Proceedings of the 7th International Conference on Economic Management and Green Development
DOI: 10.54254/2754-1169/39/20231973

The Application of Deep Reinforcement Learning in Stock Trading Models

Jinshimo Aken[1], Dengke Liang[2], Zifei Lin[3] and Chuyi Wang[4,a,*]

[1]University College London, London WC1E 6BT, The United Kingdom
[2]School of Business, Shandong University, 264200,China
[3]International Business College,South China Normal University,528225,China
[4]School of Data Science, Capital University of Economics and Business, 100070, China
a. chuyiwang@arizona.edu
**corresponding author*

Abstract: Deep Reinforcement Learning (DRL), which integrates the perceptual strength of Deep Learning (DL) with the determination strength of Reinforcement Learning (RL), has emerged as an advanced approach in stock trading. This article focuses on summarizing the research on DRL in stock trading over the past five years, with an emphasis on state definition, action design, reward design, and algorithm selection in stock trading models. Many studies have found that the use of DRL in stock trading can effectively improve investment returns and profitability. In the last years, the adoption of DRL in the stock market has increased and researchers have achieved higher returns and substantial profits through continuous model optimization. However, current research on DRL models faces challenges because of the need for large amounts of complex and uncertain stock market data and the impacts of market volatility and the influence of information asymmetry. This review compares the discrepancies in processing logic among various studies and summarizes the progress made in existing research. It also explores the current challenges and limitations, and discusses potential improvement directions for DRL models in stock trading.

Keywords: deep reinforcement learning, stocks, trading models

1. Introduction

Stock trading is an essential part of the financial sector. Accurate trading of future stock prices can inform investment decisions and lead to profitable results. Over the past few years, researchers have studied machine learning (ML), DL, RL, and DRL-based models to build precise stock price trading models. In their study, Yang et al. [1] highlight how DRL can enhance stock trading by integrating the benefits of both DL and RL. DRL can handle large amounts of data and can capture the nonlinear relationships between variables that are often necessary for accurate stock trading. In addition, DRL algorithms are able to learn and adapt to changing market conditions, enabling better trades to be made even in highly volatile markets. However, the uncertainty and complexity of the equity markets make the accuracy and stability of DRL trading models challenging. Despite these challenges, the potential benefits of DRL for equity trading make it a compelling area of research and drive researchers to develop more accurate equity trading models through continuous optimization, which can inform investment decisions and lead to better financial outcomes.

DOI: 10.54254/2754-1169/39/20231973

With the wide application of ML in the stock market, a large number of scholars have begun to focus on optimizing various processes of DRL trading algorithmic models to improve the performance of stock trading. The processing logic of the DRL model in the stock trading model consists of setting the state, choosing the action and designing the reward, selecting and executing actions according to the strategy, and optimizing the model parameters through the back propagation algorithm to maximize the profit. In terms of defining the state, scholars such as Gosh et al. [2] conducted research on how to apply a better model to perceive and extract market characteristics to fit the state. At the reward function level, Yang et al. [1] define it according to different task objectives (including risk factors, etc.) Judging the pros and cons of decision-making after environment interaction. At the level of action design, researchers suggest expanding the action space beyond just buy, sell, and hold, in order to accommodate the current situation. In the choice of algorithm, the continuous evolution and integration of various RL algorithms improves the decision-making efficiency of the stock trading system model from different dimensions.

This paper analyzes the research literature on stock trading, compares the differences in processing logic of deep reinforcement models in different studies in terms of state information acquisition, reward functions, actions, and algorithm selection, and discusses the application of DRL models in stock trading. Improvement directions. This paper further summarizes the challenges, difficulties, and shortcomings of research in this area, including how to select features, how to design appropriate risk control strategies, how to provide more accurate reward functions, and so on. The purpose of this essay is to present a general reference for future research on DRL stock trading and to promote the development and progress of the field. At the same time, it provides stock investors and investment institutions with rigorous DRL knowledge to support their more accurate assessment of stock trading markets as well as decision support.

