Cryptocurrency Day-Trading with Deep Reinforment Learning and an Ensemble Method Implementation

Jiachen Huang [*] [1] **Yifei Zhang** [*] [1]

Abstract

Deep reinforcement learning is a very popular topic and has shown strong potential in financial trading area. With the development of blockchain technology, cryptocurrency is widely invested. More and more people and institutions start to pay attention to and trade cryptocurrency. Inspired by these two topics, we tried different deep reinforcement learning (DRL) models to build an automatic cryptocurrency trading program. We also developed an ensemble learning model with dynamically in-cooperate with pre-trained DRL agents based on the Sharpe ratio, which achieves high investment return and is more robust. In our report, we will show our new proposed model and its performance on cryptocurrency trading.

1. Introduction

Deep reinforcement learning (DRL) has been introduced quantitative finance in recent years. However, there is limited exploration of implementing DRL-based quantitative trading to obtain a portfolio balance to win in the market. In this paper, we carried out a DRL based trading pipeline implement some state-of-art DRL algorithms on multiple cryptocurrency portforlio. In particular, we focuses on day trading, as its name implies, the purchase and sale of cryptos can be done during one single day. Average crypto traders are difficult to avoid unmanageable risks to retain high turnover rate. The experience of traders and their commitment determines good points for trading and it is not easy to achieve. Therefore, our project incorporates cryto tradition patterns and an novel Sharpe ratio based ensemble mechanism, provides an alternative solution for crypto trading, especially for those risk-averse traders.

In this part, we will introduce crypto and related research regarding quantitative trading and cryptocurrency trading.

1.1. Cryptocurrency

Cryptocurrency is a form of payment that allows goods and services to be exchanged online. Many companies issue their own currency, often called tokens, which can be used exclusively to exchange goods or services provided by the company. Think of them as arcade tokens or casino chips. You need to exchange real money for cryptocurrency to access goods or services. Cryptocurrencies work using a technology called blockchain. Blockchain is a technology spread across many computers to manage and record transactions. Part of the technology's appeal is its safety (cry). At present, there are many cryptocurrencies traded in the market, including Bitcoin, Ethereum, XRP, Tether, Cardano, Polkadot, Stellar, USD Coin, Dogecoin, Chainlink, and so on. We will introduce two main types of cryptocurrency: Bitcoin and Ethereum, in the following part.

1.1.1. Bitcoin

Bitcoin is a decentralized digital currency created in January 2009. It follows the ideas set out in a white paper by the mysterious and pseudonymous Satoshi Nakamoto. The identity of the person or persons who created the technology is still a mystery. Bitcoin offers the promise of lower transaction fees than traditional online payment mechanisms do, and unlike government-issued currencies, it is operated by a decentralized authority(Bit).

1.1.2. Ethereum

Ethereum is a blockchain platform with its own cryptocurrency, called Ether (ETH) or Ethereum, and its own programming language, called Solidity. As a blockchain network, Ethereum is a decentralized public ledger for verifying and recording transactions. The network's users can create, publish, monetize, and use applications on the platform, and use its Ether cryptocurrency as payment. Insiders call the decentralized applications on the network ”dApps.” As a cryptocurrency, Ethereum is second in market value only to Bitcoin, as of December 2021(Eth).

[*] Equal contribution [1] Data Science Institute, Columbia University in the City of New York, New York, NY, USA. Correspondence to: Jiachen Huang <jh4336@columbia.edu>, Yifei Zhang <yz3925@columbia.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

1.2. Related Work

There are many trading algorithms and strategies development by researchers. Apart from traditional quantitative trading methods, deep reinforcement learning (DRL) has shown huge potentials in building financial market simulators through multi-agent systems to do trading. Many quantitative trading algorithms and strategies on stock have been developed and widely used. However, cryptocurrency trading differs from stock trading in that it is more irregular and can be traded 24 hours a day, seven days a week compared to stock exchange trading, methods on cryptocurrency trading is still waiting to be explore.

Jiang et al(Jiang and Liang, 2017) presented a model-less convolutional neural network with historic prices of a set of financial assets as its input, outputting portfolio weights of the set to do cryptocurrency portfolio management. Sattarov et al(Sattarov et al.) developed an application that observes historical price movements and takes action on real-time prices.

