Quantitative Cryptocurrency Trading Based On Proximal Policy Optimization

Shitao Cai, Zhennan Chen, Xiang Wang, Hongyan Ye, Longjie Zhao
30920211154122 School of Informatics, Xiamen University, Xiamen, China
30920211154126 School of Informatics, Xiamen University, Xiamen, China
23320211154245 School of Informatics, Xiamen University, Xiamen, China
30920211154141 School of Informatics, Xiamen University, Xiamen, China
32920192291227 International College of Xiamen University, Xiamen, China

Abstract

In recent years, the cryptocurrency market has received widespread attention due to its high returns. Technicians realize that artificial intelligence technology can be used for quantitative trading of cryptocurrencies. However, the volatility and uniqueness of the cryptocurrency market make it face challenges. Our research uses deep reinforcement learning (DRL)—Proximal Policy Optimization(PPO) to automatically develop quantitative trading strategies through intelligent trading agents. And we also combined Long short-term memory(LSTM) for model optimization. In order to better simulate the real market environment, we collected high-frequency data in the cryptocurrency market for model training. After processing, our financial data will be in the form of a sparse matrix, which is of great help to the model to extract information. Experimental results demonstrate that our model can extract robust market features and be adaptive in different markets.

Introduction

Cryptocurrency is a rapidly growing asset, which was born in 2009 and gradually came into the public view after 2017, with a total market capitalization of over U.S. $1,540,000,000,000 on March 1st, 2021. It utilizes a decentralized technology called blockchain to get rid of the control of corporate entities like traditional currency. Nowadays, there are over 3000 kinds of cryptocurrency and more and more large technology enterprises and investment companies take it as an important asset allocation component.

Prior to the work of Jing-Zhi Huang (Lillicrap et al. 2015), the role of pure data-driven analysis in cryptocurrency prediction was not taken seriously, which indicates that bitcoin price can be predicted by analyzing technical indexes and big data, with little effect from fundamentals, providing a theoretical basis for machine-learning-based predictions. Since neural networks develop, it is possible to construct complex nonlinear functions and capture the long-term dependence of sequences. Neural networks have been boosting bitcoin predictions. References (Mallqui and Fernandes 2019)and(McNally, Roche, and Caton 2018) both build an artificial neural network (ANN) to predict the price of cryptocurrency, but ((McNally, Roche, and Caton 2018) concentrates on ensemble algorithms for direction prediction, rather than price prediction, which can not give references for high-frequency trading directly. Although in recent years of research, various models have achieved good results, they still have shortcomings: (1) Some popular algorithms, such as dilation causal convolution, self-attetion, etc., are not used. By introducing the most advanced structure, the forecast results may be significantly improved. (2) In the previous method, it is divided into two steps. Firstly, the model is used for prediction, and then the prediction signal is input into the relevant module for trading. This is a huge shortcoming.

Reinforcement learning (RL) seems to be the natural choice for optimal transaction execution problems because it enables trading agents to interact with the market and learn from its experience, and it makes fewer assumptions about price dynamics than closed-form solutions. RL can integrate forecasts and transactions into a model, which greatly improves efficiency, and can add some artificially defined constraints at will, which is very helpful to improve the effect of the model.

In this study, a novel framework of automatically generating high-frequency in our quantitative trading framework, there are three main components. The first part is the preprocessing of the data. We process the original data into a sparse signal matrix, and the sparseness of the data will make the model better extract features. The second part is the LSTM algorithm part. This part mainly deals with the dependence of time series data. Using LSTM is of great help to giving more accurate trading signals. The third part is the PPO algorithm part. Through the continuous interaction between the agent and the environment, the framework can select the optimal action, evaluate it according to the reward function, and then continuously optimize the subsequent behavior. Experiments have shown that our framework is very suitable for the case of large data noise such as the cryptocurrency trading market.

Related work

Reinforcement learning (RL) seems to be a natural choice for the optimal trade execution problem, as it enables the trading agent to interact with the market and to learn from its experiences and has fewer assumptions on the price dynamics than the closed-form solutions. Nevmyvaka, Feng, and Kearns have published the first large-scale empirical application of RL to optimal trade execution problems (Nevmyvaka, Feng, and Kearns 2006). Hendricks and Wilcox propose to combine the Almgren and Chriss model (AC) and RL algorithm and to create a hybrid framework mapping the states to the proportion of the ACsuggested trading volumes (Hendricks and Wilcox 2014).

