Quantitative Trading of Stocks Based on TD3 Algorithm

Kaitian Yu*
School of Computing Science, South China Normal University, Guangzhou, China, 510631
*Corresponding author: 20192133001@m.scnu.edu.cn

Abstract. Intelligent and efficient stock analysis can help investors and institutions judge stock trends, improve investment returns, and avoid investment risks. The use of deep reinforcement learning methods to process stock data and provide investment recommendations has important research value. This question proposes to use the TD3 algorithm to implement a deep reinforcement learning model, introduce commonly used stock technical indicators, design reward and action functions, and use them to backtest recent stock trading data. Finally, comparing it with the moving average strategy and other deep reinforcement learning models, it was found that in the past decade of historical data, the annualized rate of the moving average strategy was 10% -15%, while the annualized rate of the TD3 algorithm was 23% -25%. This indicates that the TD3 algorithm can help investors or institutions make judgments and effectively improve investment returns.

Keywords: Deep Reinforcement Learning, TD3, Stock.

1. Introduction

With the continuous expansion of the world financial market, the demand for investment is also gradually increasing. During this period, the concept of quantitative trading emerged, which achieved automatic trading by setting a series of strategies and using computer programs. This has greatly helped institutional and individual investors improve investment returns and reduce risks. In recent years, the rapid development of the computer industry has promoted research on algorithms related to artificial intelligence, and many experts and scholars have begun to study how to combine artificial intelligence with quantitative trading. Wang et al. used CNN to classify stocks and identify better performing stocks, and added them to their investment portfolio [1]. However, in different cycles, investors' actions are also crucial. Therefore, some scholars combine deep learning and reinforcement learning methods, using CNN or DRSN for feature extraction, and then using DQN or DDPG algorithms for decision-making, making the action space of the agent continuous [2-7]. Some scholars also use Q-learning and Actor-Critic algorithms for decision-making, and use GAN to generate noise and enhance generalization ability [8-9]. Scholars also use integration strategies to combine multiple deep learning models for simulated trading [10]. These studies aim to enable models to autonomously learn the laws of the stock market and take corresponding actions based on these laws, thereby achieving higher returns at lower risk.

This article combines the current background and designs a stock quantitative trading model based on the TD3 algorithm, using historical data of some stocks in the Chinese stock market as training and testing sets. This provides a new way for quantitative analysis, while also allowing deep reinforcement learning to be applied in more scenarios, expanding the influence of deep reinforcement learning and having important practical application value. The main contributions of this article include implementing an AI trader using the TD3 algorithm, optimizing the reward function and the interaction between the agent and the environment, and introducing technical indicators to expand the state space of the stock trading environment.

2. Design of quantitative trading algorithm for stocks

2.1. Stock trading model

The problem of stock trading is to maximize long-term returns by taking different actions at different time points. This article uses the framework of reinforcement learning to solve the problem of stock trading, and Figure 1 shows the framework of reinforcement learning for stock trading.

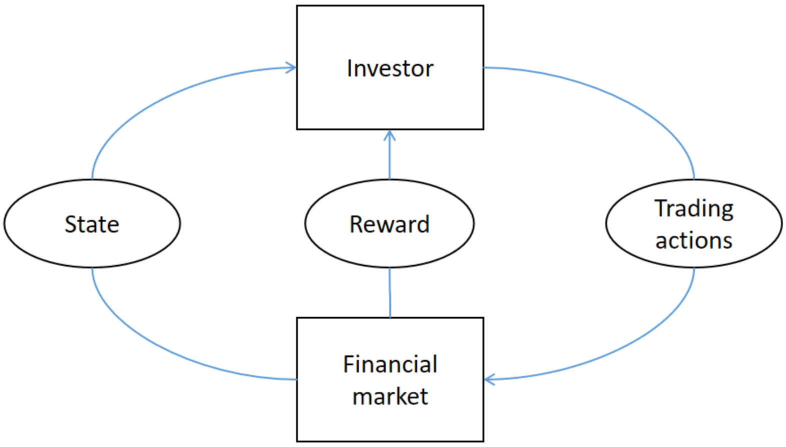

Figure 1. Strengthening the learning framework for stock trading

The process of stock trading: Investors observe the status of that trading day, give actions, and then move on to the next trading day to implement actions. Then, the environment provides feedback to investors, and investors observe the status of the new trading day, thus circulating.In order to describe the problem more rigorously, this article makes the following assumptions about the market and trading rules:

(1) The transaction price is between the lowest and highest prices of each day.
(2) A transaction fee of 3 ‰ is required.
(3) The minimum unit of stock traded is 1.