2. DRL in Stock Trading Model Processing

The interaction procedure between the trading intelligence and the financial environment in the DRL-based stock trading model is the following parts, as shown in Figure 1: State: represents the state of the environment in which the trading intelligence is located at a specific time, and it is a vector containing all information about the stock environment. Action: represents the action taken by the trading intelligence to achieve a specific goal, it is a choice among a set of feasible actions. Reward: It is a scalar value that represents the feedback from the environment to the behavior of the intelligence. Policy: It is a mapping function from states to actions, where the policy is to optimize long-term payoffs. Value function: It is a function that enters a state and outputs a value that represents the expected reward in that state. In DRL, an intelligence learns policies and value functions algorithmically to improve its decision making ability in complex environments [1]. Therefore, this paper focuses on four aspects: state definition, action design, reward design and algorithm selection to illustrate the development of research on the application of DRL in stock trading models in recent years, compare the merits and demerits of different studies on the processing logic of stock trading models, and thus draw relevant conclusions for the subsequent development of DRL in the field of finance.

DOI: 10.54254/2754-1169/39/20231973

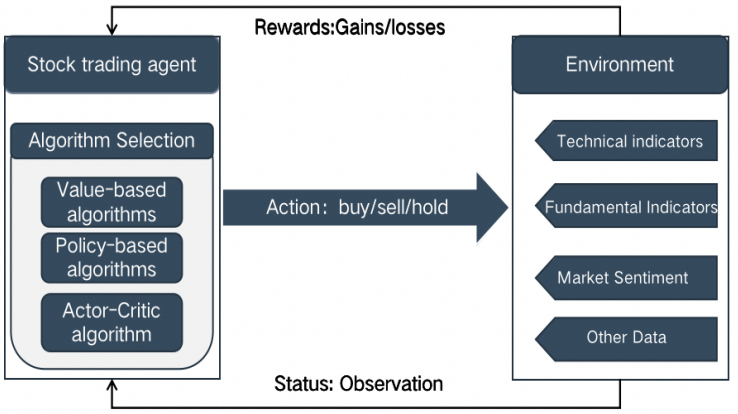

Figure 1: DRL in stock trading model processing [1].

3. DRL-based Stock Trading Models

3.1. Feature Extraction and State Definition

Most scholars point out that the design of the state of the market is complicated due to the fact that there are various elements that affect the trading of stocks, so the state settings may affect the effectiveness of the model and the experimental results. This section explores the differences in the definition of stock states and the extraction of features in different literature.

One type of research proposes that DRL models can automatically extract features from data that form part of the state and pass them to the intelligence. For example, Shi et al. [3] suggested that there may be a multicollinearity problem when computing many technical indicators, so in their study, feature selection was simplified based on the efficient market hypothesis by not explicitly extracting technical indicators other than the logarithmic growth rate of the inputs, but by designing DL networks thus capturing potential dependencies between stock information data. The results show that a DL architecture based on a policy network with CNN layers can effectively mine and extract potential dependencies in stock data. This approach largely reduces the burden of human intervention and better captures the information in the data. However, as the model can easily remember the noise and details in the data, it can lead to poor performance of the model on the test set, which can lead to overfitting problems.

Another type of research points out that indicators such as technical indicators, fundamental indicators, and investor sentiment can provide richer and interpretable characteristics, as well as more economically meaningful features. For example, in stock trading models, some studies consider the use of technical indicators of stocks as characteristic indicators to construct market states. For example, Cao et al. [4] scholars used seven financial indicators for each day of a single stock: the variance average, Bollinger Bands, relative strength indicator, overbought and oversold indicator, average trend index, 30-day moving average price, and 60-day moving average price, respectively, as technical indicators to construct the market state. Some scholars such as Ma et al. [5] used two paralleled modules to extract features: One module uses a fully connected layer to understand the present situation of the stock, and the other uses a long short-term memory (LSTM) layer designed to discover past trends. In the end, numerous experiments in the Chinese stock market have suggested that this algorithm yields higher returns than several existing algorithms. In addition, recent studies in behavioral finance have shown that investor sentiment can interfere with investors' decisions and thus lead to stock market volatility. Some studies such as Zuo [6] proposed: using a sentiment analysis

DOI: 10.54254/2754-1169/39/20231973

model based on news and commentary texts to calculate an investor sentiment influence score, inputting this sentiment influence score and then retraining the intelligences. Finally, A Dual Delay Deep Deterministic Policy Gradient Algorithm Incorporating Sentiment Analysis (CTD3-LSTM) model achieves a maximum return of 38.81%, an omega rate of 1.35, and a maximum retracement rate of 14.3% on the test dataset, all of which are significantly better than the other comparisons. In conclusion, this approach as a more mainstream research direction can avoid the shortcomings of single indicators, while increasing the diversity of features, which can reflect the market more comprehensively. However, the relationship between different feature indicators is complex, and how to select and combine features may affect the effectiveness of the model.