Lussange et al(Lussange) proposed a market simulation model using multi-agent reinforcement learning. Although the feasibility of DRL based market simulation has been demonstrated, only a small number of DRL agents have been used. The potential of DRL based market simulators has not been fully explored. Liu et al(Liu et al., 2020) proposed FinRL-Meta framework. FinRL-Meta framework was developed by AIFinannce. It is a universe of near real-market environments for data-driven financial reinforcement learning. First, they apply the DataOps paradigm to the data engineering pipeline, providing agility to agent deployment. They offer a unified and automated data processor for data accessing, data cleaning and feature engineering. Second, they build hundreds of near real-market DRL environments for various trading tasks such as high-frequency trading, cryptocurrencies trading, stock portfolio allocation, etc.. The environments are directly connected to our data processor. High-quality large datasets can be generated efficiently and encapsulated into our environments. Third, to accelerate the training process of DRL agents in large datasets, they utilize thousands of GPU cores to perform multiprocessing training(Vo and Yost-Bremm).

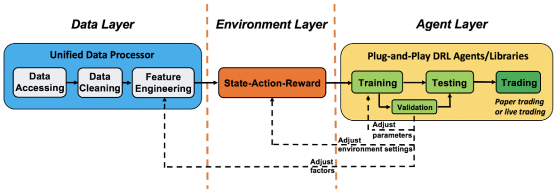

Figure 1. Overview of FinRL-Meta.

2. Approach

2.1. Overview

We tried different DRL models. There are many alternative algorithms to train the agent. Here we select four algorithm as baseline model. They are Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG), The twin-delayed deep deterministic policy gradient (TD3), Soft Actor Critic (SAC), Advantage Actor Critic (A2C). We also introduced an ensemble pipeline.

2.1.1. PPO

PPO is a first-order optimization that simplifies its implementation (Schulman et al., 2017). It defines the probability ratio between the new policy and old policy and we can call it as $r(\theta)$.

$$r(\theta) = \frac{\pi_\theta(a \mid s)}{\pi_{\theta old}(a \mid s)}$$

Now we can modify the objective function of Trust Region Policy Optimisation as below:

$$J(\theta)^{TRPO} = E\left[r(\theta)\hat{A}_{\theta old}(s, a)\right]$$

Without adding constraints, this objective function can lead to instability or slow convergence rate due to large and small step size update respectively. Instead of adding complicated KL constraint, PPO imposes policy ratio, r(θ) to stay within a small interval around 1. That is the interval between 1-ϵ and 1+ϵ. ϵ is a hyper-parameter and in the original PPO paper, it was set to 0.2. Now we can write the objective function of PPO as $J^{\text{CLIP}}(\theta) = E\left[\min\left(r(\theta)\hat{A}_{\theta_{old}}(s, a), clip(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\theta_{old}}(s, a)\right)\right]$

In the above equation, the function clip truncates the policy ratio between the range [1-ϵ, 1+ϵ]. The objective function of PPO takes the minimum value between the original value and the clipped value. Similar to what we discussed in vanilla policy gradient section above, positive advantage function indicates the action taken by the agent is good. On the other hand, a negative advantage indicated bad action. For PPO, in both cases, the clipped operation makes sure it won't deviate largely by clipping the update in the range(PPO).

2.1.2. DDPG

Deep Deterministic Policy Gradient (DDPG) is an algorithm which concurrently learns a Q-function and a policy (Casas, 2017). It uses off-policy data and the Bellman equation to learn the Q-function, and uses the Q-function to learn the policy.

This approach is closely connected to Q-learning, and is motivated the same way: if you know the optimal action-value function $Q^*(s, a)$, then in any given state, the optimal action $a^*(s)$ can be found by solving

$$a^*(s) = \arg\max_a Q^*(s,a)$$

DDPG interleaves learning an approximator to $Q^*(s,a)$ with learning an approximator to $a^*(s)$, and it does so in a way which is specifically adapted for environments with continuous action spaces. But what does it mean that DDPG is adapted specifically for environments with continuous action spaces? It relates to how we compute the max over actions in $\max_a Q^*(s,a)$.

When there are a finite number of discrete actions, the max poses no problem, because we can just compute the Q-values for each action separately and directly compare them. (This also immediately gives us the action which maximizes the Q-value.) But when the action space is continuous, we can't exhaustively evaluate the space, and solving the optimization problem is highly non-trivial. Using a normal optimization algorithm would make calculating $\max_a Q^*(s,a)$ a painfully expensive subroutine. And since it would need to be run every time the agent wants to take an action in the environment, this is unacceptable.

