

## Research Article

# Stock Trading Strategies Based on Deep Reinforcement Learning

Yawei Li,<sup>1</sup> Peipei Liu ,<sup>2</sup> and Ze Wang<sup>2</sup>

<sup>1</sup>*School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan 250014, China*

<sup>2</sup>*School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250014, China*

Correspondence should be addressed to Peipei Liu; 192115011@mail.sdufe.edu.cn

Received 1 August 2021; Revised 30 December 2021; Accepted 4 February 2022; Published 1 March 2022

Academic Editor: Cristian Mateos

Copyright © 2022 Yawei Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The purpose of stock market investment is to obtain more profits. In recent years, an increasing number of researchers have tried to implement stock trading based on machine learning. Facing the complex stock market, how to obtain effective information from multisource data and implement dynamic trading strategies is difficult. To solve these problems, this study proposes a new deep reinforcement learning model to implement stock trading, analyzes the stock market through stock data, technical indicators and candlestick charts, and learns dynamic trading strategies. Fusing the features of different data sources extracted by the deep neural network as the state of the stock market, the agent in reinforcement learning makes trading decisions on this basis. Experiments on the Chinese stock market dataset and the S&P 500 stock market dataset show that our trading strategy can obtain higher profits compared with other trading strategies.

## 1. Introduction

Stock trading is the process of buying and selling stocks to obtain investment profit. The key to stock trading is to make the right trading decisions at the right times, that is, to develop a suitable trading strategy [1]. In recent years, many studies have been based on machine learning methods to predict stock trends or prices to implement stock trading. However, long-duration prediction of the price or trend of the stock is not reliable. Besides, the trading strategy based on stock price prediction is static [2, 3]. The stock market is affected by many factors [4–6], such as changes in investor psychology and company policies, natural disasters, emergencies, etc., stock price fluctuates greatly. Compared with a static trading strategy, a dynamic trading strategy can make trading decisions dynamically according to the changes of the stock market, which has greater advantages.

At present, an increasing number of studies implement dynamic trading strategies based on deep reinforcement learning. Reinforcement learning gains increasing attention after AlphaZero defeated humans [7], has the ability of independent learning and decision-making, and has been successfully applied in the field of game playing [8, 9], unmanned driving [10, 11], and helicopter control [12].

Reinforcement learning solves the sequential decision-making problem, which can be applied to stock trading to learn dynamic trading strategies. Nevertheless, reinforcement learning lacks the ability to perceive the environment. The combination of deep learning and reinforcement learning (i.e., deep reinforcement learning) solves this problem and has more advantages when it has the decision-making ability of reinforcement learning and perception ability of deep learning.

One of the challenges when implementing stock trading based on deep reinforcement learning is the correct analysis of the state of the stock market. Financial data is nonlinear and unstable. Most of the existing studies on stock trading based on deep reinforcement learning analyze the stock market through stock data [13–15]. However, there is noise in stock data, which affects the final analysis results. Technical indicators can reflect the changes in the stock market from different perspectives and reduce the influence of noise [16, 17]. There are studies that convert financial data into two-dimensional images for analyzing the stock market [18–22]. Different data sources reflect the changes in the stock market from different perspectives. Compared with the analysis of the stock market based on a single data source, multisource data can integrate the characteristics of

different data sources, which is more conducive to the analysis of the stock market. However, the fusion of multisource data is difficult.

To a deeper analysis of the stock market and learn the optimal dynamic trading strategy, this study proposes a deep reinforcement learning model and integrates multisource data to implement stock trading. Through the analysis of stock data, technical indicators, and candlestick charts, we obtain a deeper feature representation of the stock market, which is conducive to learning the optimal trading strategy. Besides, the setting of the reward function in reinforcement learning cannot be ignored. In stock trading, investment risk should be paid attention to while considering returns and reasonably balance risk and returns. Sharpe ratio (SR) represents the profit that can be obtained under certain risks [23]. In this study, the reward function takes investment risk into consideration and combines SR and profit rate (PR) as the reward function to promote the learning of optimal trading strategies.

To verify the effectiveness of the trading strategy learned by our proposed model, we compare it with other trading strategies based on practical trading data. For stocks with different trends, our trading strategy obtained higher PR and SR, which has better robustness. In addition, we conduct ablation experiments, and the experimental results show that the trading strategy learned from analyzing the stock market based on multisource data is better than those learned from analyzing the stock market based on a single data source. The main contributions of this paper are as follows:

- (i) A new deep reinforcement learning model is proposed to implement stock trading and integrate the stock data and candlestick charts to analyze the stock market, which is more helpful to learn the optimal dynamic trading strategy.
- (ii) A new reward function is proposed. In this study, investment risk is taken into account, and the sum of SR and PR is taken as the reward function.
- (iii) The experimental results show that the trading strategy learned from the deep reinforcement learning model proposed in this paper can obtain better profits for stocks with different trends.

## 2. Related Work

In recent years, a mass of machine learning methods has been implemented in stock trading. Investors make trading decisions based on their judgment of the stock market. However, due to the influence of many factors, they cannot make correct trading decisions based on the changes in the stock market in time. Compared with traditional trading strategies, machine learning methods can learn trading strategies by analyzing information related to the stock market and discovering profit patterns that people do not know about without professional financial knowledge, which have more advantages.

There are some studies based on deep learning methods to implement stock trading. Deep learning methods usually implement stock trading by predicting the future trend or

price of stock [24–27]. In the financial field, deep learning methods are used for stock price prediction because they can obtain temporal characteristics from financial data [28, 29]; Chen et al. [30] analyzed 2D images transformed from financial data through a convolutional neural network (CNN) to classify the future price trend of stocks. When implementing stock trading based on the deep learning method, the higher the accuracy of the prediction, the more helpful the trading decision. On the contrary, when the prediction result deviates greatly from the actual situation, it will cause the fault trading decision. In addition, the trading strategy implemented by such methods is static and cannot be adjusted in time according to the changes in the stock market.

Reinforcement learning can be used to implement stock trading by self-learning and autonomous decision-making. Chakole et al. [31] used Q-learning algorithm [32] to find the optimal trading strategy, in which the unsupervised learning method *K*-means and candlestick chart were, respectively, used to represent the state of the stock market. Deng et al. [33] proposed a model Deep Direct Reinforcement Learning and added fuzzy learning, which is the first attempt to combine deep learning and reinforcement learning in the field of financial transactions. Wu et al. [34] proposed a long short-term memory based (LSTM-based) agent that could perceive stock market conditions and automatically trade by analyzing stock data and technical indicators. Lei et al. [35] proposed a time-driven feature aware jointly deep reinforcement learning model (TFJ-DRL), which combines gated recurrent unit (GRU) and policy gradient algorithm to implement stock trading. Lee et al. [36] proposed HW\_LSTM\_RL structure, which first used wavelet transforms to remove noise in stock data, then based on deep reinforcement learning to analyze the stock data to make trading decisions.

Existing studies on stock trading based on deep reinforcement learning mostly analyze the stock market through a single data source. In this study, we propose a new deep reinforcement learning model to implement stock trading, and analyze the state of the stock market through stock data, technical indicators, and candlestick charts. In our proposed model, firstly, different deep neural networks are used to extract the features of different data sources. Secondly, the features of different data sources are fused. Finally, reinforcement learning makes trading decisions according to the fused features and continuously optimizes trading strategies according to the profits. The setting of reward function in reinforcement learning cannot be ignored. In this study, the SR is added to the reward function setting, and the investment risk is taken into account while considering the profits.

## 3. Methods

We propose a new deep reinforcement learning model and implement stock trading by analyzing the stock market with multisource data. In this section, first, we introduce the overall deep reinforcement learning model, then the feature extraction process of different data sources is described in

detail. Finally, the specific application of reinforcement learning in stock trading is introduced.

**3.1. The Overall Structure of the Model.** Implementing stock trading based on deep reinforcement learning and correctly analyzing the state of the stock market is more conducive to learning the optimal dynamic trading strategy. To obtain the deeper feature representation of the stock market state and learn the optimal dynamic trading strategy, we fuse the features of stock data, technical indicators, and candlestick charts. Figure 1 shows the overall structure of the model.

The deep reinforcement learning model we propose can be divided into two modules, the deep learning module for extracting features of different data sources and the reinforcement learning module for making trading decisions. Candlestick charts features are extracted by the CNN and bidirectional long short-term memory (BiLSTM); stock data and technical indicators are as the input of the LSTM network for feature extraction. After extracting the features of different data sources, contacting the features of the different data sources to implement feature fusion, the fused features can be regarded as the state of the stock market, and the reinforcement learning module makes trading decisions on this basis. In addition, in the reinforcement learning module, the algorithms used are Dueling DQN [37] and Double DQN [38].