2.2. State space design

In deep reinforcement learning, agents take action through state information, and in stock trading problems, investors take action through stock information. The financial market environment is complex and noisy, so it is necessary to design a reasonable state space to help investors make judgments. Figure 2 shows the state space of this problem, which is composed of technical indicators, fundamental information, and investor information. Technical indicators include MACD and MA, investor information includes cash holdings and positions, and fundamental information includes opening and closing prices. Among them, technical indicators represent the trend of stocks over a certain period of time, which can play a role in preserving medium - and long-term time series information.

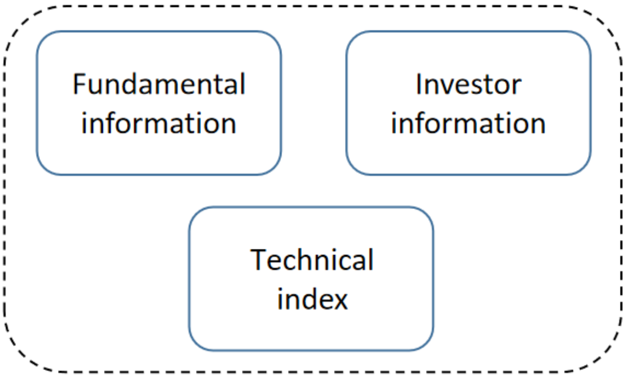

Figure 2. State space

2.3. Action space design

In the stock trading environment, this article chooses to use neural networks to make actions, which are a 1-dimensional vector with a length of 2: [action, amount], and its value range is also between [0,1]. The first element action represents the type of action taken, and the second element amount represents the number of stocks traded. In addition, when setting actions to interact with the environment, the operation of amount within the (0,0.1) range is prohibited.

2.4. Reward function design

The reward function is used to guide Agent learning, so the design of the reward function is crucial. On the one hand, investors hope for high investment returns and on the other hand, they hope for low risks. Therefore, this article designs the following reward function to balance returns and risks:

(1) When selling, if the current stock price is higher than 1.1 times the cost price, i.e. the increase is greater than 10%, a reward r will be given:

$$r = \frac{P}{net\_worth} \tag{1}$$

Where P is the specific amount of profit from this sale, net_ Worth represents the current total assets.

$$P = (c-b) * \alpha * \lambda \tag{2}$$

Where c represents the current price, b represents the cost price, and a represents the quantity sold, λ Represents the deduction of handling fees.

(2) When buying, if the current stock price is less than 0.7 times the cost price, i.e. a decrease of more than 30%, a reward r will be given:

$$r = \frac{new - basis}{basis} \tag{3}$$

Among them, 'new' refers to the cost price of the new stock after this purchase, while 'basis' refers to the original cost price of the stock.

(3) After completing a training session for a stock, two rewards r1 and r2 are given using total pullback and total profit:

$$r_1 = \frac{N - M}{I} \tag{4}$$

$$r_2 = \frac{N - I}{I} \tag{5}$$

M refers to the maximum amount of funds reached during this backtesting, N is the total amount of funds at the end of the backtesting, and I is the principal. Use r1+r2 as a reward for the end of the backtesting.

Although it is difficult for intelligent agents to minimize risks in the end, designing a reasonable reward function can effectively constrain risks.

2.5. TD3 algorithm network design

This article uses the TD3 algorithm for quantitative trading of stocks. TD3 uses an actor network to output an action, and two critical networks to evaluate the action. TD3 has made three improvements on the basis of DDPG: 1. Two Critical networks have been introduced to solve the overestimation phenomenon. 2. Delayed updates improve algorithm stability. 3. Add noise to the action during training to smooth the regularization target strategy. Figure 3 shows the framework of the TD3 algorithm. Due to the introduction of two Critical networks in TD3, the algorithm includes a total of six neural networks, while DDPG consists of four neural networks.

