Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4514300, 10 pages
https://doi.org/10.1155/2022/4514300

Research Article

Financial Stock Investment Management Using Deep Learning Algorithm in the Internet of Things

Jianjuan Fan[1] and Shen Peng[2]

[1]*Faculty of Information, University of Toronto, Ontario, Canada*
[2]*Computer Science and Technology, Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Jianjuan Fan; jianjuan.fan@mail.utoronto.ca

Received 23 March 2022; Revised 11 April 2022; Accepted 22 June 2022; Published 16 July 2022

Academic Editor: Vijay Kumar

This paper aims to explore a new model to study financial stock investment management (SIM) and obtain excess returns. Consequently, it proposes a financial SIM model using deep Q network (DQN) as reinforcement earning (RL) algorithm and Long Short-Term Memory (LSTM) as deep neural network (DNN). Then, after training and optimization, the proposed model is back-tested. The research findings are as follows: the LSTM neural network (NN)-based model will import the observation of the market at each time and the change of transaction information over time. The LSTM network can find and learn the potential relationship between time series data. There are two hidden layers and one output layer in the model. The hidden layer is an LSTM structure and the output layer is the fully connected NN. DQN algorithm first stores the experience sample data of the agent-environment interaction into the experience pool. It then randomly selects a small batch of data from the experience pool to train the network. Doing so removes the correlation and dependence between samples so that the DNN model can better learn the value function in the RL task. The model can predict the future state according to historical information and decide which actions to take in the next step. Meanwhile, five stocks of Chinese A-shares are selected to form an asset pool. The initial 500,000 amount of the account is divided into five equal shares, which are invested and traded. Overall, the model account's rate of return (RoR) during the back-test is 32.12%. The Shanghai Stock Exchange (SSI) has risen by 19.157% in the same period. Thus, the model's performance has exceeded the SSI's in the same period. E stock has the maximum RoR of 78.984%. The RoR of A, B, and C stocks is 54.129%, 11.594%, and 9.815%, respectively. B stock presents a minimum RoR of 6.084%. All these stocks have got positive returns. Therefore, the proposed financial SIM based on the DL algorithm is scientific and feasible. The research content has certain significant reference for the DL-based financial SIM.

1. Introduction

The rapid domestic economic development provides venture capitalists (VCs) with huge profit space in the stock market. The stock market is paid more and more attention to by the people, including the academic, industrial, and individual VCs [1]. However, due to excessive numbers of individual VCs and serious irrational investments in the domestic market, the stock price rises or falls sharply, making the market cycle complex and unstable. Meanwhile, with the increasing expansion of available data, it is difficult to completely extract valuable information from complex data by human analysis alone [2]. Therefore, how to reasonably and effectively manage the investment of financial stocks has become a problem studied by more scholars worldwide. With the continuous expansion of data and the upgrading and renewal of computer software and hardware infrastructure, deep learning (DL) has achieved blowout development in the era of big data. The financial field usually contains a large amount of data. High dimensions and high noise characterize these data. The traditional econometric methods are sometimes not suitable for analyzing such data. Financial data prediction has always been a challenging problem in the financial field [3].

The DL can tackle the complex high-dimensional financial data and lend itself to learning control strategies through data representation. Therefore, the DL algorithm provides a new idea for financial stock investment management (SIM). To cite a few examples, Guo et al. [4] tried to build illiquidity and idiosyncratic volatility indexes. After controlling idiosyncratic volatility indexes, they found a significant positive correlation between illiquidity indexes and equity risk premium. Arnott and Harvey [5] demonstrated the principles and procedures of applying machine learning (ML) in the financial field from the dimensions of research motivation, statistical methods, sample data selection, model complexity, and model cross-validation. Simonian and Wu [6] used a ridge regression model to replicate hedge funds and avoid model overfitting through the objective function (OF) regularization. The present work proposes a method to select the best regularization term coefficient for a time series. The empirical results show that the ridge regression model outperforms the ordinary least squares (OLS) regression model over some time. Additionally, more investment strategies based on DL algorithms have come into the public view in recent years. However, most current DL-based trading strategies focus on portfolio screening and rise and fall prediction. There is still less research on how to reduce trading risk effectively. Based on the traditional single objective reinforcement learning (RL), this paper creatively puts forward a multiobjective deep reinforcement learning (DRL) model. It aims to reduce transaction risks and volatility while maximizing the total return of the account.