Because the action space is continuous, the function $Q^*(s,a)$ is presumed to be differentiable with respect to the action argument. This allows us to set up an efficient, gradient-based learning rule for a policy $\mu(s)$ which exploits that fact. Then, instead of running an expensive optimization subroutine each time we wish to compute $\max_a Q^*(s,a)$, we can approximate it with $\max_a Q^*(s,a) \approx Q(s,\mu(s))$(DDP).

2.1.3. TD3

While DDPG can achieve great performance sometimes, it is frequently brittle with respect to hyperparameters and other kinds of tuning. A common failure mode for DDPG is that the learned Q-function begins to dramatically overestimate Q-values, which then leads to the policy breaking, because it exploits the errors in the Q-function. Twin Delayed DDPG (TD3) is an algorithm that addresses this issue by introducing three critical tricks:

- Trick One: Clipped Double-Q Learning. TD3 learns two Q-functions instead of one (hence "twin"), and uses the smaller of the two Q-values to form the targets in the Bellman error loss functions.
- Trick Two: "Delayed" Policy Updates. TD3 updates the policy (and target networks) less frequently than the Q-function. The paper recommends one policy update for every two Q-function updates.
- Trick Three: Target Policy Smoothing. TD3 adds noise to the target action, to make it harder for the policy to exploit Q-function errors by smoothing out Q along changes in action.

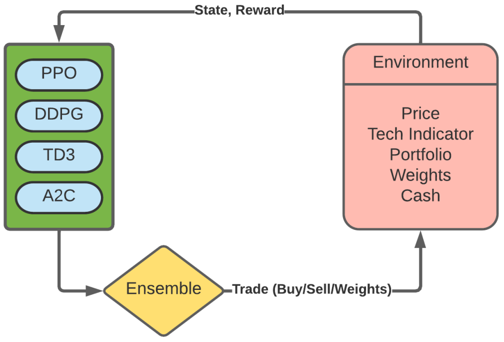

Figure 2. Model Structure

Together, these three tricks result in substantially improved performance over baseline DDPG(DDP).

2.1.4. A2C

An actor-critic algorithm is a policy gradient algorithm that uses function estimation in place of empirical returns Gt in the policy gradient update (Babaeizadeh et al., 2016). The actor is the policy π_θ and the critic is usually a value function V_ϕ. The actor is trained using gradient ascent on the policy gradient, and the critic is trained via regression. The regression target can either be the empirical returns G_t, or can be the Bellman target $r_t + \gamma V_\phi(s_t)$. An advantage actor-critic is just an actor-critic that uses the advantage $\hat{A}(s_t)$ instead of V^{π_θ}(A2C).

2.2. Ensemble Model

As discussed in the introduction, we would like to implement and test a DRL based framework that employees multiple agents and achieve an ensemble trading algorithm based on Sharpe Ratio. After preliminary test, we choose four candidate algorithms for comparison and to enter our ensemble mechanism: PPO, DDPG, TD3 and A2C. The general structure of our pipeline can be found in the following Figure.2 and the detailed implementation and explanation of the ensemble procedure can be found the later sections.

3. Experiment

In this part, we shows the cryptocurrency trading results of our program.

3.1. Experiment Settings

We select ten popular cryptocurrencies on the market. They are Bitcoin (BTC), Ethereum (ETH), Cardano (ADA), Binance Coin (BNB), Ripple (XRP), Solana (SOL), Polkadot (DOT), Dogecoin (DOGE), Avalanche (AVAX), Uniswap (UNI).

3.2. Environment

One of the preparation steps for our deep reinforcement learning frame is the environment construction. We choose to formulate the environment based on OpenAI gym, finRL, as well as TA-Lib for the quantatitive indicator processing (Brockman et al., 2016).