3.2. Design of Movement

DRL has been a popular research field in the application of stock trading. Different literature sources present varying definitions and implementation approaches regarding the trading actions.

Regarding buying and selling actions, earlier studies, such as Cao et al. [4] defined the action space to include buy, sell, and hold actions for individual stocks or a range of action values [-1, 1] for multiple stocks. In this context, values greater than 0 indicate buy actions, values less than 0 indicate sell actions, and values equal to 0 indicate no trading action.

As research progressed, some scholars introduced the concept of thresholds based on the buy and sell signals. When the difference in probabilities between buy and sell signals is below a predetermined threshold, the strategy sets the action as a predefined one. If the difference exceeds the threshold, the strategy greedily selects the action, as proposed by Jeonga and Kim [7]. This approach maintains a certain level of exploration within the threshold, controlling the risk level of the strategy and avoiding over-reliance on greedy strategies. This helps in identifying more potential trading opportunities and prevents over fitting to historical data. However, this approach may lead to suboptimal strategy performance in high-volatility markets. Additionally, setting the threshold accurately poses a significant challenge, requiring precise market predictions; otherwise, the strategy's performance can become unstable.

These methods solely consider the trading behavior of individual stocks and overlook the correlation among stocks in a portfolio. In terms of portfolio management, for trading models involving multiple stocks, some scholars have started defining the action space of trading strategies as the weights of the portfolio, as described by Jiang and Liang [8]. They adjust the actions on the weights of every asset in the portfolio to maximize the portfolio's returns. Furthermore, Li and Qin [9] highlight the criticality of risk management in portfolio management, including constraints on individual stock weights, setting target risk levels, and investment horizons. This research provides a new solution in the field of financial portfolio management, attracting significant attention from the industry.

In conclusion, studies indicate that DRL outperforms traditional ML methods in terms of returns in stock trading. However, due to the complexity and randomness of the market, specific action design requires more precise and accurate algorithms. Further research is needed to explore appropriate risk control strategies in depth.

3.3. Designing Rewards

In the process of applying DRL to stock trading, the reward function designed according to different trading strategies can measure the quality of the agent's operation, so that the agent can obtain the maximum reward value and optimize the strategy to maximize the assets held. The benchmarks for agents to obtain rewards mainly include: return on investment, value function, profit, risk, etc. The

DOI: 10.54254/2754-1169/39/20231973

following will describe the optimization of the reward function by researchers from the two perspectives of the representation of the reward function and the adjustment of the item.

In terms of the choices of the reward function, researchers have also made different attempts. In the trading market, the portfolio behavior is mostly continuous. Scholars such as Silver et al. [10] used the critic-actor deterministic strategy gradient, took the Q function as the reward function and another neural function as the action function to output continuous action. However, based on the difficulty and instability of training two neural networks, Jiang and Liang [8] combined a simple deterministic policy gradient in the portfolio, used a direct reward function to improve model efficiency, and presented the reward value with logarithms, simplifying the calculation. In addition, a single reward function is often unable to evaluate the performance with different standards. Shi et al. [3] proposed two reward functions simultaneously in a double DQN model (two DQN models were designed to solve the overvaluation of actions), showing that the trained model with the accrual basis reward function (the operation of the stock will cause the stock return to be easily affected by the stock price change of the next trading day) has higher returns, and the one with the cash basis reward function (only considering the real-time stock returns on the trading day) has higher accuracy. The agent's risk-adjusted performance is also frequently evaluated in the reward functions. The Sharpe ratio considers both returns and risks, and is adopted by a large number of scholars. While Mohan et al. [11] adopted the Sortino ratio, which only considers downside risks and is more applicable where financial market downsides are common.

With regard to terms adjustment, researchers make trade-offs based on different considerations. In order to ensure that the transaction is economically feasible, the impact of stock transaction costs (such as handling fees) should be taken into account, Jiao [12], Liang and Jiang [8] pointed out that transaction costs should be deducted to varying degrees in the reward function to improve the training speed of the model. In addition, market volatility is an important factor affecting stock prices. Many scholars who hold the same view as Yang et al. [1] introduce market volatility into the reward function to make the agent better focus on and deal with risks. Researchers such as Zhang et al. [13] improved the reward function with a volatility term to expand the trading position when the volatility is low. At the same time, the fluctuation target is limited to reduce the impact of market fluctuations on rewards. Different from predecessors who set the reward of the "holding" action to 0, Yang et al. [14] included "holding" in the reward as the market fluctuates. The data analysis findings from examining 10 actual stocks demonstrate that the recommended model offers greater gains while lowering the level of risk for investors.