**3.2. Deep Learning Module.** The purpose of this study is to obtain a deeper feature representation of the stock market environmental state through the fusion of multisource data to learn the optimal dynamic trading strategy. Although raw stock data can reflect changes in the stock market, they contain considerable noise. To reduce the impact of noise and perceive the changes of the stock market more objectively and accurately, relevant technical indicators are used as one of the data sources for analyzing the stock market in this study. Candlestick charts can reflect the changes in the stock market from another perspective. This paper fuses the features of the candlestick charts.

**3.3. Stock Data and Technical Indicator Feature Extraction.** Due the noise in stock data, we use relevant technical indicators to reduce the impact of noise. The technical indicators reflect the changes in the stock market from different perspectives. In this paper, stock data and technical indicators are used as inputs to the LSTM network to better capture the main trends of stocks. The raw stock data we use include opening price, closing price, high price, low price, and trading volume. The technical indicators used in this paper are the MACD, EMA, DIFF, DEA, KDJ, BIAS, RSI, and WILLR. The indicators are calculated by mathematical formulas based on stock prices and trading volumes [39], as reported in Table 1.

To facilitate subsequent calculations, we perform missing value processing on the stock data. First, the stock data is cleaned, and the missing data are supplemented with 0. In addition, the input of the neural network must be a real

value, so we replaced the NaNs in the stock data and technical indicators with 0. Data with different value ranges may show gradient explosion during neural network training [42]. To prevent this problem, we normalize the stock data and technical indicators; normalization is performed to transform the data to a fixed interval. In this work, the stock data and technical indicators of each dimension are normalized, and the data are converted into ranges [0, 1]. The normalization formula is as follows:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \quad (1)$$

where  $X$  represents the original data,  $X_{\min}$  and  $X_{\max}$  represent the minimum and maximum values of the original data, respectively,  $X_{\text{norm}}$  represents the normalized data. The neural network structure for extracting stock data and technical indicators is shown in Figure 2. LSTM network is a variant of recurrent neural network (RNN), and its unit structure is shown in Figure 3. LSTM solves the problem of gradient disappearance and gradient explosion in the long sequence training process. In the LSTM network,  $f$ ,  $i$ , and  $o$  represent a forget gate, an input gate, and an output gate, respectively. A forget gate is responsible for removing information from the cell state. The input gate is responsible for the addition of information to the cell state. The output gate decides which next hidden state should be selected.  $C_t$  is the state of the memory cell at time  $t$ ;  $\tilde{C}_t$  is the value of the candidate state of the memory cell at time  $t$ ;  $\sigma$  and  $\tanh$  are the sigmoid and tanh activation functions, respectively;  $W$  and  $b$  represent the weight and deviation matrix, respectively;  $x_t$  is the input vector;  $h_t$  is the output vector; in this paper,  $x_t$  is the data after the contacting of stock data and technical indicators,  $x_t$  and other specific calculation formulas are as follows:

$$\begin{aligned} x_t &= (\text{open, low, close, ..., MACD, RSI, Willr}) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t &= \sigma(W_o [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t). \end{aligned} \quad (2)$$

In the entire process of feature extraction of stock data and technical indicators, the stock data and technical indicators are first cleaned and normalized. Then, the normalized data are used as the input of the LSTM network for feature extraction. Finally, the final feature is obtained by feature extraction through the two-layer LSTM network.

**3.4. Candlestick Chart Feature Extraction.** To extract more informative features, in this study, historical stock data are transformed into candlestick charts; candlestick charts contain not only the candlestick but also other information, which can be divided into two parts, the upper part is the candlestick and moving average of the closing price, the lower part is the trading volume histogram and its moving

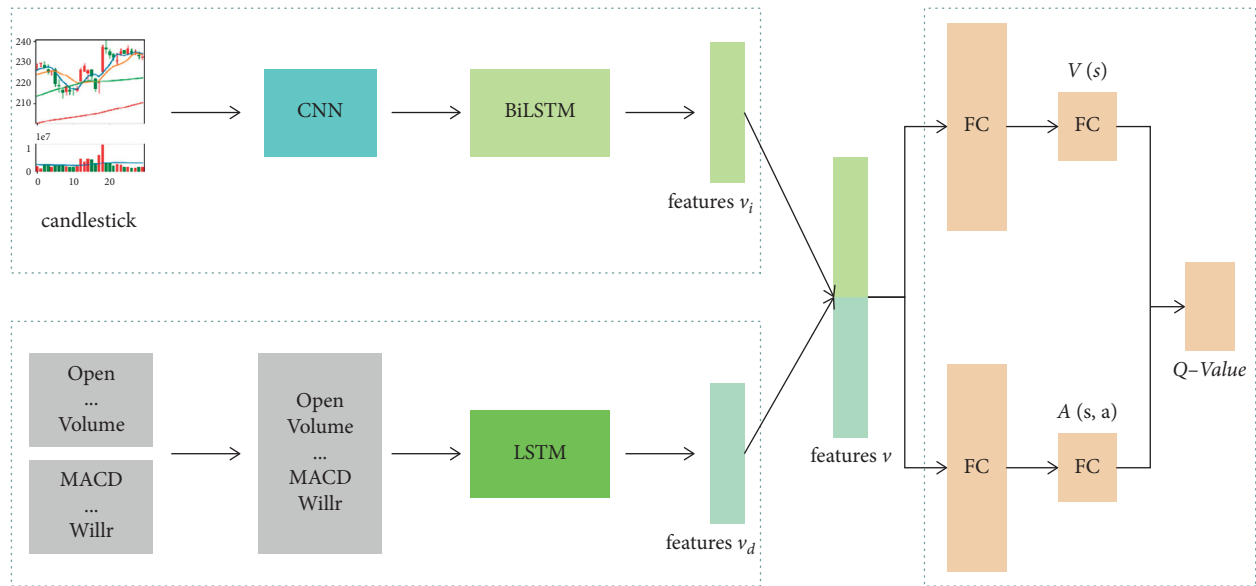


FIGURE 1: The overall structure of the model.  $v_i$  represents candlestick chart feature,  $v_d$  represents the feature of stock data and technical indicators, and the feature vector obtained by contacting these two feature vectors is used as the input of the two fully connected (FC) layers. In this paper, FC layers are used to construct the dueling DQN network; the two FC layers represent the advantage function  $A(s, a)$  and state value function  $V(s)$  in dueling DQN. The final  $Q$  value is obtained by adding the outputs of the two functions.

TABLE 1: The list of used technical indicators.

Technical indicator	Indicator description
MACD	Moving average convergence/divergence
EMA	Exponential moving average
DIFF	Diff
DEA	Difference exponential average
KDJ	Stochastics
BIAS	Bias
RSI	Relative strength index
WILLR	Williams %R

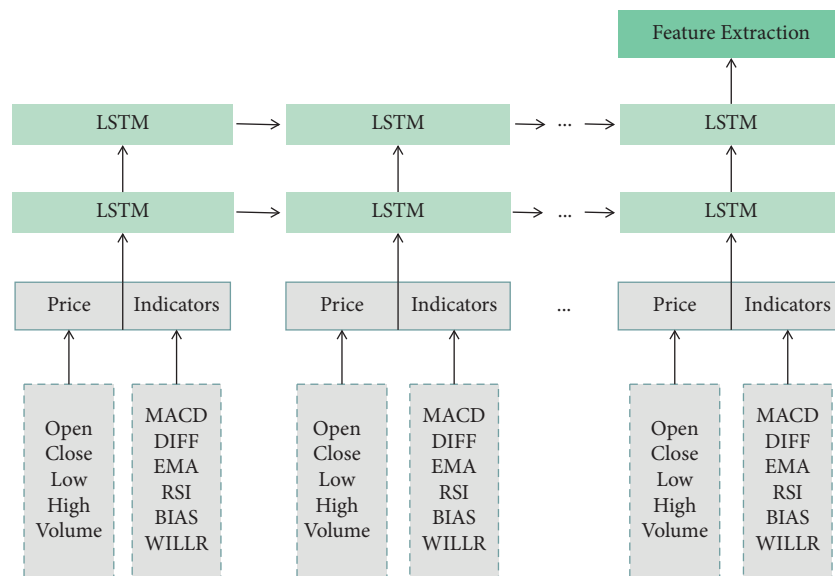


FIGURE 2: The network structure for extracting features of stock data and technical indicators.



