitrarily by a power of 2. This kind of normalization is not satisfactory because it doesn't reduce at an equivalent scale the gap between two values of two populations.

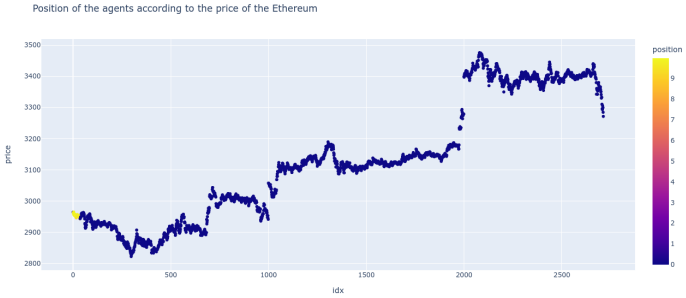


Fig. 4. Position of the agent over ETH variation

V. CONCLUSION

To conclude, during this project we adapted a library named FinRL to the crypto-currencies trading market. First, we trained several agents from different reinforcement learning methods and compared them with a simple trading strategy such as Buy and Hold Bitcoin and Equal weight Portfolio. After our experimentation, we confirmed that Deep Reinforcement learning methods were able to perform well in multi-crypto trading, we tried to improve upon the existing models. To do so, we implemented a customized version of the PPO method with an LSTM layer in the actor and critic neural network architecture. The addition of this layer will enable us to capture the sequential information from the previous states. After that, we modified the reward signal for training the agent. Finally, we studied the possibility of transfer learning of trading strategies between different crypto-currencies.

VI. LIMITATIONS AND FUTURE WORK

The first limitation of the project is that we based most of the analysis on the return only while in the real-world this is not the only factor considered. Next, the reward function and market state representation chosen is too simple and does not model all the complexities of a real financial market. Some possible future work directions for this project are the comparison of the trading agent with more advanced trading algorithm, the inclusion of market uncertainty factors peculiar to the crypto market. For example, the inclusion of the gas fees applied on the Ethereum blockchain, for faster transaction.

Finally, to mimic the behaviour of a real trader, we would like to implement a sentiment analysis model which will take as input the news related to economy and output a market sentiment score which represents how much this news can impact the financial market.

REFERENCES

[1] Liu, Xiao-Yang Yang, Hongyang Gao, Jiechao Wang, Christina. (2021). FinRL: deep reinforcement learning framework to automate trading in quantitative finance. 1-9. 10.1145/3490354.3494366.

[2] Z. Jiang and J. Liang, "Cryptocurrency portfolio management with deep reinforcement learning," 2017 Intelligent Systems Conference (IntelliSys), 2017, pp. 905-913, doi: 10.1109/IntelliSys.2017.8324237.

[3] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," Applied Stochastic Models in Business and Industry, 2016. [Online].

[4] Sattarov O, Muminov A, Lee CW, Kang HK, Oh R, Ahn J, Oh HJ, Jeon HS. Recommending Cryptocurrency Trading Points with Deep Reinforcement Learning Approach. Applied Sciences. 2020; 10(4):1506. https://doi.org/10.3390/app10041506

[5] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. and Wierstra, D.. Continuous control with deep reinforcement learning. ICLR 2016.

[6] Hongyang Yang, Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy, 2020.

[7] Tidor-Vlad Pricope, Deep Reinforcement Learning in Quantitative Algorithmic Trading: A Review

[8] GORDON RITTER, MACHINE LEARNING FOR TRADING, 2017

[9] https://www.quantrocket.com/codeload/quant-finance-lectures/quant_finance_lectures/Lecture33-Portfolio-Analysis-with-pyfolio.ipynb.html

[10] https://lilianweng.github.io/posts/2018-04-08-policy-gradient/

[11] M.F.M Osborne, Brownian Motion in the stock market