This paper intends to explore new models to study financial SIM and reduce stock volatility while obtaining excess returns. This study applies the deep Q network (DQN) algorithm to the financial SIM. Firstly, the Long Short-Term Memory (LSTM) neural network (NN) is studied, and the model based on LSTM is implemented. The transaction action is transformed to the specified value range with a one-layer fully connected network. Then, the DQN algorithm is used to train the model. The training results are normalized. Meanwhile, the gradient descent method (GDM) is used to optimize the model. Finally, the trained model is back-tested to verify the training results and prove the scientificity and feasibility. This research can improve the return and reduce the transaction risk through the multiobjective DRL model. It has a certain reference significance for the financial management based on the DL algorithm.

2. Financial SIM Based on DL

In the traditional investment theory, the investment method of stock market research is usually nonquantitative, and the most common classification is the fundamental analysis (FA) and technical analysis (TA) method [7]. The FA method analyzes a country, an industry, and a specific company from the perspective of politics, macroeconomy, and microeconomy. It judges whether the macroeconomic environment, interindustry competition, intraindustry competition, and corporate governance level can improve the stock price, thereby helping VCs decide whether to invest in the stock. VCs need to study the company's fundamentals, financial statements, and financial ratios before investing. The FA method uses income, growth expectation, RoR, net growth rate (NGR), and other macro data to evaluate the enterprise's current financial status and growth potential. It also quantifies the "internal value" of the enterprise and judges whether it is suitable for long-term investment [8, 9]. VCs who use FA believe that the stock price is mainly affected by enterprise-level, national-level, or even international-level political and economic factors. Studying the supply-demand curve, macroeconomic, psychological, and policy factors can predict the development trend of the financial market (FM). Product innovation has also been developed along with other indexes, such as financial ratio, management policy, and marketing strategy, to help study the fundamentals of the company. Meanwhile, VCs also collect information, such as the enterprise's financial report, industry dynamics, and market trends, to accurately analyze enterprises' basic value and growth potential [10, 11].

The premise of TA is that the historical data of stocks reflect the future stock prices [12]. TA extracts the enterprise's basic situation and holds that the price is the manifestation of all the fundamental FM dynamics factors. TA does not explicitly consider the enterprise's internal and external features when studying market price changes, thus avoiding analyzing subjective economic factors. Besides, TA strives to identify the market price fluctuation laws and contends that the market transaction data are sufficient to determine the price trend. The view—history will reappear—helps researchers model the historical financial assets into time series and convert the transaction data into stock indexes mathematically, such as moving average (MA), relative strength index (RSI), momentum (MOM), and exponential moving average (EMA). Then, they draw relevant charts and analyze the price changes of the FM [13, 14]. Additionally, TA inputs historical prices and technical indexes, thus simplifying market trend prediction into pattern recognition and output the price dynamics, being a preferred short-term investment analysis approach. Lastly, TA processes transaction data utilizing mathematical tools, thus being more flexible than the FA that considers various factors [15].

Although the traditional investment strategy has gained some achievements, it still has some defects. First of all, the traditional active investment analyzes and forecasts the FM based on VCs' subjective judgment to generate artificial strategies, which is partly limited by the human brain's information processing ability, and thus, only a few variables are considered. Secondly, behavioral finance points out that VCs' psychology is often characterized by overconfidence and risk underestimation as well as blind conformity or regret. Cognitive bias and emotionalization are easy to harm investment decision-making (DM) [16–18]. The relationship between traditional investment strategies is shown in Figure 1.