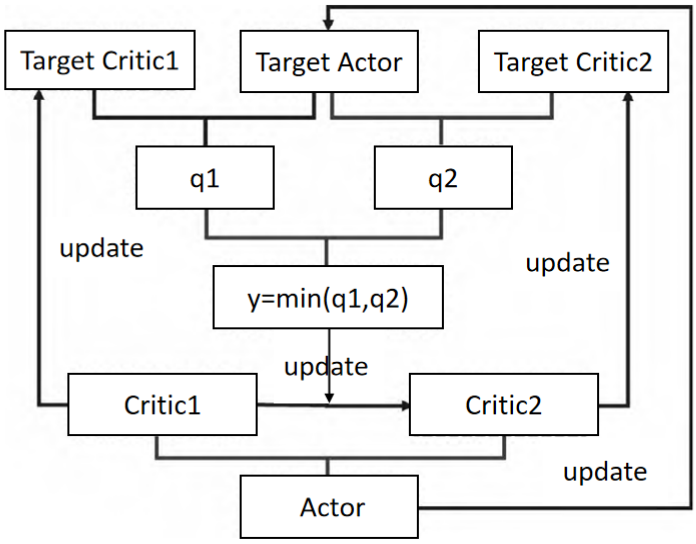

Figure 3. TD3 algorithm framework

The basic process of TD3 algorithm: Firstly, use an Actor network to select an action, and then input the action into the environment to obtain the next state and immediate reward. Then, store the next state and reward as experience in the experience pool. Next, randomly select a batch of experiences from the experience pool and use Target Criticc1 and Target Criticc2 to calculate the Q value for the next state. Then update Critic1 and Critic2. Finally, update the Actor. During the training process, perform soft updates on Target Actor, Target Criticc1, and Target Critic2 after a certain number of iterations.

Among them, the size of the input layer of the Actor network is the size of the state space. The input layer and the hidden layer use ReLU as the activation function to increase the expression ability of the neural network. The ReLU function is defined as:

$$f(x) = \max(0, x) \tag{6}$$

The output layer uses the tahh function to map the action interval to [0,1] to comply with the setting of stock trading actions. The tanh function first maps the action to [-1,1], then adds 1 to the action vector [action, amount] and divides it by 2, and then maps the action vector to [0,1].

The role of Critical is to evaluate the actions taken by the Actor, so not only the input state but also the actions taken by the Actor are required. Therefore, the input layer size of the Critical network is the length of the stock environment state plus the length of the action vector. Figure 4 shows the structures of the Critical network and the Actor network:

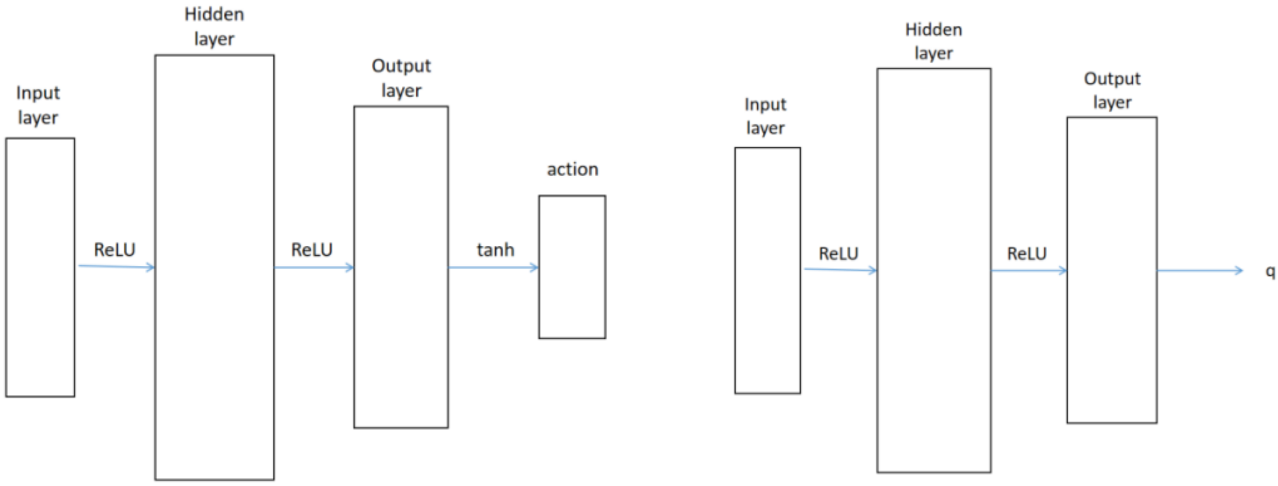

Figure 4. Actor and Critical networks

3. Experiments and results

The framework and environment configuration of this experiment: Python 3.7.4, framework version: PaddlePaddle 2.4.0. The stock data used in the experiment comes from the interface provided by Tushare.