To address the high dimensions and the complexity of the underlying dynamics of the financial market, (Ning, Lin, and Jaimungal 2018)adapt and modify the Deep Q-Network (DQN) (Ning, Lin, and Jaimungal 2018) for optimal trade execution, which combines the deep neural network and the Q-learning, and can address the curse of dimensionality challenge faced by Q-learning. Lin and Beling(Lin and Beling 2020) analyze and demonstrate the flaws when applying a generic Q-learning algorithm and propose a modified DQN algorithm to address the zero-ending inventory constraint. The policy gradient algorithm is a policy function, and then uses gradient descent to update the parameters of the network. The core idea of police gradient is through parameters θ to control the action strategy of the agent, which is expressed as $\pi(\theta)$ Theoretically, when the model is known, the value expectation of the strategy can be solved. However, usually, the environmental model is unknown, so the actual expected value can be obtained through the trajectory value of statistical data. The gradient rise (fall) method is adopted to optimize the strategy period value to achieve the goal of optimizing the strategy parameters. Based on (Mallqui and Fernandes 2019)(Munim, Shakil, and Alon 2019)appling the neural network auto-regression (NNAR) to complete the-next-day prediction and finds that NNAR is inferior to ARIMA in daytime prediction, demonstrating that naive neural networks are possibly not useful than traditional methods. With the development of Recurrent Neural Networks (RNNs), the long sequence prediction method has been developed unprecedentedly. LSTM provides a more credible method for long-time prediction. (McNally, Roche, and Caton 2018) compare the capability of capturing longer range dependencies between LSTM and SVR and proves LSTM is more suitable for time-series prediction. Deep learning algorithms have been widely exploited to explore the law of tendency in bitcoin price.

However, the researchers in previous research all use manually designed attributes which adds a further burden to the system development in the real-world, since feature engineering requires significant domain knowledge and efforts. We desire an end-to-end optimal trade execution system without feature engineering.

In this paper, We propose an end-to-end optimal trade execution framework which can account for temporal correlations, and also use a sparse signal matrix to enhance the data.

Proposed Solution

The Proximal Policy Optimization(PPO) algorithm is a policy gradient algorithm proposed by OpenAI (Schulman et al. 2017). PPO algorithm proposes a new objective function, which can be updated in a small batch in multiple training steps, and solves the problem that the step size is difficult to determine in the Policy Gradient Methods. If the step size is too small, the training time will be too long. If the step size is too large, the useful information will be masked by noise, or the performance will be disastrous and difficult to converge. The overall framework of reinforcement learning(Guan et al. 2020)is shown in Figure 1.

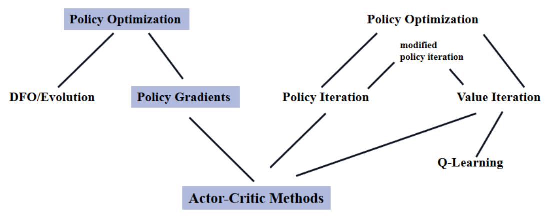

Figure 1: Overview of classes of reinforcement learning algorithms.

Partially Observable MDP

In this subsection, we introduce domain-specific characteristics in QT and further explain the reason why it is suitable to model the whole QT process as a partially observable Markov Decision Process (POMDP). In the financial market, security prices are formed by orders from bulls (investors with optimistic market outlooks) and bears (investors with pessimistic market outlooks). At a high level, prices are influenced by macroeconomic and microeconomic activities. The unpredictable events and trading behaviors lead to the noisy financial market. Thus We cannot directly observe the actual market states. For instance, no one knows exactly whether a piece of good news leads the price up or whether orders can be executed at expected prices. The only data we can use are historical prices and volumes. In other words, the price and volume is a part of the underlying market state. The technical indicators in technical analysis can be treated as the observations of the prospective price trends. In general, QT is exactly a sequential decision-making problem about what and when to trade. The POMDP is a realistic generalization of a Markov Decision Process (MDP) for modeling the QT problem. In general, an MDP is a 5-tuple $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$. Specifically, $\mathcal{S}$ is a finite set of states. $\mathcal{A}$ is a finite set of action set. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a state transition function, which consists of a set of conditional transition probabilities between states. $\mathcal{R}$: $\mathcal{S} \times \mathcal{A} \to \mathcal{R}$ is the reward function, where $\mathcal{R}$ is a continuous set of possible rewards. $\mathcal{R}$ indicates the immediate reward from taking an action in a state. And $\gamma \in [0, 1)$ is the discount factor. For the deterministic policy, the goal of an agent is to learn a policy $\mu : \mathcal{S}$