3. LSTM NN

The concept of LSTM NN was first proposed in 1997. Unlike the basic RNN, the LSTM NN has stronger memorization,

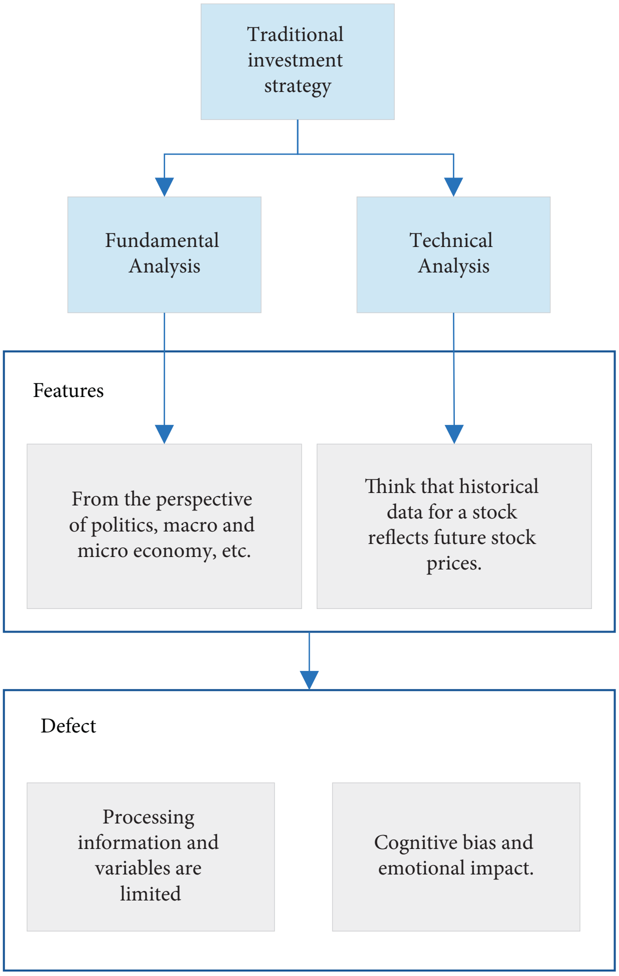

Traditional investment strategy
Fundamental Analysis
Technical Analysis
Features
From the perspective of politics, macro and micro economy, etc.
Think that historical data for a stock reflects future stock prices.
Defect
Processing information and variables are limited
Cognitive bias and emotional impact.

Figure 1: Traditional investment strategy.

can express the implicit relationship between sequences faster, more accurately, and better, and is better at processing long sequence signal data. Since its proposal, LSTM has been extensively applied to predict sequences and natural language processing (NLP) in place of the basic RNN model [19]. Furthermore, the LSTM NN introduces two variables: the output and the memory, along with a gating mechanism to control the information flow. When the information is input into the LSTM NN, the information that meets the requirements is left, and the information that does not meet the criteria is forgotten [20, 21]. The gating mechanism of LSTM NN is depicted in Figure 2.

The LSTM NN has two state vectors: C and h, in which C is the internal state vector or the memory state vector and h represents the output vector. Figure 3 projects the LSTM NN structure.

In Figure 3, the output vector h_{t-1} of the previous timestamp and the input x_t of the current timestamp pass through the AF tanh to obtain a new output vector h_t. Compared with the basic RNN, LSTM only has one output vector (h) and adds a state vector C_t. At the same time, the gating unit is used to control the forgetting and refreshing of information. The AF tanh is calculated by the following equation:

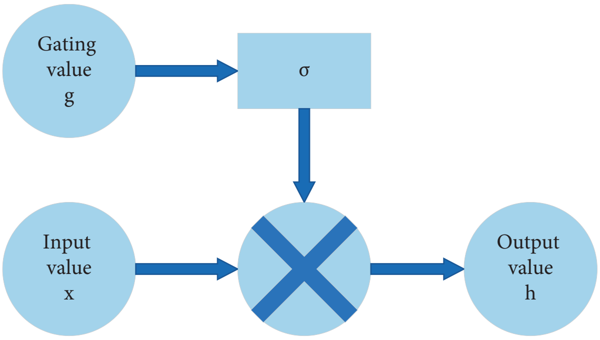


Figure 2: LSTM NN gating mechanism.