The reward value setting directly affects the overall convergence and stability of the model. Usually, in the field where the DRL network model is used better, It focuses on using current rewards to optimize neural networks. However, the following aspects of the reward function still need to be optimized: a single reward function based on different strategy choices, the function is targeted but limited, so multiple reward functions will be applied at the same time in many studies, but this also brings complexity to the model problem of reduction in computing speed. At the same time, the combination of portfolio theory such as hedging strategies and the reward function is rarely used; researchers face the conflict of simplifying and improving the operation speed and over-optimizing to reduce the efficiency of the model in the choice of reward value.

3.4. Algorithm Selection

The objective of enhancing the DRL algorithm is to optimize the aggregate profis achieved by the agent. In applying DRL to stock trading, the agent's objective during the learning process is to discover the optimal investment strategy to achieve the highest total profit. Using DRL algorithms, an investor can leverage extensive data to consistently refine their investment decisions and increase their potential returns. According to Zhou [15], researchers mainly focus on three types of algorithms,

DOI: 10.54254/2754-1169/39/20231973

as shown in Figure 2. Each with its own advantages and disadvantages that may have better performance in different scenarios. The selection of specific algorithms should be based on the application scenario and problem requirements.

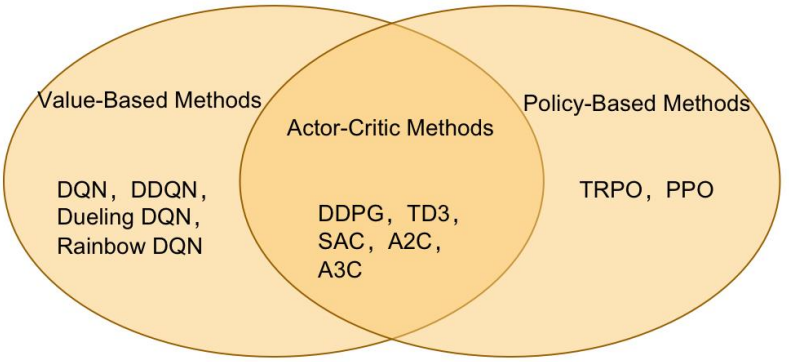

Figure 2: The relationship between different DRL algorithms [15].

3.4.1. Value-based DRL Algorithm

The core idea of Value-based algorithms is to evaluate the effect of taking a certain action using a value function and then carry out policy optimization based on this to obtain an optimal policy. Using the basic algorithm Q-learning, the researchers designed the Deep Q-Network (DQN) algorithm, which uses deep neural networks to learn the Q-value function to handle high-dimensional state and action spaces. DQN adopts techniques including experience replay and fixed target network to enhance the stability of Q-value fitting and effectively solve the problems of sample correlation and unstable target function. DQN algorithm has been widely used in stock trading, such as Jeonga and Kim [7] who used the DQN algorithm to update Q-values through neural networks to analyze which strategies can be used to improve profits. DDQN is an improved algorithm to counter the overestimation problem in DQN. This approach utilizes two separate Q-networks that work independently to improve the ability of agents. Researchers such as Shi et al. [3] demonstrated through experiments that DDQN has less overvaluation and better stability, it has achieved higher profits in stock trading. Dueling Q-Networks is a novel deep Q-learning algorithm that splits the Q-value function using different network structures to learn the value of states and observable policies to improve learning efficiency. However, novel algorithms do not necessarily surpass the original ones, the effectiveness of the same algorithm varies greatly in different fields. Li et al. [16] compared the performance of DQN, DDQN, and Dueling Q-Networks in the same trading data, showing that the best-performing DRL model was DQN rather than the improved algorithm based on it. In addition, several researchers proposed the idea of using multiple neural networks in the agent. For example, Ma et al. [5] proposed a new parallel multi-module DRL (PMMRL) algorithm to optimize the DQN model to consider the long-term historical trends and related information of the market. This algorithm performs well in experimental environments.