3.1. The experimental data

This article uses the historical data of some stocks in the 300 component stocks of Shanghai and Shenzhen from 2000-01-01 to 2023-01-01 as the training set, and the historical data of the following two groups of stocks from 2013-01-10 to 2023-04-07 as the testing set. Table 1 shows the list of stocks in the testing set, while the training set has a large number of stocks, which will not be listed one by one here. The stocks in the training set are: 603369,600703,688012,603986,300529,000568, 600519,000858,600809,002821,002304,002371,300759,603501,300760,603259,300782,300661.

Table 1. Test set

Stock code	Group 1 Stock name	Stock code	Group 2 Stock Name
600900	Yangtze Power	600276	Hengrui Pharmaceutical
002415	Hikvision	600763	Tongce Medical
002594	BYD	300122	Zhifei Biology
002475	Luxshare	300253	Weining Health
600438	Tongwei Co., Ltd	000423	Ejiao
002230	IFLYTEK	000661	Changchun High tech
600036	China Merchants Bank	600085	Tongrentang
002352	SF Holdings	300015	Aier Ophthalmology
000651	Gree Electric Appliances	300347	Taige Pharmaceuticals
600436	Pien tze huang	002317	Zhongsheng Pharmaceutical

The first group of stocks was randomly selected from industry leaders, while the second group of stocks was randomly selected from the pharmaceutical sector.

3.2. Model training

Unlike the training and testing methods mentioned in most literature, which use the previous period of data as the test set, and then use the latter period of data as the training set. This article follows the quantitative trading process shown in Figure 5: train a strategy, use historical data for backtesting, and continuously optimize the strategy during the backtesting process to achieve good returns on historical data.

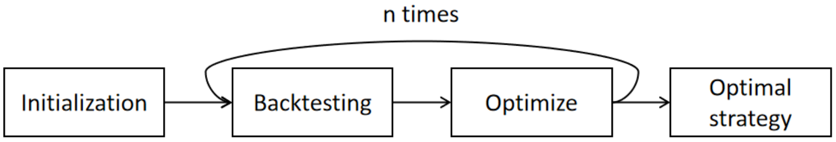

Figure 5. Deep reinforcement learning quantitative trading

In terms of hyper-parameter, the initial learning rate is 0.0001; Set the experience pool capacity to 50000; Batch size is 64; The soft update parameter is 0.005.

3.3. Test results and analysis

In order to verify the effectiveness of the TD3 algorithm, this experiment will use the TD3 algorithm, DDPG algorithm, and moving average strategy to test the data of the test set back and forth, provide a profit change chart, and compare the evaluation indicators of the back testing under each strategy. The first group of stocks is a mixed class, with significant differences in their trends. Using them to test can verify the universality of the model and to some extent simulate its performance in the real stock market. The second group of stocks is of the same type, and their movements are similar, which can play a role in averaging the model effect to a certain extent. Figures 6 and 7 show the comparison of the changes in returns of each strategy under two groups of stocks.