$$f(x)_{\tanh} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{1}$$

The input gate can control the LSTM NN input. Firstly, the input x at the current time and the output h_{t-1} at the previous time are transformed nonlinearly. The input vector is normalized to the interval of -1 and 1 through tanh AF. Yet, the input vector will not all be memorized by the LSTM NN but is controlled by the input gate. The control vector of the input gate comes from the nonlinear transformation of the input x_t and the output h_{t-1} of the previous timestamp through the AF, generally a sigmoid function [22]. The input gate control vector determines the acceptance degree of the LSTM NN to the new input vector of the current timestamp. When the control vector is 0, the LSTM NN does not accept any new input. In comparison, all new inputs are accepted when the control vector is 1. Controlled by the forget gate and the input gate, the LSTM NN selectively reads the memory of the previous timestamp and the new input of the current timestamp, forming a certain combination of these state vectors and, thereby determining the current timestamp's state vector. Figure 4 shows the specific process.

Unlike basic RNN, LSTM NN will not directly output the internal state vector. The state vector h of the basic RNN is both memory and output, so the state vector c equals the output vector h. By contrast, in the LSTM NN, the output gate controls the state vectors to be output. The output gate vector results from the offline combination of the state vector h_{t-1} of the previous timestamp and the input vector x_t through the AF transformation, generally a sigmoid function. When the output gate vector is 0, the output is closed, and the internal memory of the LSTM NN is completely blocked and cannot be used as the output. At this time, the vector 0 is output. Then, when the output gate vector is 1, the output is fully opened, and all the state vectors of the LSTM NN are used for output. The output of the LSTM NN comes from the multiplication of the output gate vector and the nonlinear transformation result of the state vector c_t through the tanh AF. In other words, the memory vector acts with the output gate after tanh AF to obtain the output of LSTM. Since the output gate vector falls into 0 and 1 and the tanh (c_t) falls into -1 and 1, then the output of the LSTM NN falls into -1 and 1 [23]. Figure 5 specifies the process of LSTM output.

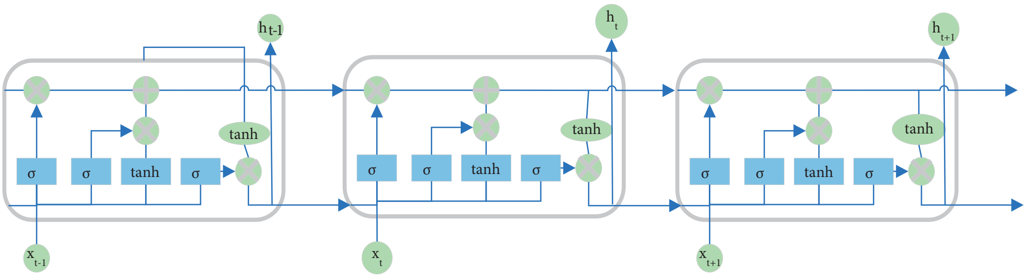


Figure 3: LSTM NN structure.

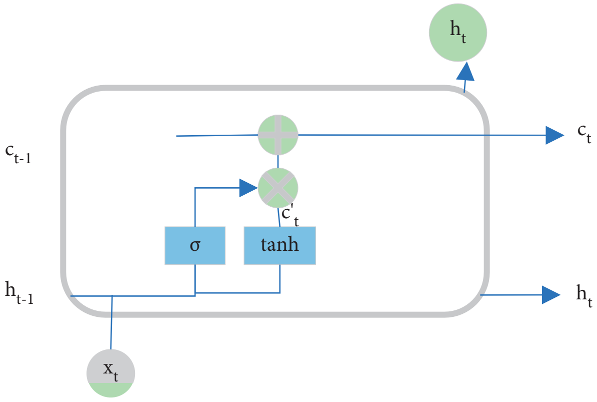


Figure 4: Schematic diagram of the input gate.