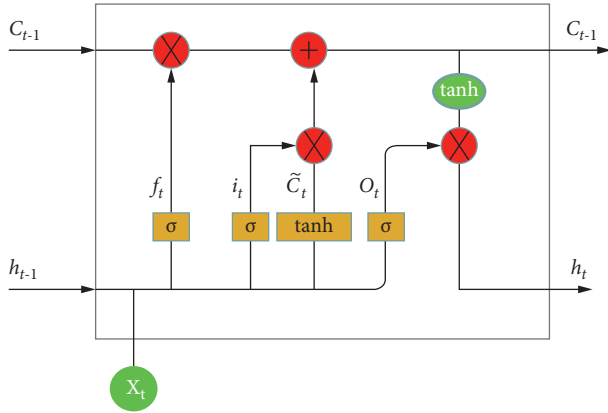


FIGURE 3: LSTM network unit structure.

average. Generally, the candlestick consists of body, upper shadow, and lower shadow. The body is the difference between the closing price and the opening price of the stock during the trading session, as shown in Figure 4. If the opening price is lower than the closing price, it indicates that the price is rising, this kind of candle is called a bullish candle, and the color of the candlestick is red. And if the open price is higher than the close price, it indicates that the price has fallen, and the color of the candlestick is green. For a bullish candlestick, the upper shadow is the difference between the high price and the close price, and the lower shadow represents the difference between the low price and the open price. For a bearish candlestick, the upper shading indicates the difference between the high price and the open price, and the lower shading indicates the difference between the low price and the close price. The trading time of stocks can range from one minute to one month. The candlestick chart is based on days in this study.

The network structure of extracting candlestick chart features is shown in Figure 5. In this study, we first obtain the features of the candlestick chart through three layers of convolution and pooling, then transform the obtained vector, input the BiLSTM network for feature extraction, and finally obtain the final features.

**3.5. Reinforcement Learning Module.** Reinforcement learning includes agent, environment, state, action, and reward. The agent chooses action according to the environmental state, which will get the immediate reward every time it chooses an action. The agent constantly adjusts the learning strategy according to the reward value to obtain the largest cumulative reward value. For example, in the process of stock trading, if the trading action selected by the agent gains a profit, it will get a positive reward value. In contrast, if there is a loss after choosing a trade action, the agent will get a negative reward value. Reward promotes the agent to make the correct action in future behavior choices. Most previous works used trading profits as an immediate reward of reinforcement learning to optimize the trading strategies. However, this only considers the changes in the profits after each trading action is taken and does not consider the investment risk. In this paper, we take investment risk into

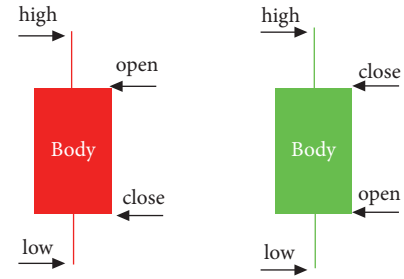


FIGURE 4: Candlestick representation.

consideration, SR is an indicator for evaluating transactions and is used to optimize the trade-off between profitability and risk. SR is the expected return minus the risk-free rate and then divided by the variance of the return. Considering both investment risk and return change, the reward function obtained by the sum of the two is more advantageous than the reward function based on return change, and this is proved by experiments, so the immediate reward is set to the sum of PR and the SR, the specific formula is as follows:

$$SR = \frac{E(R_p) - R_f}{\sigma_p},$$

$$PR = \frac{P_t - P_{t-1}}{P_{t-1}},$$

$$r = SR + PR,$$

(3)

$$R_t = \sum_{k=0}^T \gamma^k r_{t+k+1},$$

where  $P_t$  is the sum of the assets owned by the investors at time  $t$ ,  $E(R_p)$  is the expected portfolio return,  $R_f$  is the risk-free rate, and  $\sigma_p$  is the standard deviation of the return,  $R_t$  is cumulative rewards,  $\gamma \in (0, 1)$ , which is a discount factor.

In this paper, we combine the Double DQN algorithm and the Dueling DQN algorithm, both of which are improved algorithms based on the DQN algorithm. In the value-based deep reinforcement learning algorithm, actions are selected according to the Q value. In the DQN algorithm, there are two networks with the same structure, the main network and the target network. Initially, the parameters of the two networks are the same. During the training process, the target network and the main network update the parameters in different ways. In the DQN algorithm, under the  $\epsilon$ -greedy strategy, the agent that has a greater probability chooses the action corresponding to the maximum Q value, which will cause the Q value to be overestimated.

Compared with the DQN algorithm, the Dueling DQN algorithm changes the calculation method of the Q value through the addition of the state value function  $V(s)$  and advantage function  $A(s, a)$ . The value function  $V(s)$  is used to evaluate the quality of the state, and the advantage function  $A(s, a)$  is used to evaluate the quality that the agent chooses action  $a$  in state  $s$ . The calculation formula of the Q value is as follows:

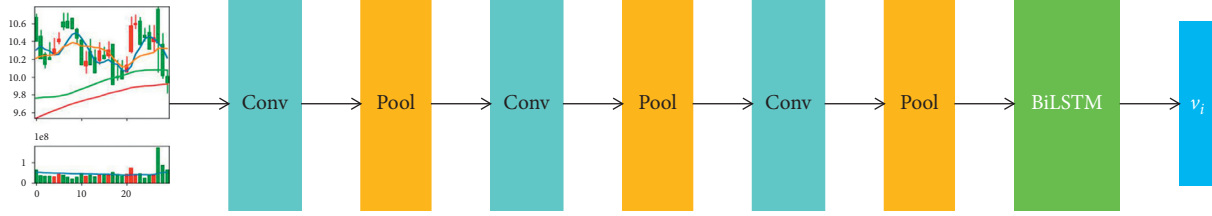


FIGURE 5: The network structure for extracting features of a candlestick chart.  $v_i$  is the final features.

$$Q(s, a; \theta, \alpha, \beta) = A(s, a; \theta, \alpha) + V(s; \theta, \beta), \quad (4)$$

where  $\alpha$  and  $\beta$ , respectively, represent the parameters in the value function  $V(s)$  and advantage function  $A(s, a)$ ,  $\theta$  represents other parameters in the deep reinforcement learning modal.

Double DQN changes the calculation of the  $Q$  value of the target network and solves the  $Q$  value overestimation problem in the DQN algorithm, which can be combined with the Dueling DQN algorithm to improve the overall performance of the model. The formula for calculating the  $Q$  value of the target network in the Double DQN algorithm is as follows:

$$Y_t = r_{t+1} + \gamma Q\left(s_{t+1}, \underset{a}{\operatorname{argmax}} Q(s_{t+1}; \theta_t); \theta'_t\right), \quad (5)$$

where  $\theta$  and  $\theta'$  represent the parameters in the main network and target network, respectively.

The loss function is the mean square error of the  $Q$  value of the main network and the target network. The formula is shown as follows:

$$L(\theta) = E[(Y_t - Q(s, a; \theta))^2]. \quad (6)$$

In this study, we analyze the stock market from stock data, technical indicators, and candlestick charts and fuse the features of the different data sources to obtain stock market state features representation and help the agent learn the optimal dynamic trading strategy. Trading action  $a_t = \{\text{long, neural, short}\} = \{1, 0, -1\}$ , long, neural and short represent buy, hold, and sell, respectively. When the trading action is long, cash is converted into stock as much as possible, and when the trading action is short, all shares are sold into cash. In addition, transaction costs in stock trading cannot be ignored. High-frequency transactions result in higher costs; the transaction cost in this paper is 0.1% of the stock value [40]. The trading process is shown in Algorithm 1.

**Input:** stock data, technical indicators, candlestick chart;

Initialize the experience replay memory  $D$  to capacity  $C$ ;

Initialize the main  $Q$  network with random weights  $\theta$ ;

Initialize the target  $Q$  network with  $\theta' = \theta$ ;

**for** episode 1 to  $N$  do

**for**  $t = 1$  to  $T$  do

        The sum of the features extracted by the deep learning module represents the environment state  $\mathbf{s}_t$ ;  
        With the probability  $\mathcal{Z}$  choose a random action  $\mathbf{a}_t$ ;  
        Otherwise select  $a_t = \underset{a}{\operatorname{argmax}} Q(s_t, a; \theta)$  ;  
        Get the reward  $\mathbf{r}_t$  and next state  $\mathbf{s}_{t+1}$ ;  
        Store the transaction  $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{s}_{t+1})$  to  $D$ ;

**if**  $t \% n = 0$  **then**

        Sample minibatch  $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{s}_{t+1})$  randomly from  $D$ ;

        Set:

$$Y_t = \begin{cases} r_t & \text{if the state } s_{t+1} \text{ is terminal,} \\ r_t & \text{otherwise.} \end{cases}$$

        Train the network with loss function

$$L(\theta) = E[(Y_t - Q(s, a; \theta))^2]$$

        Update the target network parameters  $\theta' = \theta$  every  $N$  steps;

**end if**

**end for**

**end for**

## 4. Experiment and Results

This section mainly introduces the dataset, evaluation metrics, comparison methods, implementation details, and experimental result analysis.