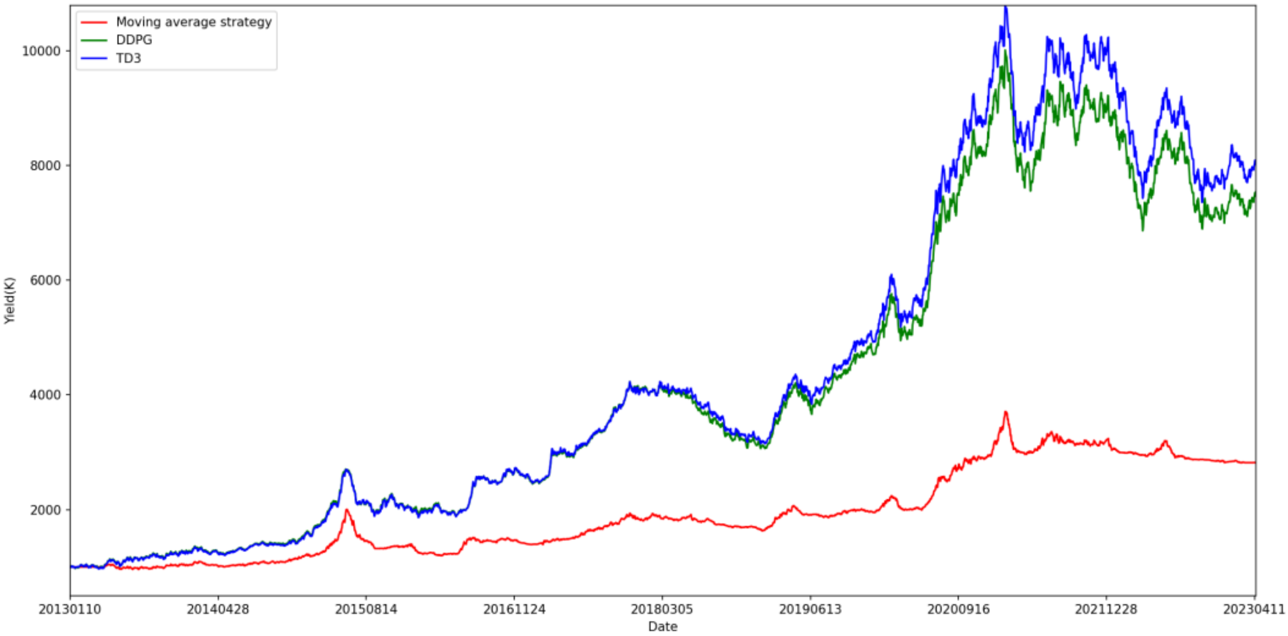

Figure 6. The first group of stock wealth trend chart

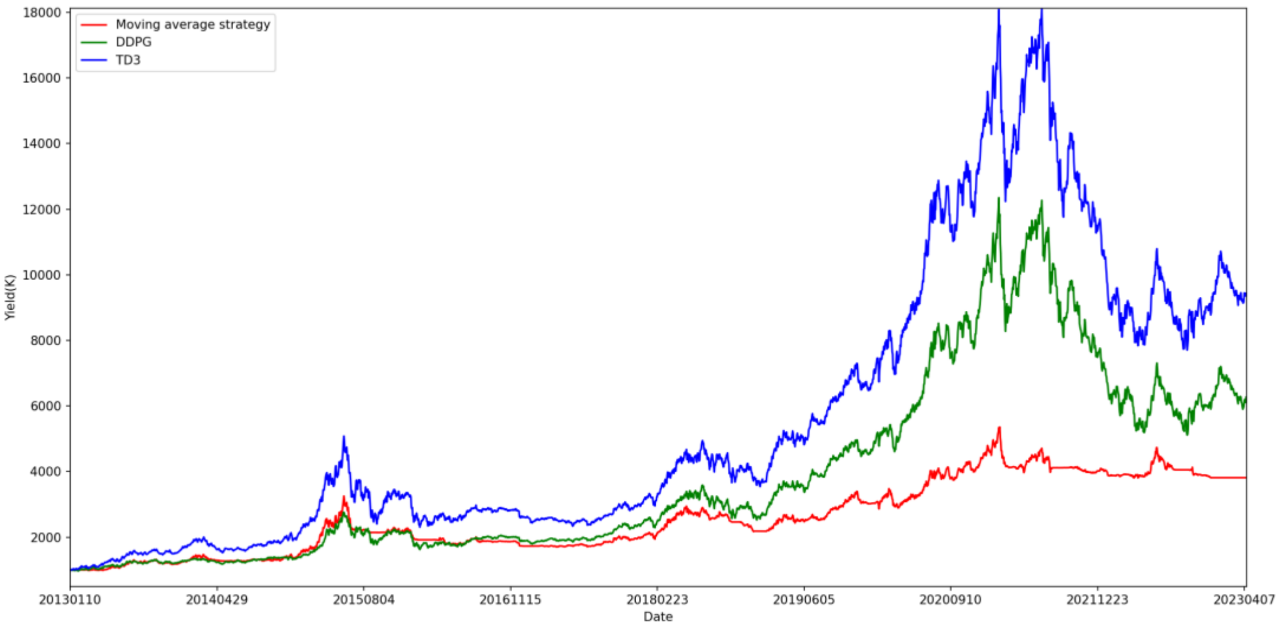

Figure 7. The second group of stock wealth trend chart

The red line represents the moving average strategy, the green line represents DDPG, and the blue line represents TD3. The moving average strategy achieved a cumulative return of 231.7% in the backtesting, while DDPG and TD3 achieved cumulative returns of 589.6% and 774.9%, respectively. It can be seen that the TD3 algorithm performs much better than the moving average strategy and also performs better than the DDPG algorithm. To further compare strategies, Tables 2 and 3 present more indicators to evaluate the effectiveness of quantitative trading.