In particular, we focus on historical assets prices, portfolio, and technical indicators. We employee a set of technical indicators that are specifically chosen for they can better describe the market trend and trading timing. Those indicators are CCI, DX, MACD, and RSI(Maitah et al., 2016) (Gurrib et al., 2018). While you can find detailed introduction of those indicators in the next section, we are going to walk through the basic construction of our environment. Besides basic construction such as dimensions etc, we include total asset, portfolio, price and tech indicators to represent the state space. Each component is defined as:

- a_t: total balance at time step t.
- p_t: current price .
- w_t: portfolio weights
- $MACD_t$: Moving Average Convergence Divergence is calculated using close price.
- DX_t: Directional Index (DX) is calculated using high, low and close price
- RSI_t: Relative Strength Index (RSI) is calculated using close price. RSI quantifies the extent of recent price changes.
- CCI_t: Commodity Channel Index compares current price to average price over a time window to indicate a buying or selling action.

In our implementation, the computation of tech indicators are done by introducing the open-source library TA-Lib and FinRL.

3.3. Metrics

3.3.1. Moving Average Convergence Divergence (MACD)

Moving average convergence divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-period exponential moving average (EMA) from the 12-period EMA.

The result of that calculation is the MACD line. A nine-day EMA of the MACD called the ”signal line” is then plotted on top of the MACD line, which can function as a trigger for buy and sell signals. Traders may buy the security when the MACD crosses above its signal line and sell—or short—the security when the MACD crosses below the signal line. Moving average convergence divergence (MACD) indicators can be interpreted in several ways, but the more common methods are crossovers, divergences, and rapid rises/falls(MAC).

MACD=12-Period EMA26-Period EMA.

$$EMA_{Today} = \left(Value_{Today} * \left(\frac{Smoothing}{1+Days}\right)\right)$$
$$+ EMA_{Yesterday} * \left(1 - \left(\frac{Smoothing}{1+Days}\right)\right)$$

where: EMA=Exponential moving average

3.3.2. Relative Strength Index (RSI)

The relative strength index (RSI) is a momentum indicator used in technical analysis that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset. The RSI is displayed as an oscillator (a line graph that moves between two extremes) and can have a reading from 0 to 100. The indicator was originally developed by J.

Traditional interpretation and usage of the RSI are that values of 70 or above indicate that a security is becoming overbought or overvalued and may be primed for a trend reversal or corrective pullback in price. An RSI reading of 30 or below indicates an oversold or undervalued condition.

$$RSI_{stepone} = 100 - \left[\frac{100}{1 + \frac{Averageagain}{Averageloss}}\right]$$

3.3.3. Commodity Channel Index (CCI)

The Commodity Channel Index (CCI) is a momentum-based oscillator used to help determine when an investment vehicle is reaching a condition of being overbought or oversold.

Developed by Donald Lambert, this technical indicator assesses price trend direction and strength, allowing traders to determine if they want to enter or exit a trade, refrain from taking a trade, or add to an existing position. In this way, the indicator can be used to provide trade signals when it acts in a certain way(CCI).

$$\text{CCI} = \frac{TypicalPrice - \text{MA}}{0.015 \times MeanDeviation}$$

- Typical Price= $\sum_{i=1}^{P}((\text{ High }+\text{ Low }+\text{ Close }) \div 3)$
- P=Number of periods
- MA=Moving Average
- Moving Average= $\left(\sum_{i=1}^{P} TypicalPrice\right) \div P$
- Mean Deviation= $\left(\sum_{i=1}^{P} \mid \text{Typical Price-MA} \mid\right) \div P$

3.3.4. Directional Movement Index (DX)

The Directional Movement Index (DX) is an intermediate result in calculating of the Average Directional Index (ADX) that was developed by J. Welles Wilder to evaluate the strength of a trend and to define a period of sideway trading. The Directional Movement index is based on the positive and negative Directional indicators and is used to spot crossovers of positive and negative directional indicators - when bullish/bearish change of power takes place(DX).

3.3.5. Sharp Ratio

Sharpe ratio has been one of the most referenced risk/return measures used in finance, and much of this popularity is attributed to its simplicity. Most finance people understand how to calculate the Sharpe ratio and what it represents. The ratio describes how much excess return you receive for the extra volatility you endure for holding a riskier asset. Remember, you need compensation for the additional risk you take for not holding a risk-free asset(sha).

$$S(x)=\frac{\left(r_{x}-R_{f}\right)}{StdDev\left(r_{x}\right)}$$

Where:

- $x=$ The investment
- $r_{x}=$ The average rate of return of x
- $R_{f}=$ The best available rate of return of a risk-free security (i.e. T-bills)
- $StdDev\left(r_{x}\right)=$ The standard deviation of r_{x}

3.4. Ensemble Implementation

Considering the high-volatility of crypto trading, we would like to build a more robust trading framework, in particular for risk-averse players. One ensemble strategy was proposed to dynamically in-cooperate with pre-trained DRL agents based on the Sharpe ratio in previous trading sessions. First of all, our pipeline trains a given number of models with customized agent arguments. In this implementation, we requires the same environment for all agents. But we can even realise the requirement of same environment as long as they can give same prediction steps(Yang et al., 2020).