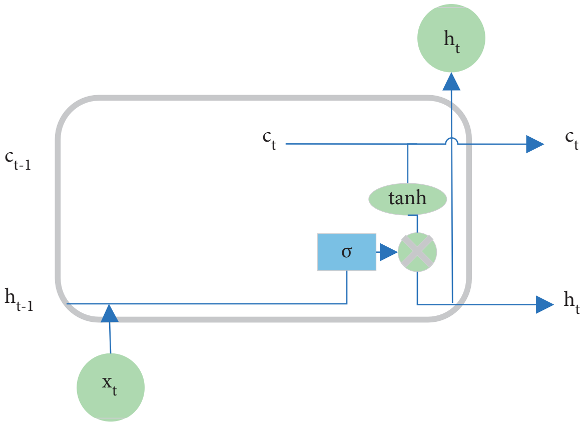


Figure 5: Schematic diagram of the output gate.

There are many state vectors and gate vectors in LSTM NNs, several nonlinear AFs are also used internally, and the calculation process is relatively complex. However, the function of each gate control is clear, and the typical gate control behavior is illuminated in Figure 6.

4. Financial SIM Model Based on DL

For the whole transaction model, the network structure of the model is essential because it determines the effect and convergence speed of model training. According to the time-series features of financial data, this paper hopes to design the model with two functionalities: one is to predict the future state according to historical information, and the other is to decide what action to take. When the LSTM NN is used to implement the financial SIM model, the input includes the observation of the market at each time and the change of transaction information over time. The LSTM NN can find and learn the potential relationship between time series data (TSD). Moreover, given the hardware conditions of the machine used in the training model, if the search space is too large, the multilayer dense NN may be too bulky to train, while the structure and weight sharing features of the LSTM NN reduce the difficulty of training [24, 25]. Accordingly, the proposed financial SIM model is based on the LSTM NN, including two hidden layers and a single output layer. Specifically, the hidden layer is the LSTM and the output layer is the fully connected NN.

Meanwhile, total return should be maximized while the trading strategy needs to consider the trading risk (mainly, the standard deviation (SD) of return). Therefore, to realize multiobjective trading strategy learning, this section first sets certain weights for different objectives and linearly fuses all objectives according to the set weights. Then, based on this objective training model, the investment strategy is finally output.

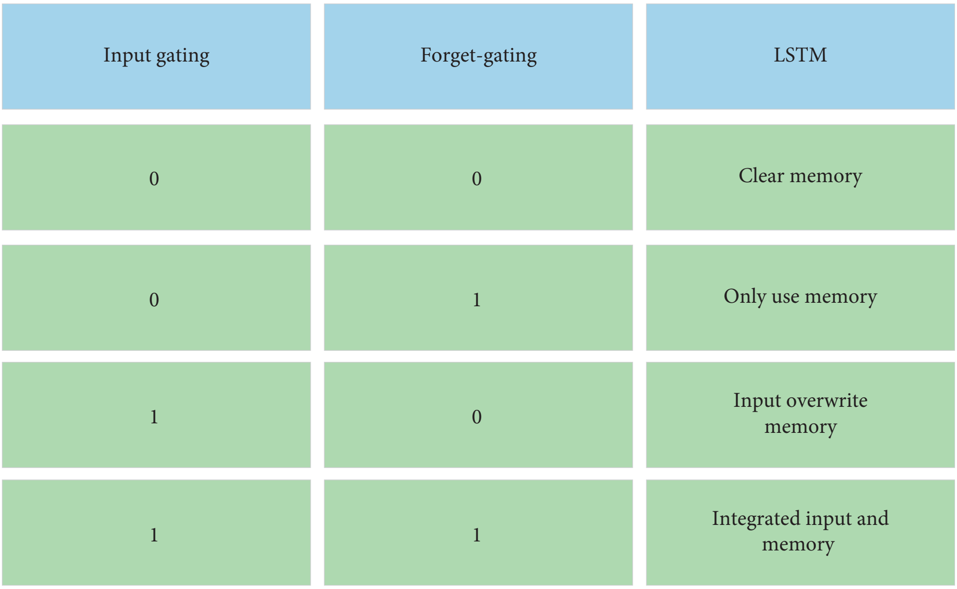
Subsequently, the SD of daily return is used to measure the fluctuation of daily return. The first OF of the trading model is the average daily return R, and the return at the current time is equal to the stock price variation-induced capital gain minus the transaction handling fee [26]. The second OF is the SD of average daily return σ. Then, based on the VC utility function $U = E(r) - 0.5\,A\sigma^2$, the two OFs are linearly combined to obtain the total OF (U). Equations (2)–(4) calculate the multi-OF.