Table 2. Comparison of evaluation indicators for the first group of strategies

Evaluating indicator	MA	DDPG	TD3
Accumulated Yield	181.9%	654.1%	707.1%
Annualized Yield	10.9%	22.4%	23.22%
Sharpe Ratio	0.5	0.38	0.37

Table 3. Comparison of evaluation indicators for the second group of strategies

Evaluating indicator	MA	DDPG	TD3
Accumulated Yield	281.4%	525.1%	842.7%
Annualized Yield	14.3%	20.1%	25.2%
Sharpe Ratio	0.87	0.44	0.32

From Tables 2 and 3, it can be seen that the annualized rate of TD3 has reached around 24%, surpassing the 21% of DDPG and 13% of the moving average strategy. Although the Sharpe ratio of TD3 is lower than the moving average strategy, it can also bring better returns in some cases, which is related to the reward setting encouraging AI traders to achieve higher returns.Based on the above experimental results, it can be seen that the TD3 strategy is the best investment strategy among these strategies in terms of yield. Moreover, as the backtesting period is 10 years and the TD3 strategy demonstrates high profitability, it can greatly demonstrate that the TD3 strategy has a certain degree of stability and long-term profitability.

4. Conclusions

From Tables 2 and 3, it can be seen that the annualized rate of TD3 has reached around 24%, surpassing the 21% of DDPG and 13% of the moving average strategy. Although the Sharpe ratio of

TD3 is lower than the moving average strategy, it can also bring better returns in some cases, which is related to the reward setting encouraging AI traders to achieve higher returns. Based on the above experimental results, it can be seen that the TD3 strategy is the best investment strategy among these strategies in terms of yield. Moreover, as the backtesting period is 10 years and the TD3 strategy demonstrates high profitability, it can greatly demonstrate that the TD3 strategy has a certain degree of stability and long-term profitability.

In addition, although the strategy proposed in this article has been validated in real historical data, in order to facilitate backtesting, this article has simplified the problem to some extent. If this strategy is to be used in the process of real trading, more real details need to be considered.

References

[1] Wang W , Li W , Zhang N , et al. Portfolio formation with preselection using deep learning from long-term financial data[J]. Expert Systems with Application, 2020, 143(Apr.):113042.1-113042.17.

[2] Sun R , Jiang Z , Su J . A Deep Residual Shrinkage Neural Network-based Deep Reinforcement Learning Strategy in Financial Portfolio Management[C]// 2021 IEEE 6th International Conference on Big Data Analytics (ICBDA). IEEE, 2021.

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

[4] Meiying J. Portfolio Management Based on Deep Reinforcement Learning TD3 Algorithm [D]. Changchun: Changchun University of Technology, 2022.

[5] Gao Z , Gao Y , Hu Y , et al. Application of Deep Q-Network in Portfolio Management[C]// 2020 5th IEEE International Conference on Big Data Analytics (ICBDA). IEEE, 2020.

[6] Lw A , Xs A , Min X A , et al. Portfolio trading system of digital currencies: A deep reinforcement learning with multidimensional attention gating mechanism - ScienceDirect[J]. Neurocomputing, 2020, 402:171-182.

[7] Ren X , Jiang Z , Su J . The Use of Features to Enhance the Capability of Deep Reinforcement Learning for Investment Portfolio Management[C]// 2021 IEEE 6th International Conference on Big Data Analytics (ICBDA). IEEE, 2021.

[8] Jbc A , Msk A , Gdm A , et al. A Q-learning agent for automated trading in equity stock markets[J]. Expert Systems with Applications, 163.

[9] Wu X , Chen H , Wang J , et al. Adaptive Stock Trading Strategies with Deep Reinforcement Learning Methods[J]. Information Sciences, 2020, 538.

[10] Feng M. A Method for Integrating Deep Learning Models Based on Investment Portfolios [D].Sichuan:Southwest Jiaotong University,2020.