**4.1. Datasets.** In this study, we verify the dynamic trading strategy learned from the proposed model on datasets of Chinese stocks and S&P 500 stock market stocks and compare them with other trading strategies. The period range of the dataset is from January 2012 to January 2021. The training period ranges from January 2012 to December 2018; the testing period ranges from January 2019 to January 2021. For stock data, it includes the daily open price, high price, low price, close price, and trading volume of the stock, as shown in Table 2.

**4.2. Metrics.** The evaluation indicators used in this paper are PR, the annualized rate of return (AR), SR, and max drawdown duration (MDD). The details are as follows:

- (i) PR refers to the difference between the assets owned at the end of the stock transaction and the original assets divided by the original assets.

TABLE 2: Stock data structure example.

Date	Open	High	Low	Close	Volume
...	...	...	...	...	...
2018/4/5	177.01	185.49	175.75	91.98	2749769
2018/4/6	185.30	190.95	184.08	94.13	2079350
2018/4/7	189.06	194.00	186.98	94.28	2565820
2018/4/8	192.03	196.00	188.20	93.98	1271339
...	...	...	...	...	...

- (ii) AR is the ratio of the profits to the principal of the investment period of one year. The formula is defined as follows:

$$AR = \frac{\text{total profits}}{\text{principal}} * \frac{365}{\text{trading days}} * 100. \quad (7)$$

- (iii) SR is a standardized comprehensive evaluation index, which can consider both profits and risks at the same time to eliminate the adverse impact of risk factors on performance evaluation.
- (iv) MDD refers to the maximum losses that can be borne during trading, and the lower the value, the better the performance of the trading strategy.

#### 4.3. Baselines

- (i) Buy and Hold (B&H) [41] refers to the construction of a certain portfolio according to the determined appropriate asset allocation ratio, and the maintenance of this portfolio during the appropriate holding period without changing the asset allocation status. And B&H strategy is a passive investment strategy.
- (ii) Based on the Q-learning algorithm, two models are proposed to implement stock trading [31]. The two models perceive the stock market environment in different ways, model 1 analyzes the stock market through the k-means method, model 2 analyzes the stock market through a candlestick chart, and the experimental results show that model 1 performs better than model 2, so we only compare with model 1. In model 1, the size  $n$  of clusters is set to 3, 6, and 9, and we compare them, respectively.
- (iii) A LSTM-based agent is proposed to learn the temporal relationship between data and implement stock trading [34], and the effects of different combinations of technical indicators on trading strategies are verified. In this paper, only a group of technical indicators with the best results are compared.
- (iv) The time-driven feature-aware jointly deep reinforcement learning (TFJ-DRL) [35] model, which used GRU to extract temporal features of stock data and technical indicators and implement stock trading through the policy gradient algorithm.
- (v) HW\_LSTM\_RL [36] is a structure that combines wavelet transformation and deep reinforcement

learning for stock trading and is a relatively new method.

**4.4. Implementation Details.** This study is based on deep reinforcement learning to implement stock trading and fuses the features of stock data, technical indicators, and candlestick charts as the state of the stock market. LSTM network extracts features of stock data and technical indicators, the size of the hidden layer is 128, and the size of the candlestick chart is  $390 \times 290$ . In the process of learning the optimal trading strategy, an episode is the trading period ranges from January 2012 to December 2018. The episode in training is 200. In the  $\epsilon$ -greedy strategy of reinforcement learning, the  $\epsilon = 0.8$ . The length of the sliding time window is set to 30 days, and the learning rate is 0.0001.

#### 4.5. Experimental Results

**4.5.1. Comparative Experiment on the Chinese Stock Dataset.** We select 10 Chinese stocks with different trends for comparative experiments, and the initial amount is 100,000 Chinese Yuan (CNY). The results of the experiment are shown in Tables 3–6. We select three stocks with different trend changes to further demonstrate the PR changes, as shown in Figure 6.

The traditional trading strategy B&H is a passive trading strategy, which has an advantage in stocks with rising prices. However, it does not perform well for stocks with large price fluctuations or downward trends. It can be seen from Figure 6 that for the stock 002460 with an upward trend, the B&H trading strategy can obtain a higher PR, while for the other two stocks, 601101 and 600746, with different trends, the PR obtained are not as good as the other trading strategies. Trading strategies learned based on the Q-learning algorithm are dynamic, compared with the traditional trading strategy B&H. In most cases, the trading strategies learned based on model 1 can obtain higher PR, AR, SR, and lower MDD for stocks with different trends in different fields. Nonetheless, reinforcement learning lacks the ability to perceive the environment. Compared with the trading strategies learned based on the deep reinforcement learning model, the trading strategies learned based on model 1 do not have obvious advantages. LSTM\_based, TFJ-DRL, and HW\_LSTM\_RL are all methods based on deep reinforcement learning. The data sources analyzed by these methods are relatively single, compared with the traditional trading strategy B&H and trading strategies learned based on Model 1. The trading strategies learned by these methods can obtain more profits for stocks with different trends in different fields. From the experimental results, we can see that the dynamic trading strategies learned by our proposed model have better performance. It can be seen from Tables 3–6 that stocks with different trends, the trading strategy learned by the model proposed in this paper, have better performance. Compared with other trading strategies, the evaluation indicators of our trading strategy are the highest in most cases. On the whole, the average PR of our trading strategy is

TABLE 3: PR comparison of different methods in the Chinese stock market dataset.

Stock	B&H [41]	PR (%)						
		Model 1 [31]			LSTM-based [34]	TFJ-DRL [35]	HW_LSTM_RL [36]	Ours
		$n = 3$	$n = 6$	$n = 9$				
002460	365.62	-50.66	-19.48	-68.11	133.29	379.32	257.68	<b>418.12</b>
601101	-18.93	-14.38	-18.07	-19.77	-18.79	-28.59	-25.09	<b>-9.28</b>
600746	2.49	16.17	7.54	31.55	2.85	0.14	4.05	<b>32.30</b>
600316	438.22	472.61	480.32	524.03	492.31	502.73	536.28	<b>595.93</b>
600028	-19.22	20.87	22.35	19.56	18.70	12.49	21.38	<b>23.94</b>
600900	26.21	<b>28.25</b>	22.11	25.67	9.44	19.63	27.14	24.35
002129	189.41	193.78	163.50	173.92	176.40	178.05	190.31	<b>196.67</b>
600704	-2.91	10.89	16.54	12.34	9.91	15.26	18.28	<b>22.98</b>
600377	0.58	6.09	3.82	10.41	11.12	6.73	9.41	<b>14.05</b>
300122	285.45	268.52	259.87	298.91	275.26	290.35	301.48	<b>309.67</b>
Average	126.69	52.68	50.62	53.69	111.05	137.61	134.09	<b>162.87</b>

TABLE 4: AR comparison of different methods in the Chinese stock market dataset.

	AR (%)							
	B&H [41]	Model 1 [31]			LSTM-based [34]	TFJ-DRL [35]	HW_LSTM_RL [36]	Ours
Stock		$n = 3$	$n = 6$	$n = 9$				
002460	67.25	-51.75	-4.72	-100	45.64	68.09	59.14	<b>70.37</b>
601101	-4.73	-4.63	-7.73	-7.70	-4.64	-11.62	-8.97	<b>0.95</b>
600746	11.58	13.31	9.47	15.52	21.76	16.59	12.30	<b>28.18</b>
600316	72.53	74.23	74.72	76.08	68.28	67.12	72.29	<b>79.83</b>
600028	5.32	5.21	6.81	4.69	6.99	5.72	6.33	<b>7.39</b>
600900	12.35	<b>13.82</b>	10.47	12.04	5.83	11.36	12.74	11.69
002129	53.14	54.26	48.12	50.33	49.41	51.42	54.02	<b>58.90</b>
600704	2.32	7.43	10.06	8.49	9.32	9.95	11.47	<b>13.28</b>
600377	1.73	3.11	1.63	5.84	6.42	2.58	4.23	<b>7.70</b>
300122	60.18	57.12	55.84	61.40	60.52	62.47	59.24	<b>65.39</b>
Average	28.17	17.21	20.47	12.67	26.95	28.37	28.28	<b>34.37</b>

TABLE 5: SR comparison of different methods in the Chinese stock market dataset.