3.4.2. Policy-based DRL Algorithm

Policy-Based algorithms typically use neural networks as function approximators to parameterize the policy function, in order to obtain the optimal decision and the highest cumulative reward, the parameters of the neural network can be adjusted using optimization algorithms or supervised learning techniques. To improve the efficiency of the algorithm in data collection and training, a reasonable update step size needs to be determined when updating the policy. To this end, researchers

DOI: 10.54254/2754-1169/39/20231973

have designed the Trust Region Policy Optimization (TRPO) algorithm. This algorithm has better stability. Proximal Policy Optimization (PPO) is an improvement on TRPO, which solves some problems with TRPO by modifying the objective function, improving the efficiency and stability of the algorithm, and is currently a commonly used algorithm for stock trading through DRL. Researchers such as Shreyas et al. [17] evaluated the performance of TRPO and PPO in maximizing profits for stock trading. Their research showed that, compared to PPO, TRPO provided better trading strategies.

3.4.3. Actor-critic Algorithm

Actor-Critic algorithm is an important mechanism in DRL. Its core idea is to constantly update strategy and value estimation by combining the use of strategy function and value function, and finally get the optimal strategy. Among them, the Actor function adopts a Policy Based algorithm, which uses the neural network fitting the policy function to select an appropriate action in the continuous action space, while the Critic function adopts a Value Based algorithm and uses the neural network fitting the state value function to carry out single or multi-step update. The following will introduce three algorithms that perform well in stock market applications: Deep Deterministic Policy Gradient (DDPG) algorithm performs well in the environment of learning continuous action space, and it can use deep neural network to deal with high-dimensional state and action space information. Luo et al. [18] searched for the optimal trading strategy in stock market trading through DDPG algorithm and obtained considerable annual returns. Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm makes some improvements to the DDPG algorithm, including using two Critic networks and one Actor network, and using delayed update and target strategy noise correction to improve learning speed and stability. Soft Actor-Critic (SAC) algorithm combines the advantages of strategy gradient method and entropy regularization to improve learning efficiency and stability. Nguyen et al. [19] studied the performance of DDPG, TD3 and SAC in the US stock environment. There is no difference in the trading behavior of these three algorithms, and all can generate considerable profits.

On the other hand, further optimized Actor-Critic algorithm has also been widely applied, among which, Advantage Actor-critic algorithm (A2C) uses advantage function to accelerate the learning process. It can take full advantage of multiple worker processes for parallel computing. In addition, Asynchronous Advantage Actor-critic algorithm (A3C) was improved based on A2C, mainly optimizing Critic evaluation point, parallel training framework, and network structure. Kang et al. [20] used A3C algorithm to solve stock selection and portfolio management problems, and verified the advantages and disadvantages of A3C algorithm in shortening time and accelerating convergence during training, and pointed out that the model is more effective in training environment than in test environment.

After summarizing and concluding the existing researches, this paper finds that improving the DRL algorithm can effectively improve the processing power of DRL model in stock trading, to achieve better returns. At present, the research mainly focuses on updating and iterating DQN models to improve learning and training efficiency. In the future, researchers can explore more improved algorithms, such as Rainbow DQN, for stock trading. In addition, at present, the research of multi-module DL algorithm in the field of stock trading is not sufficient, but this kind of algorithm has been proved to have excellent performance. Further, researchers can continue to develop DRL algorithms in the field of stock trading.

DOI: 10.54254/2754-1169/39/20231973

4. Conclusion

Over the past decade, DRL has become more widely used in stock trading. Academic researchers constantly improve the processing efficiency of DRL model in stock trading by improving the feature extraction and state definition of the model, designing actions, designing reward functions and algorithm selection, to adapt to the complex stock market. Starting from the above four aspects, this paper summarizes the research on DRL in stock trading in the past five years, and finds that researchers have successfully obtained higher returns and considerable profits in the stock market through continuous optimization of the model. However, the current DRL model still has some limitations. For example, stock market data are very complex and uncertain, which is not conducive to model training. Therefore, the training of the model faces great challenges. At the same time, the model's performance in the test environment is not as good as that in the training environment and other problems need further research.