Proximal Policy Optimization

PPO alternates between sampled data through interaction with the environment, and uses random gradient rise optimization to replace the objective function. The standard strategy gradient method performs gradient update for each data sample, while PPO adopts a new objective function,

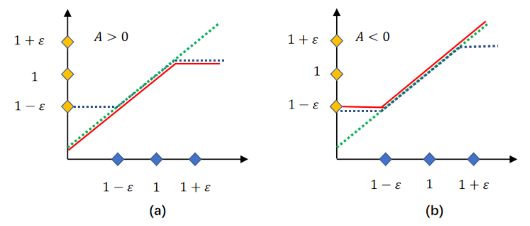

Figure 2: Clip function

which can realize multi cycle and small batch update. PPO has yielded state-of-the-art results in policy search, a sub field of reinforcement learning, See Figure 2, with one of its key points being the use of a surrogate objective function to restrict the step size at each policy update(Zhu and Rosendo 2020). PPO-clip updates policies via:

$$\theta_{k+1} = \arg\max_{\theta} \mathrm{E}_{\mathrm{s},\mathrm{a}\sim\pi_{\theta_{\mathrm{k}}}} \left[\mathrm{L}\left(s, a, \pi\right)\right]) \tag{1}$$

where π is the policy, θ is the policy parameter,k is the k^{th} step, a and s are action and state respectively. It typically takes multiple steps of SGD to optimize the objective L:

$$\min\left(\pi \mathrm{A}^{\pi\theta_{\mathrm{k}}}(\mathrm{s},\mathrm{a}), clip\left(\pi, 1-\epsilon, 1+\epsilon\right) \mathrm{A}^{\pi\theta_{\mathrm{k}}}(\mathrm{s},\mathrm{a})\right) \tag{2}$$

In this way, the clip function avoids excessively large policy updates and reduces the problem of catastrophic steps(Melo and Máximo 2019).

Problem Formulation

In this section, we provide the PPO formulation for the optimal trade execution problem and describe the state, action, reward, and the algorithm used in the experiment.

State:Due to the huge noise of financial data, the effect of using original data is often not good, so we have processed it into a sparse signal matrixWe invited the quantitative analysts on the ground to formulate the most appropriate financial signals for us.

Action:To compare different trading strategies, we stipulate that the agent makes trades with the minimum security amount. The trading action here is defined as a continuous probability vector $a_t = [P_{long}, P_{hold}, P_{short}]$.

Reward:Using the reward function in DRL algorithm (Moody and Saffell 1999) for reference, we select the differential Sharpe ratio (D) as our reward function. Here, the Sharpe ratio (Sr) (Sharpe 1966) is an evaluation for risk-adjusted return. The Sharpe ratio (Sr) indicates the ratio of the excess return (cumulative return minus risk free return) over one unit of total risk. Without the loss of mathematical generality, at time t, Sr_t is defined as:

$$Sr_t = \frac{E\left[\mathbf{R}_{t-n:t}\right]}{\sigma\left[\mathbf{R}_{t-n:t}\right]} \tag{3}$$

Assumption: The most important assumption in our experiment is that the actions that DRL agent takes have only a temporary market impact, and the market is resilient and will bounce back to the equilibrium level at the next time step. This assumption also suggests that the DRL agent's actions do not affect the behaviors of other market participants.