SR								
	B&H [41]	Model 1 [31]			LSTM-based [34]	TFJ-DRL [35]	HW_LSTM_RL [36]	Ours
Stock		$n = 3$	$n = 6$	$n = 9$				
002460	1.72	-1.01	-0.08	-1.56	1.04	1.75	1.47	<b>1.83</b>
601101	-0.13	-0.21	-0.41	-0.32	-0.13	-0.31	-0.24	<b>0.03</b>
600746	0.25	0.39	0.29	<b>0.62</b>	0.49	0.30	0.41	0.52
600316	1.77	1.79	1.83	1.89	1.65	1.73	1.92	<b>2.00</b>
600028	0.23	0.21	0.34	0.19	0.35	0.22	0.27	<b>0.49</b>
600900	0.74	<b>0.81</b>	0.67	0.72	0.34	0.68	0.71	0.70
002129	1.23	1.25	1.18	1.16	1.20	1.12	1.22	<b>1.34</b>
600704	0.08	0.35	0.41	0.39	0.38	0.40	0.43	<b>0.50</b>
600377	0.10	0.22	0.15	0.35	0.39	0.24	0.30	<b>0.46</b>
300122	1.64	1.67	1.52	1.69	1.48	1.61	1.70	<b>1.78</b>
Average	0.76	0.55	0.59	0.51	0.72	0.77	0.82	<b>0.97</b>

162.87, AR is 34.37, SR is 0.97, and MDD is 29.98, which are higher than other trading strategies.

**4.5.2. Comparative Experiment on the S&P 500 Stock Dataset.** To further verify the performance of the trading strategy learned by the model proposed in this paper, in

the S&P 500 dataset, we selected 10 stocks with different trends and compared them with other trading strategies; the initial capital is 100,000 USD. The results of the experiment are shown in Tables 7–10. In this section, we select ten stocks with different trends and show their price changes and PR changes in detail, as shown in Figure 7.



TABLE 6: MDD comparison of different methods in the Chinese stock market dataset.

MDD (%)								
	B&H [41]	Model 1 [31]			LSTM-based [34]	TFJ-DRL [35]	HW_LSTM_RL [36]	Ours
Stock		$n = 3$	$n = 6$	$n = 9$				
002460	41.32	60.70	50.50	69.37	56.68	48.05	41.32	<b>41.32</b>
601101	56.98	42.17	<b>39.50</b>	43.90	56.29	59.38	57.80	46.44
600746	42.58	44.56	40.65	<b>36.01</b>	52.47	53.53	39.68	49.84
600316	25.36	26.71	26.59	26.83	28.91	30.12	34.40	<b>25.12</b>
600028	50.03	53.17	51.77	56.82	17.20	43.95	23.39	<b>15.58</b>
600900	16.19	<b>15.29</b>	19.40	16.38	19.95	17.23	18.34	17.42
002129	35.48	35.21	38.76	34.92	35.48	42.83	32.57	<b>30.41</b>
600704	25.77	26.91	25.08	26.96	30.79	36.24	24.16	<b>23.37</b>
600377	19.76	19.84	20.62	18.26	20.31	23.48	18.22	<b>15.45</b>
300122	38.07	38.28	40.96	36.05	42.60	39.12	38.55	<b>34.89</b>
Average	35.15	36.28	35.58	36.55	36.07	39.39	32.84	<b>29.98</b>

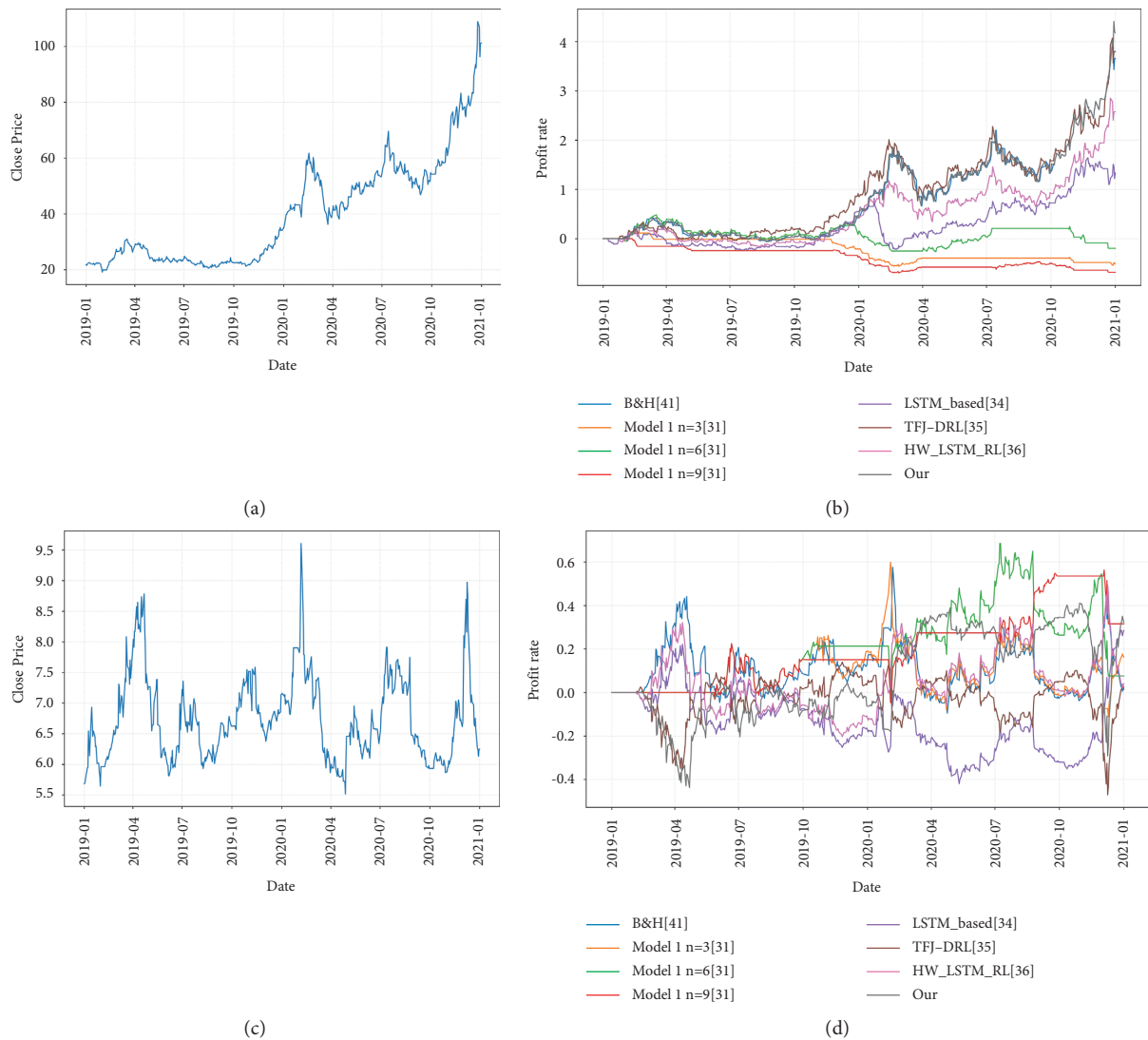


FIGURE 6: Continued.