Then, we could call our built-in function that evaluates all agents by their performance in previous training window and to pick the best performing on achieving the highest Sharpe ratio.

The next step is to switch and ensemble the chosen agent for next steps, until the evaluation generates a different ensemble combinations. Choosing different agents can make use of each agents, with different models or trained on different time periods, to gain a better response to short-term trends. Some agents are more adjusted to volatile sessions, while others are good at finding best target to trade. The higher the Sharpe ratio, the better a strategy is in a given trading session.

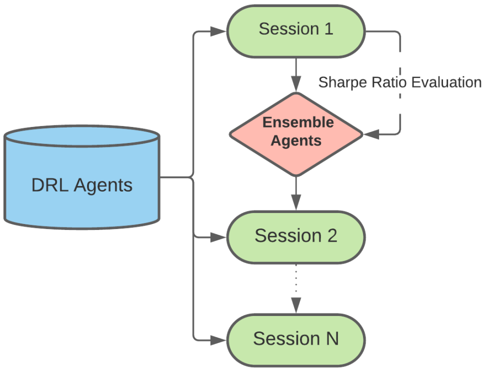

Figure 3. Ensemble Structure

3.5. Trading Performance

Model	Return	Sharp Ratio
PPO	**0.124**	0.0116
DDPG	0.076	0.0069
TD3	0.038	0.0020
A2C	0.059	0.0050
Ensemble	**0.117**	**0.0134**
Benchmark	0.017	0

Table 1. Comparison on Trading Algorithms

To proceed our trading performance test, we run the model for several tests in different time periods. Then we obtained a reasonable result by choosing proper hyper-parameters that can generate stable output. From Table 1, it is clear that some agents are more risk-free while others may have higher variances. PPO has the highest return of 12.4%, which is very significant in this short time. At the same time, DDPG ranks the second with a return of 7.6% and

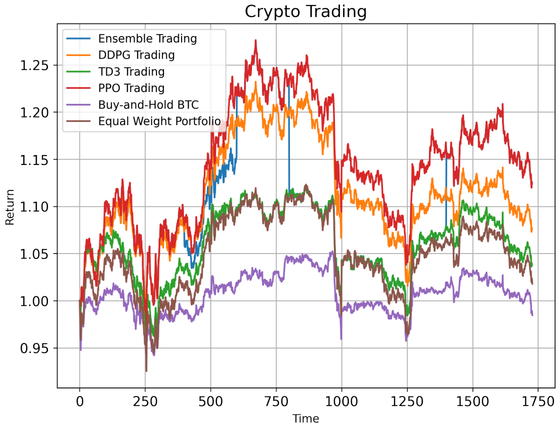

Figure 4. Crypto Trading Performance

Sharpe ratio of 0.0069. A2C and TD3 are the third and the fourth agents respectively with return rate near 6% and 4%. Nevertheless, all agents show good performance that surpass the benchmark of equal weight portfolio, not to mention the Buy-and-Hold BTC strategy. In the short period of time, it is reasonable to conclude that the DRL based trading model performs better than naive trading.

Similarly, our Figure 4 demonstrates that PPO and our ensemble strategies. outperform other trading strategies. Due to the limited time of implementation, actually the ensemble strategy has a slightly lower return compared with the more aggressive PPO. But the ensemble did have a better volatility that can make it works smoothly. It is notable that the ensemble strategy also outperforms other agents explicitly. It is not surprising since the ensemble structure should avoid the worst performance in any given sessions. Therefore, the testing result illustrate that our models not only exceed naive trading benchmarks in day trading, but also secure a minimum performance by implementing our ensemble structure.

4. Conclusion

In our report, we compared different models on the task of cryptocurrency trading. From our experiment, PPO performed best on the return. We also developed an ensemble learning model with dynamically in-cooperate with pre-trained DRL agents based on the Sharpe ratio. Our new proposed ensemble model based on sharp-ratio is more robust than single DRL model.