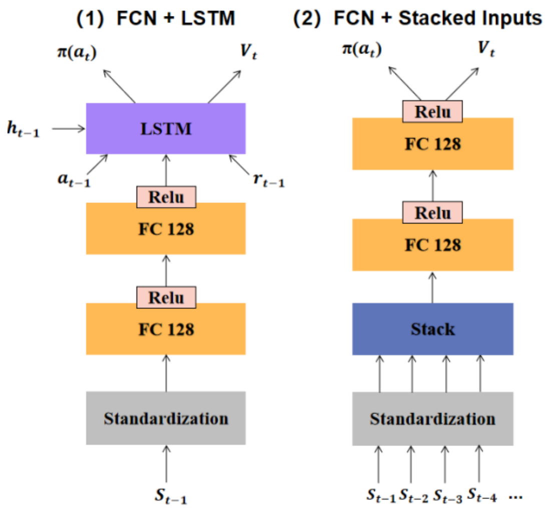

Figure 3: PPO Architectures

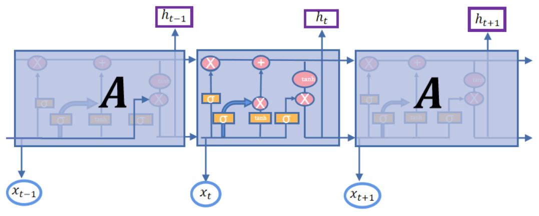

Figure 4: LTSM Modules

The market resilience assumption is the core assumption of this article and also all previous research applying RL for optimal trade execution problems. The reason is that we are training and testing on the historical data and cannot account for the permanent market impact. However, the cryptocurrency we choose in the article are liquid, and the actions are relatively small compared with the market volumes. Therefore, the assumption should be reasonable.

Most of the assumptions are also the core assumptions in previous research because we need a highfidelity market simulation environments or data collected by implementing the DRL algorithm in the real market rather than historical data to account for these factors such as order delays, permanent market impact, and agent interactions, etc.

PPO Architecture

The selected network architectures are illustrated in Figure 3.

Network Architecture: In the PPO algorithm, we have implemented two network architectures: 1) FCN with two hidden layers, each hidden layer has 128 hidden nodes, and each hidden node has ReLU activation function. The input layer has 22 nodes, including private attributes (such as remaining inventory and time consumption) and LOB attributes (such as level 5 bid / ask prices and volumes). After that, we concatenate the model output with previous rewards and actions and feed them to an LSTM network with

Table 1: Model Performances Comparison.

Method	**Return rate**	**Sharp ratio**	**Worst loss**	**MDD ratio**	**MDD Duration**	**Profit/loss ratio**	**Risk ratio**
PPO+LSTM	**7.98%**	7.87%	**-2.42%**	**1.10%**	**2310**	**99.15%**	**7.23**
PPO	-5.84%	7.66%	-15.02%	3.43%	3478	94.90%	-23.23
DDPG	-6.98%	**7.89%**	-22.16%	25.32%	3501	95.57%	-5.01
DQN	-16.05%	6.30%	-23.24%	24.25%	3491	95.64%	-0.66

a cell size of 128. The LSTM outputs the policy π_t and state-value function π_t 2) FCN with the same settings as 1), except that we stack the nearest LOB attributes as model input. We choose Adam optimizer for weights optimization(Lin and Beling 2020).

Long Short-Term Memory: Long short term memory (LSTM) is a special RNN, which is mainly used to solve the problems of gradient disappearance and gradient explosion in the process of long sequence training. LSTM networks are composed of a number of cells, and an input gate, an output gate, and a forget gate within each cell regulate the flflow of information into and out of the cell, as demonstrated in Figure 4. In short, LSTM can perform better in longer sequences than ordinary RNN.

Experiments

In this section, we present the proposed framework's performance. To verify the effectiveness of the proposed framework, we compare it with the PPO, DDPG, DQN model as well as several DRL models. Our proposed framework converges fast and has significantly outperformed the baseline models on cryptocurrency during the backtesting. In our experiment, we apply DeepMind's framework to assess the stability in the training phase and the performance evaluation in the backtesting(Mnih et al. 2015).

Experimental Setup

Datasets: We're using the data of the most important cryptocurrency, which is Bitcoin. We selected five minutes of high-frequency trading data that lasted from 2018/5/1 to 2018/9/1. We also did some processing on these data.

Algorithms: We use several common models as a contrast. To present these methods,we list them as follow.