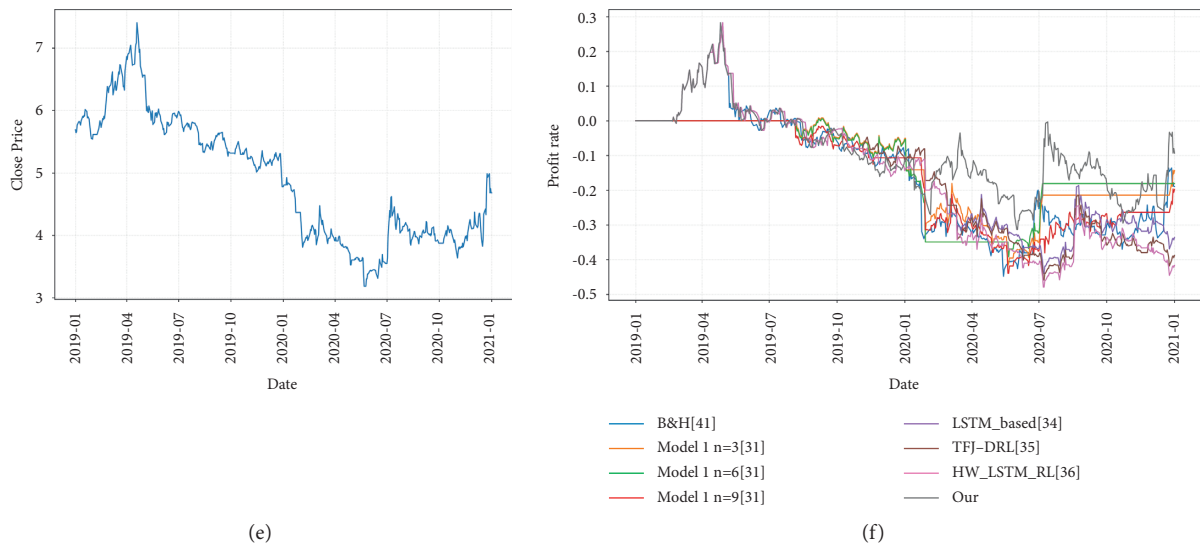


FIGURE 6: Changes in the price and PR of stocks with different trends. (a) 002460. (b) 002460. (c) 600746. (d) 600746. (e) 601101. (f) 601101.

TABLE 7: PR comparison of different methods in the S&amp;P 500 market dataset.

	PR (%)							
	B&H [41]	Model 1 [31]			LSTM-based [34]	TFJ-DRL [35]	HW_LSTM_RL [36]	Ours
Stock		$n = 3$	$n = 6$	$n = 9$				
AMD	296.09	323.08	315.36	337.61	301.28	283.45	336.72	<b>352.67</b>
AAL	-55.78	-11.73	-75.31	-28.66	-60.89	-70.49	-18.58	<b>-3.69</b>
BIO	122.72	136.15	116.40	129.37	128.24	133.61	140.27	<b>143.67</b>
BLK	79.56	83.41	75.94	84.28	74.25	78.97	82.62	<b>85.39</b>
TSLA	1060.14	1167.90	1005.81	986.43	958.22	1024.26	1130.22	<b>1184.59</b>
AAPL	216.41	218.39	206.28	224.83	197.34	209.41	213.70	<b>227.61</b>
GOOGL	54.76	22.87	34.83	61.24	39.29	50.87	52.97	<b>62.47</b>
IBM	0.41	2.46	5.51	-2.31	18.25	8.50	4.18	<b>28.30</b>
HST	-15.05	-10.81	-18.92	-14.06	-17.51	-14.32	-10.63	<b>-3.47</b>
PG	47.52	49.84	50.06	46.61	40.94	44.72	50.25	<b>52.04</b>
Average	180.68	198.16	171.60	182.53	167.94	174.90	198.17	<b>212.96</b>

TABLE 8: AR comparison of different methods in the S&amp;P 500 market dataset.

AR (%)								
	B&H [41]	Model 1 [31]			LSTM-based [34]	TFJ-DRL [35]	HW_LSTM_RL [36]	Ours
Stock		$n = 3$	$n = 6$	$n = 9$				
AMD	63.19	65.01	64.23	67.49	63.13	54.20	69.41	72.26
AAL	-13.27	-0.04	-48.11	-13.73	-20.70	-35.88	6.82	32.25
BIO	38.92	41.57	36.93	39.61	28.29	37.43	40.57	42.73
BLK	31.10	31.48	29.85	32.17	25.30	31.72	32.96	33.20
TSLA	99.24	100.03	97.15	96.09	84.61	94.83	96.25	103.85
AAPL	51.27	52.28	49.54	53.23	46.49	50.30	52.81	55.71
GOOGL	23.87	22.04	14.19	25.57	19.11	22.86	22.97	25.94
IBM	5.06	5.12	6.16	2.59	12.58	8.30	5.65	16.34
HST	4.16	4.69	3.73	4.01	5.04	4.92	6.44	8.92
PG	20.56	20.87	21.65	18.02	19.83	21.41	20.36	24.14
Average	32.41	34.30	27.53	32.51	28.37	29.01	35.42	41.43

TABLE 9: SR comparison of different methods in the S&P 500 market dataset.

Stock	SR							
	B&H [41]	Model 1 [31]			LSTM-based [34]	TFJ-DRL [35]	HW_LSTM_RL [36]	Ours
		$n = 3$	$n = 6$	$n = 9$				
AMD	1.55	1.64	1.59	1.69	1.42	1.30	1.61	<b>1.73</b>
AAL	−0.16	0	−0.56	−0.54	−0.24	−0.36	0.12	<b>0.42</b>
BIO	1.29	1.32	1.14	1.30	1.27	1.12	1.22	<b>1.36</b>
BLK	0.98	0.92	0.83	1.16	0.85	0.96	1.06	<b>1.24</b>
TSLA	2.08	2.11	1.87	1.79	1.69	1.92	2.13	<b>2.21</b>
AAPL	1.76	1.78	1.62	1.81	1.68	1.74	1.71	<b>1.85</b>
GOOGL	0.85	0.56	0.93	0.92	0.71	0.81	0.86	<b>1.11</b>
IBM	0.16	0.19	0.23	0.10	0.42	0.28	0.21	<b>0.55</b>
HST	0.08	0.09	0.05	0.07	0.05	0.07	0.11	<b>0.16</b>
PG	0.89	0.91	0.93	0.75	0.65	0.90	0.82	<b>0.98</b>
Average	0.95	0.95	0.86	0.91	0.85	0.87	0.99	<b>1.16</b>

TABLE 10: MDD comparison of different methods in the S&P 500 market dataset.

Stock	MDD (%)							
	B&H [41]	Model 1 [31]			LSTM-based [34]	TFJ-DRL [35]	HW_LSTM_RL [36]	Ours
		$n = 3$	$n = 6$	$n = 9$				
AMD	34.28	34.06	34.82	33.95	35.47	39.05	30.41	<b>32.11</b>
AAL	74.72	<b>30.20</b>	82.01	41.15	77.64	76.10	48.31	34.06
BIO	19.72	19.23	21.65	19.28	25.83	22.64	20.16	<b>18.09</b>
BLK	42.27	45.21	49.32	38.75	50.28	48.45	40.91	<b>36.27</b>
TSLA	33.73	31.37	36.04	38.61	41.29	30.73	35.20	<b>27.06</b>
AAPL	20.37	21.38	23.09	19.71	22.74	21.29	18.43	<b>16.51</b>
GOOGL	32.41	43.69	38.62	44.41	20.22	30.84	22.52	<b>24.01</b>
IBM	38.96	33.68	31.71	32.65	31.60	30.37	24.62	<b>30.55</b>
HST	52.05	47.26	49.25	47.81	59.18	48.20	44.36	<b>43.08</b>
PG	23.14	23.27	27.18	29.02	33.38	30.43	25.58	<b>20.06</b>
Average	37.17	32.93	39.37	34.53	39.76	37.81	31.05	<b>28.18</b>

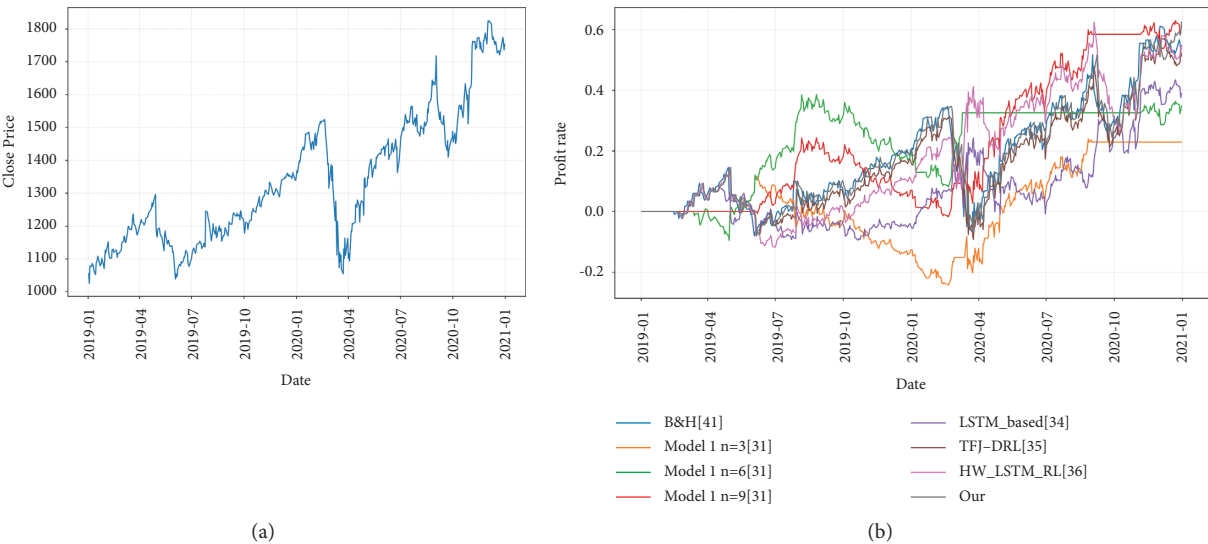


FIGURE 7: Continued.