For future work, we are trying to incorporate more models into our ensemble model, also we will use the ensemble model in the training process. There are some proposed ensemble method (Wiering and Van Hasselt, 2008) such as SUNRISE that could be introduced in this problem setting (Lee et al., 2020). In addition, more more parameter tuning and dedicated environment design will be highly valuable in improving the performance of our model. One way to do so is to explore more features for the state space such as adding transaction cost (Bao and Liu, 2019) and market indicators (Li et al., 2019).

5. Acknowledgement and Statement

Our repo contains main modules developed for our project. However, there are also many modifications in three other repos to fit our demand. You can find repo links in the README file.

Contributions: We would like to conclude that two team members have the equal contribution.

References

Actor-critic methods, advantage actor-critic (a2c) and generalized advantage estimation (gae). URL https://avandekleut.github.io/a2c/.

Bitcoin definition. URL https://www.nerdwallet.com/article/investing/cryptocurrency-7-things-to-know.

Commodity channel index (cci). URL https://www.investopedia.com/terms/c/commoditychannelindex.asp.

Twin delayed ddpg. URL https://spinningup.openai.com/en/latest/algorithms/td3.html.

Directional movement index (dx). URL https://www.investopedia.com/terms/e/ema.asp.

Ethereum. URL https://www.investopedia.com/terms/e/ethereum.asp.

Moving average convergence divergence (macd). URL https://www.investopedia.com/terms/m/macd.asp.

Proximal policy optimization. URL https://openai.com/blog/openai-baselines-ppo/.

What is cryptocurrency? here's what you should know. URL https://www.nerdwallet.com/article/investing/cryptocurrency-7-things-to-know.

Understanding the sharpe ratio. URL https://www.investopedia.com/articles/07/sharpe_ratio.asp.

Mohammad Babaeizadeh, Iuri Frosio, Stephen Tyree, Jason Clemons, and Jan Kautz. Reinforcement learning through asynchronous advantage actor-critic on a gpu. *arXiv preprint arXiv:1611.06256*, 2016.

Wenhang Bao and Xiao-yang Liu. Multi-agent deep reinforcement learning for liquidation strategy analysis. *arXiv preprint arXiv:1906.11046*, 2019.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Noe Casas. Deep deterministic policy gradient for urban traffic light control. *arXiv preprint arXiv:1703.09035*, 2017.

Ikhlaas Gurrib et al. Performance of the average directional index as a market timing tool for the most actively traded usd based currency pairs. *Banks and Bank Systems*, 13 (3):58–70, 2018.

Zhengyao Jiang and Jinjun Liang. Cryptocurrency portfolio management with deep reinforcement learning, 2017.

Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. SUNRISE: A simple unified framework for ensemble learning in deep reinforcement learning. *CoRR*, abs/2007.04938, 2020. URL https://arxiv.org/abs/2007.04938.

Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang, and Xiao-Yang Liu. Dp-lstm: Differential privacy-inspired lstm for stock prediction using financial news. *arXiv preprint arXiv:1912.10806*, 2019.

Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. *Deep RL Workshop, NeurIPS 2020*, 2020.

Lazarevich I. Bourgeois-Gironde Lussange, J.

Mansoor Maitah, Petr Procházka, Michal Cermak, and Karel Šrédl. Commodity channel index: Evaluation of trading rule of agricultural commodities. *International Journal of Economics and Financial Issues*, 6(1):176–178, 2016.

Otabek Sattarov, Azamjon Muminov, Cheol Won Lee, Hyun Kyu Kang, Ryumduck Oh, Junho Ahn, Hyung Jun Oh, and Heung Seok Jeon. Recommending cryptocurrency trading points with deep reinforcement learning approach. *Applied Sciences*. doi: 10.3390/app10041506. URL https://www.mdpi.com/2076-3417/10/4/1506.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Au Vo and Christopher Yost-Bremm. A high-frequency algorithmic trading strategy for cryptocurrency. *Journal of Computer Information Systems*. doi: 10.1080/08874417.2018.1552090. URL https://doi.org/10.1080/08874417.2018.1552090.

Marco A Wiering and Hado Van Hasselt. Ensemble algorithms in reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38 (4):930–936, 2008.

Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. *Available at SSRN*, 2020.