- **PPO**: The PPO algorithm uses the 32 states and shaped rewards defined in(Lin and Beling 2019).
- **DDPG**:The DDPG algorithm integrates deep learning neural network with Deterministic Policy Gradient, proposed by (Lillicrap et al. 2015)
- **DQN**:The modified DQN algorithm proposed by(Lin and Beling 2019).

Trainning and Stability: Assessing the stability and the model performance in the training phase is straightforward in supervised learning by evaluating the training and testing samples. However, it is challenging to evaluate and track the RL agent's progress during training,since we usually use the average episode rewards gained by the agent over multiple episodes as the evaluation metric to track the agent's learning progress. The average episode reward is usually noisy since the updates on the parameters of the policy can seriously change the distribution of states thatthe DRL agent visits.

In Figure 6,we observe that the framework converge fast in 100 epochs.The reward function can reflect the degree of convergence. Although it is still fluctuate before 200 epochs, it tend to converge and smooth finnally. It indicates that the framework has good convergence effect in training.

Main Evaluation and Backtesting: We will use five-minutes data to conduct experiments. Five minutes bar reflects fluctuations within 5 minutes. For RL, it is difficult to keep action continuity on such highfrequency data. But in the real financial market, minutefrequent data are quite common. For better simulation, we take into accounts practical constraints, including the transaction fee and the constant slippage. Furthermore, we assume that each order can be traded in the opening time of five minutes bar and the reward is calculated in the closing time. We initialize our account with \$ 500,000 in cash at the beginning of the test. The most widely used criteria of the interest in QT are used to evaluate the policy performance:

- **Total return rate** Tr:=$(P_{end} - P_{start})/P_{start}(P$ is the total value of the position and cash).
- **Sharpe ratio** (Sharpe 1966) Sr:=$[\mathbf{r}]/\sigma[\mathbf{r}]$ considers benefits and risks synthetically and reflects the excess return over unit systematic risk.
- **Maxium Drawdown ratio** (Magdon-Ismail and Atiya 2004) Mdd:= $\max(P_i - P_j)/P_i,\ j > i$ measures the largest decline in history and shows the worst possible scenario.
- **Worst loss** Worst loss= P_{worst}/P_{start} , Maxium loss over investment time.
- **Maxium Drawdown Duration** Maxium Drawdown duration describes the time it takes for holding value to reach a new high from the start of retracement.

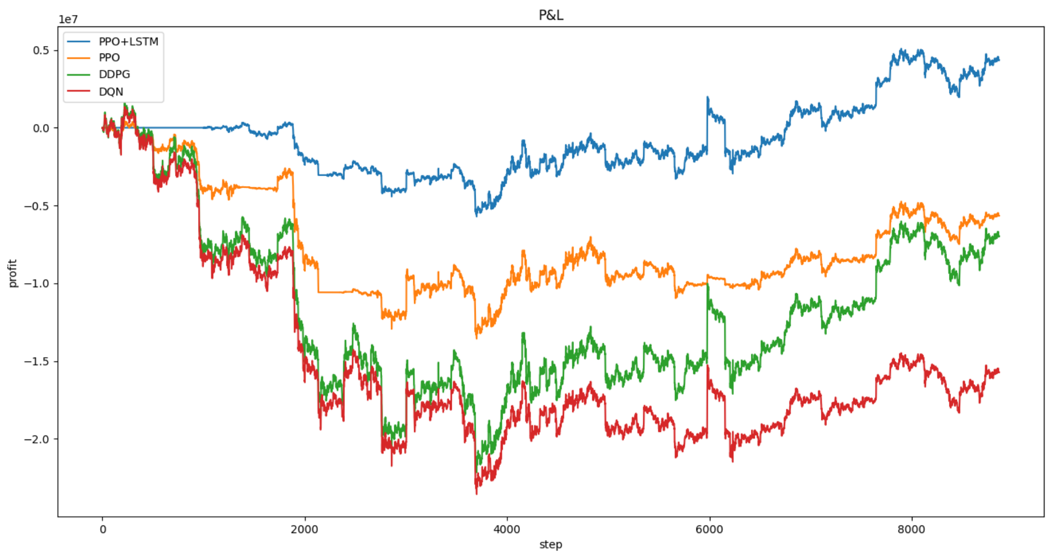

Figure 5: The comparison of Models' profit.The PPO used LSTM is better than other methods obviously.