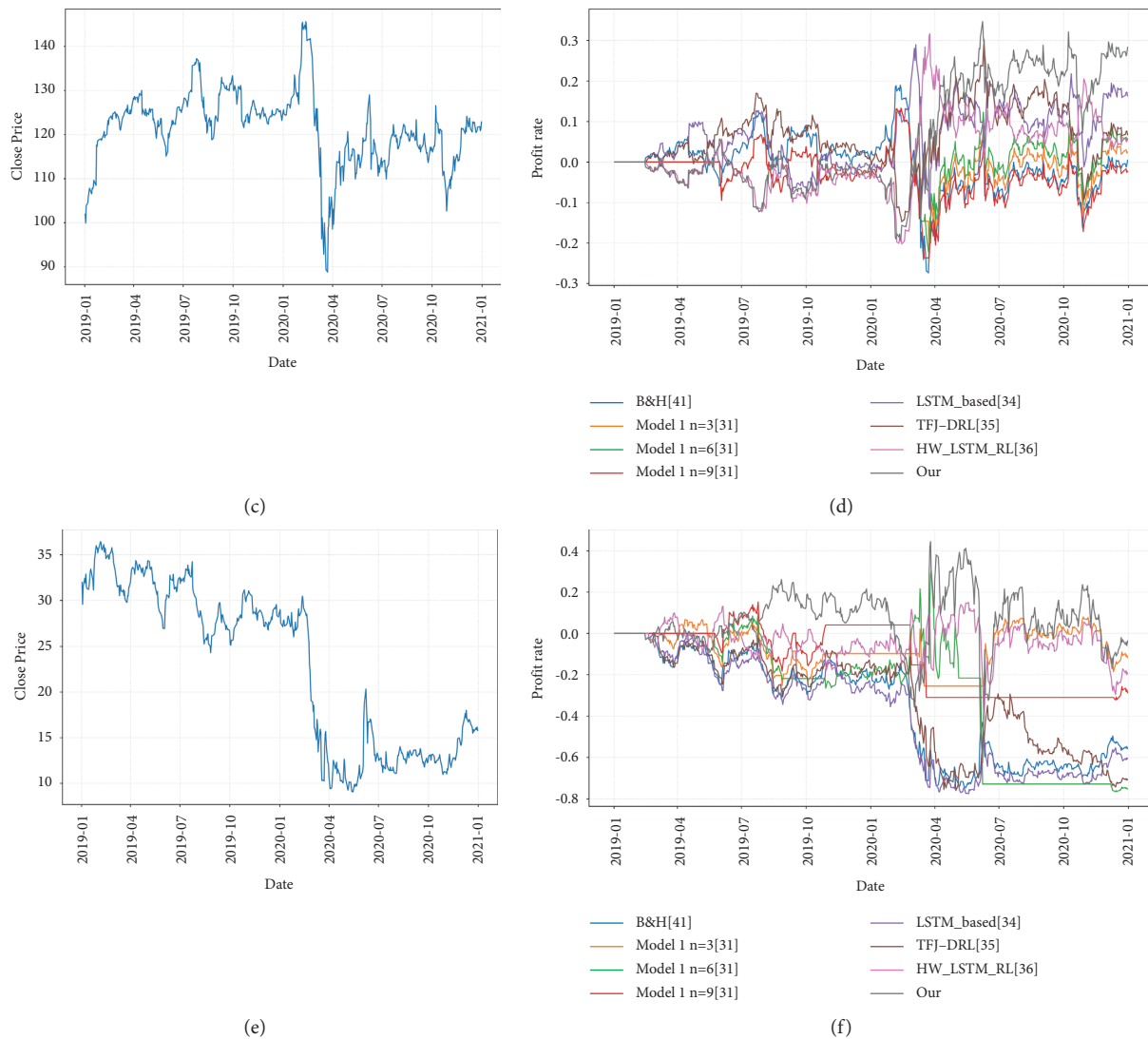


FIGURE 7: Changes in the price and PR of stocks with different trends. (a) GOOGL. (b) GOOGL. (c) IBM. (d) IBM. (e) AAL. (f) AAL.

For stocks with different trends in S&P 500, our trading strategy also has a better performance. It can be seen from Tables 7–10 that compared with other trading strategies, our trading strategy can obtain higher yields, SR, AR, and MDD, and on the whole, our PR reached 212.96, SR reached 1.16, obviously higher than other trading strategies.

To further verify the performance of our proposed model in different stock markets, we conducted Mann–Whitney  $U$  test on the profits rate of 20 stocks selected from the Chinese stock market and the S&P 500 stock market. The results showed that  $P = 0.677 > 0.05$ , indicating that there is no significant difference between the returns obtained by our model in the Chinese stock market and the S&P 500 stock market, and it has good generalization ability.

**4.5.3. Reward Function Comparison Experiment.** In this section, we set the reward function with SR and without SR and select two stocks from the Chinese stock market and the S&P 500 stock market for comparison experiments. The

stocks selected from the Chinese stock market are 600746 and 601101, and the stocks selected from the S&P 500 are GOOGL and IBM, the training period ranges from January 2012 to December 2018, and the testing period ranges from January 2019 to January 2021. The experimental results are shown in Table 11.

It can be seen from the experimental results in Table 11 that compared to the reward function without the SR, when the reward function contains the SR, the learned trading strategies have a better performance overall. Different from the existing algorithmic trading based on deep reinforcement learning, most of which take profit rate as reward function, this study takes investment risk into account and adds SR and profit rate as reward function, and the learned trading strategy obtains higher PR, AR, SR, and MDD.

**4.5.4. Ablation Experiments.** In this section, to verify the effectiveness of multisource data fusion, we conduct an ablation experiment. Three groups of comparative

TABLE 11: Comparison results of the reward function experiment.

Stock	Reward function with SR				Reward function without SR			
	PR (%)	AR (%)	SR	MDD (%)	PR (%)	AR (%)	SR	MDD (%)
600746	595.93	28.18	0.52	28.18	528.31	25.72	0.46	33.39
601101	-9.28	0.95	0.03	46.44	-14.99	-2.31	-0.06	48.54
GOOGL	62.47	25.94	1.11	24.01	52.18	22.90	0.95	29.40
IBM	28.30	16.34	0.55	30.55	24.39	14.28	0.40	35.20



FIGURE 8: Profit curves of the ablation experiment on stock GOOGL.

TABLE 12: Comparison results of ablation experiments on stock GOOGL.

	PR (%)	AR (%)	SR	MDD (%)
Group one	44.74	22.36	0.95	27.35
Group two	54.76	23.70	0.98	25.61
Group three	<b>62.47</b>	<b>25.94</b>	<b>1.11</b>	<b>24.01</b>

experiments are carried out, all of which are based on deep reinforcement learning to implement stock trading. The first group analyzes the stock market through stock data and technical indicators; the second group analyzes the stock market through a candlestick chart; and the third group analyzes the stock market through stock data, technical indicators, and candlestick chart. We select the trading data of GOOGL stock from January 2012 to January 2021 as the dataset for this section, in which January 2012 to December 2018 is the training data, and January 2019 to January 2021 is the test data. The comparison results are shown in Figure 8 and Table 12.

The experimental results show that compared with the trading strategies learned in the first two groups, the trading strategies learned in the third group can obtain higher PR, SR, AR, and lower MDD. This also proves that the analysis of multisource data can obtain a deeper feature representation of the stock market, which is more conducive to learning the optimal trading strategy.

## 5. Conclusion

Correct analysis of the stock market state is one of the challenges when implementing stock trading based on deep reinforcement learning. In this research, we analyze multisource data based on deep reinforcement learning to

implement stock trading. Stock data, technical indicators, and candlestick charts can reflect the changes in the stock market from different perspectives, we use different deep neural networks to extract the features of the data source and fuse features, and the fused features are more helpful to learn the optimal dynamic trading strategy.