Future research directions can focus on using multi-module structure to design agents, and improve the prediction and processing ability of DRL model in the real stock market environment, to obtain higher and more stable returns. Furthermore, in stock trading, high market volatility, information asymmetry and other factors will affect the performance of DRL algorithms. Therefore, even more advanced DRL algorithms need to be fully verified and optimized to achieve long-term stable returns. In the actual investment, investors should choose the applicable DRL algorithm to minimize the incompatibility between strategy and stock selection, to reduce the risk of strategy. In summary, this paper studies the application of DRL in stock trading, provides useful references and enlightenment for future research, reveals the shortcomings and limitations of current research, and presents challenges and opportunities for further exploration and development.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] *Yang, H., Liu, X. Y., Zhong, S., Walid, A.: Deep reinforcement learning for automated stock trading: An ensemble strategy. In: Proceedings of the first ACM international conference on AI in finance, 1–8 (2020).*

[2] *Ghosh, P., Neufeld, A., Sahoo, J. K.: Forecasting directional movements of stock prices for intraday trading using LSTM and random forests.Finance Research Letters 46, 102280 (2022).*

[3] *Shi, Y., Li, W., Zhu, L., Guo, K., Cambria, E.: Stock trading rule discovery with double deep Q-network. Applied Soft Computing, 107, 107320 (2021).*

[4] *Cao, D. L., Cui, C. R., Yang, X.: A comparative study of domestic financial market investments based on deep reinforcement learning. Journal of Nanjing University (Natural Sciences), 59(2), 333-342 (2023).*

[5] *Ma, C., Zhang J., Liu, J., Ji L., Gao F.: A parallel multi-module deep reinforcement learning algorithm for stock trading. Neurocomputing, 449:290-302 (2021).*

[6] *Zuo X. D.:Reinforcement learning based on investor sentiment and depth of stock portfolio optimization research. China Management Informationization, 26(3),130-132 (2023).*

[7] *Jeong, G., Kim, H. Y.: Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning. Expert Systems with Applications, (117), 125–138 (2019).*

[8] *Jiang, Z., Liang, J.: Cryptocurrency portfolio management with deep reinforcement learning. In: IEEE 2017 Intelligent Systems Conference (IntelliSys), 905-913 (2017).*

[9] *Li, Y., Qin, Y.: Deep reinforcement learning for portfolio management. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2594-2600 (2017).*

[10] *Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D.: Mastering the game of Go with deep neural networks and tree search. Nature, (529), 484-489 (2016).*

[11] *Mohan, V., Singh, J. G., Ongsakul, W.: Sortino ratio based portfolio optimization considering EVs and renewable energy in microgrid power market. IEEE Transactions on Sustainable Energy, 8(1), 219-229 (2016).*

DOI: 10.54254/2754-1169/39/20231973

[12] Jiao, Y.: Stock Portfolio Management and Empirical Research Based on Deep Reinforcement Learning. Xian: Northwest University, (2021).

[13] Zhang, Z., Zohren, S., Roberts, S.: Deep reinforcement learning for trading. arXiv preprint arXiv:1911.10107 (2019).

[14] Yang, B., Liang, T., Xiong, J., Zhong, C. Deep reinforcement learning based on transformer and U-Net framework for stock trading. Knowledge-Based Systems, 262, 110211 (2023).

[15] Zhou, B.: Lecture 5 Policy Optimization I, https://github.com/zhoubolei/introRL/blob/master/lecture5.pdf, last accessed 2023/6/8.

[16] Li, Y, M., Ni, P., Chang, V.: An Empirical Research on the Investment Strategy of Stock Market based on Deep Reinforcement Learning model. In: Proceedings of the 4th International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS 2019), 52-58 (2019). International Conference on Complex Information Systems, Heraklion, Crete, Greece (2019).

[17] Lele, S., Gangar, K., Daftary, H., Dharkar, D.: Stock Market Trading Agent Using On-Policy Reinforcement Learning Algorithms (2020).

[18] Luo, S, Y., Lin, X, D., Zheng, Z, X.: A novel CNN-DDPG based AI-trader: Performance and roles in business operations. Transportation Research Part E: Logistics and Transportation Review, 131, 68–79 (2019).

[19] Nguyen, H, T., Luong, N, H.: Applying Deep Reinforcement Learning in Automated Stock Trading. Soft Computing: Biomedical and Related Applications 981, 280–292 (2021).

[20] Kang, Q, M., Zhou, H, Z., Kang, Y, F.: An Asynchronous Advantage Actor-Critic Reinforcement Learning Method for Stock Selection and Portfolio Management. In: ICBDR '18: Proceedings of the 2nd International Conference on Big Data Research, pp. 141–145. Association for Computing Machinery, Weihai, China (2018).