- **Profit/loss ratio** x_p/x_l ,$x_p$is the average of the gains over time, and Xl is the average of the losses over the same period , The profit/loss ratio is the average profit on winning trades divided by the average loss on losing trades over a specified time period.
- **Risk ratio**:Pend/Maxium Drawdown ratio,compares the potential profit of a trade to its potential loss.

Figure 5 shown that the PPO used LSTM has better performance than other methods. Not only can it achieve prominent profit, with steps growing, but also can keep stable and tend to increase.Except that, our framework also performe well in the others hand. In the table 1,we specific conduct experiment in detail, and compare various indicators to present our framework's performance. The tabel shown that our framework get first-class grade except Sharp ratio. For all that, the DDPG model just better than our method 0.02%.

Conclusion

In this article, we propose an end-to-end optimal trade execution framework based on PPO. In our experiments, we demonstrated that our trading framework performed better than PPO, DQN and DDPG in almost all metrics. Additionally, we have also demonstrated that DRL agents are able to learn good execution strategies even with the sparse rewards. In addition, our data processing is also effective, and it has been proved that priori knowledge of professionals is needed in many fields.

In the future, we are planning to relax the assumption that the DRL agent's actions are independent of other market participants' actions and model the interactions of multiple DRL agents and their collective decisions in the market.

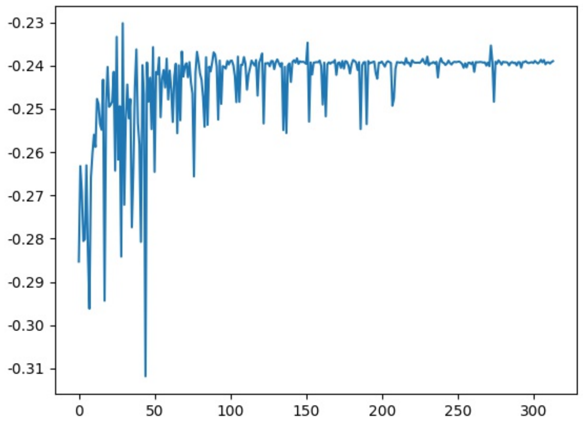

Figure 6: The reward function of PPO used LSTM,reflecting the fluctuation of the framework's training process.The model converge fast in 100 epochs.

References

Guan, Y.; Ren, Y.; Li, S. E.; Sun, Q.; Luo, L.; and Li, K. 2020. Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization. *IEEE Transactions on Vehicular Technology*, 69(11): 12597–12608.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *Computer ence*.

Lin, S.; and Beling, A. 2019. Optimal Liquidation with Deep

Reinforcement Learning. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019) Deep Reinforcement Learning Workshop, Vancouver, Canada*.

Lin, S.; and Beling, P. A. 2020. An End-to-End Optimal Trade Execution Framework based on Proximal Policy Optimization. In *IJCAI*, 4548–4554.

Magdon-Ismail, M.; and Atiya, A. F. 2004. Maximum drawdown. *Risk Magazine*, 17(10): 99–102.

Mallqui, D. C.; and Fernandes, R. A. 2019. Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. *Applied Soft Computing*, 75: 596–606.

McNally, S.; Roche, J.; and Caton, S. 2018. Predicting the price of bitcoin using machine learning. In *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)*, 339–343. IEEE.

Melo, L. C.; and Máximo, M. R. O. A. 2019. Learning humanoid robot running skills through proximal policy optimization. In *2019 Latin american robotics symposium (LARS), 2019 Brazilian symposium on robotics (SBR) and 2019 workshop on robotics in education (WRE)*, 37–42. IEEE.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Munim, Z. H.; Shakil, M. H.; and Alon, I. 2019. Next-day bitcoin price forecast. *Journal of Risk and Financial Management*, 12(2): 103.

Nevmyvaka, Y.; Feng, Y.; and Kearns, M. 2006. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd international conference on Machine learning*, 673–680.

Ning, B.; Lin, F. H. T.; and Jaimungal, S. 2018. Double deep q-learning for optimal execution. *arXiv preprint arXiv:1812.06600*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhu, W.; and Rosendo, A. 2020. Proximal Policy Optimization Smoothed Algorithm. *arXiv preprint arXiv:2012.02439*.