It can be concluded from the experimental results that the trading strategies learned based on the deep reinforcement learning method can be dynamically adjusted according to the stock market changes and have more advantages. Compared with other trading strategies, our trading strategy has better performance for stocks with different trends, and the average SR value is the highest, which means that under the same risk, our trading strategy can get more profits. However, textual information such as investor comments and news events has an impact on the fluctuations of stock prices and cannot be ignored. It is important to obtain informative data from relevant texts for stock trading. In future research, we will consider different text information and train more stable trading strategies.

## Data Availability

The experimental data in this article can be downloaded from Yahoo Finance (<https://finance.yahoo.com/>).



## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This research was funded by the National Natural Science Foundation of China (Grant nos. 61972227 and 61902217), the Natural Science Foundation of Shandong Province (Grant nos. ZR2019MF051, ZR2020MF037, and ZR2019BF043), the NSFC-Zhejiang Joint Fund of the Integration of Informatization and Industrialization (Grant no. U1909210), Key Research and Development Project of Shandong Province (Grant nos. 2019GGX101007 and 2019GSF109112), and the Science and Technology Plan for Young Talents in Colleges and Universities of Shandong Province (Grant no. 2020KJN007).

## References

- [1] Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," *IEEE Access*, vol. 7, pp. 108014–108022, 2019.
- [2] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 02, pp. 654–669, 2018.
- [3] X. Ding, Y. Zhang, and T. Liu, "Deep learning for event-driven stock prediction," in *Proceedings of the Twenty-fourth international joint conference on artificial intelligence*, pp. 2327–2333, Buenos Aires Argentina, 2015.
- [4] S. Carta, A. Corrigan, A. Ferreira, A. S. Podda, and D. R. Recupero, "A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning," *Applied Intelligence*, vol. 51, no. 8, pp. 889–905, 2021.
- [5] R. Chia, S. Y. Lim, P. K. Ong, and S. F. Teh, "Pre and post Chinese new year holiday effects: Evidence from Hong Kong stock market," *The Singapore Economic*, vol. 60, no. 04, pp. 1–14, 2015.
- [6] Q. Huang, T. Wang, D. Tao, and X. Li, "Biclustering learning of trading rules," *IEEE Transactions on Cybernetics*, vol. 45, no. 20, pp. 2287–2298, 2014.
- [7] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7076, pp. 354–359, 2017.
- [8] D. Silver, T. Hubert, J. Schrittwieser et al., "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.
- [9] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [10] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3389–3396, Singapore, June 2017.
- [11] P. Wolf, C. Hubschneider, M. Weber, and A. Bauer, "Learning how to drive in a real world simulation with deep q-networks," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, no. IV, pp. 244–250, Los Angeles, CA, USA, June 2017.
- [12] A. Y. Ng, H. J. Kim, M. I. Jordan, and S. Sastry, "Autonomous helicopter flight via reinforcement learning," in *Proceedings of the Conference and Workshop on Neural Information Processing Systems*, vol. 16, 2003.
- [13] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS One*, vol. 12, no. 7, p. e0180944, 2017.
- [14] K. Chen, Y. Zhou, and F. Dai, "A LSTM-based method for stock returns prediction: a case study of China stock market," in *Proceedings of the IEEE International Conference on Big Data*, pp. 2823–2824, Santa Clara, CA, USA, Nov 2015.
- [15] S. Karaoglu, U. Arpacı, and S. Ayvaz, "A deep learning approach for optimization of systematic signal detection in financial trading systems with big data," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 2017, no. Special Issue, pp. 31–36, 2017.
- [16] C. J. Neely, D. E. Rapach, J. Tu, and G. Zhou, "Forecasting the equity risk premium: the role of technical indicators," *Management Science*, vol. 60, no. 7, pp. 1772–1791, 2014.
- [17] A. Gorgulho, N. Rui, and N. Horta, "Applying a GA kernel on optimizing technical analysis rules for stock picking and portfolio composition," *Expert Systems with Applications*, vol. 38, no. 11, pp. 14072–14085, 2011.
- [18] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: time series to image conversion approach," *Applied Soft Computing*, vol. 70, pp. 525–538, 2018.
- [19] T. Kim, H. Y. Kim, and A. Montoya, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PLoS ONE*, vol. 14, no. 2, p. e0212320, 2019.
- [20] M. U. Gudelek, S. A. Boluk, and A. M. Ozbayoglu, "A deep learning based stock trading model with 2-D CNN trend detection," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pp. 1–8, Honolulu, HI, USA, 2017.
- [21] A. Tsantekidis, N. Passalis, and A. Tefas, "Forecasting stock prices from the limit order book using convolutional neural networks," in *Proceedings of the IEEE 19th Conference on Business Informatics*, vol. 1, pp. 7–12, Thessaloniki, Greece, 2017.
- [22] Y. Y. Chen, W. L. Chen, and S. H. Huang, "Developing arbitrage strategy in high-frequency pairs trading with filter-bank CNN algorithm," in *Proceedings of the IEEE International Conference on Agents*, pp. 113–116, Singapore, 2018.
- [23] W. F. Sharpe, "The sharpe ratio," *Journal of Portfolio Management*, vol. 21, no. 1, pp. 49–58, 1994.
- [24] S. Liu, C. Zhang, and J. Ma, "CNN-LSTM neural network model for quantitative strategy analysis in stock markets," *Neural Information Processing*, vol. 2017, pp. 198–206, 2017.
- [25] Y. Yan and D. Yang, "A stock trend forecast algorithm based on deep neural networks[J]," *Scientific Programming*, vol. 2021, Article ID 7510641, , 2021.
- [26] D. T. Tran, M. Magris, and J. Kanninen, "Tensor representation in high-frequency financial data for price change prediction," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pp. 1–7, Honolulu, HI, USA, 2017.
- [27] M. Dixon, D. Klabjan, and J. H. Bang, "Classification-based financial markets prediction using deep neural networks," *Algorithmic Finance*, vol. 6, no. 3-4, pp. 67–77, 2017.
- [28] J. Long, Z. Chen, and W. He, "An integrated framework of deep learning and knowledge graph for prediction of stock

- pricetrend: an application in Chinese stock exchange market," *Applied Soft Computing*, vol. 91, 2020.
- [29] S. W. Lee and H. Y. Kim, "Stock market forecasting with super-high dimensional time-series data using ConvLSTM, trend sampling, and specialized data augmentation," *Expert Systems with Applications*, vol. 161, 2020.
  - [30] J. F. Chen, W. L. Chen, C.-P. Huang, S.-H. Huang, and A.-P. Chen, "Financial time-series data analysis using deep convolutional neural networks," in *Proceedings of the 2016 17th International conference on cloud computing and big data*, pp. 87–92, Macau, China, 2016.
  - [31] J. B. Chakole, M. S. Kolhe, and G. D. Mahapurush, "A Q-learning agent for automated trading in equity stock markets," *Expert Systems with Applications*, vol. 163, 2021.
  - [32] C. J. C. H. Watkins, *Learning from delayed rewards*, 1989.
  - [33] Y. Deng, F. Bao, and Y. Kong, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 653–664, 2016.
  - [34] J. Wu, C. Wang, and L. Xiong, "Quantitative trading on stock market based on deep reinforcement learning," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–8, Budapest, Hungary, 2019.
  - [35] K. Lei, B. Zhang, and Y. Li, "Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading," *Expert Systems with Applications*, vol. 140, 2020.
  - [36] J. Lee, H. Koh, and H. J. Choe, *Learning to trade in financial time series using high-frequency through wavelet transformation and deep reinforcement learning*, Applied Intelligence, 2021.
  - [37] Z. Wang, T. Schaul, and M. hessel, "Dueling network architectures for deep reinforcement learning," in *Proceedings of the International conference on machine learning*, 2016.
  - [38] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, 2016.
  - [39] X. Wu, H. Chen, and J. Wang, *Adaptive Stock Trading Strategies with Deep Reinforcement Learning Methods*, Information Sciences, 2020.
  - [40] T. Théate and D. Ernst, "An application of deep reinforcement learning to algorithmic trading," *Expert Systems with Applications*, vol. 173, 2020.
  - [41] E. Chan, *Algorithmic trading: winning strategies and their rationale*, John Wiley & Sons, vol. 625, , 2013.
  - [42] A. J. Hussain, A. Knowles, P. J. G. Lisboa, and W. El-Deredy, "Financial time series prediction using polynomial pipelined neural networks," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1186–1199, 2008.