$$R = \frac{(1-c)\left(\sum_{t=1}^{T} R_t \sum_{i=1}^{n} R_i \mid \theta\right)}{T}, \tag{2}$$

$$\sigma = \operatorname{std}\left(R_i|\theta\right), \quad i = 1, 2 \ldots T, \tag{3}$$

$$U = R - A\sigma^2. \tag{4}$$

In equations (2)–(4), T indicates the model's accumulative transaction days, n signifies the single-day transactions, θ is the model parameter, and c denotes the transaction fee ratio, including stamp duty. Meanwhile, A means the weight corresponding to the fluctuation OF, and the total OF can be adjusted through the weight. Here, A is set to 0.2. Because the U calculated by the above equations is



Input gating	Forget-gating	LSTM
0	0	Clear memory
0	1	Only use memory
1	0	Input overwrite memory
1	1	Integrated input and memory

Figure 6: Gate behavior of the LSTM NN.

large, the network input cannot be ideal. Therefore, the total OF should be normalized. Simply put, the mean of all U is subtracted from each U, then divided by the SD of U, and finally, input into model training [27–29].

The action space of reinforcement learning (RL) is the collection of all effective actions of the model, determining the range of action settings. The present work considers the buying and selling of stock in the stock market, thus only involving a discrete action space, including actions such as buying, selling, and wait-and-seeing. The proposed model uses a fully connected NN as the output layer to output a one-dimensional (1D) tensor containing three elements. First, the model uses the program to generate a random number subject to 0 and 1 uniform distributions. Suppose the random number is less than the preset value. In that case, the index corresponding to the largest element is selected as the action value among the three elements of the output vector. Otherwise, an integer in 0 and 2 is randomly selected as the action value. By referring to relevant codes and combined with multiple tests, the present experiment sets the preset value to 0.85. Action value = 1 indicates a stock-buying action. Similarly, action value = 2 suggests a stock-selling action. Finally, action value = 0 means a hold-and-waiting action.

Next, the proposed model is trained in the following ways: the LSTM NN is used to approximate the state function, namely, to estimate the Q value. The model input is the current time state f_t, and the network outputs the estimated value Q(s and a) of all actions in the action space. Figure 7 plots the specific network structure of approaching the Q value

In Figure 7, the Q value is estimated at each time, and the Mean Square Error (MSE) is taken as the loss function of the training model [30]. Then, the following equation calculates the MSE.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\left|y_i - \hat{y}_i\right|\right)^2. \tag{5}$$

In Figure 7, the real-time Q value is estimated to be $Q(s, a)$, and the target Q value is expressed by $r + \gamma \max Q(s', a')$. Differently put, the target Q value equals the expected Q value after the state is transferred to s$'$+ the current time environmental reward r. γ is the attenuation factor $\in [0, 1]$. Afterward, the loss function should be minimized to maximize the Q value. After the gradient of the loss function is obtained, the parameters are updated iteratively. According to the GDM, the model loss function and parameters are renewed by equations (6)–(8):

$$Q^*(s,a) \longleftarrow Q(s,a) + \alpha\{r + \gamma \mathrm{max} Q(s',a') - Q(s,a)\}, \tag{6}$$

$$L(s,\ a|\theta_i) = \left(r + \gamma \mathrm{max} Q(s',a'|\theta_i) - Q(s,a)|\theta_i\right)^2, \tag{7}$$

$$\theta_{i+1} = \theta_i + \alpha \nabla_\theta L(\theta_i). \tag{8}$$

In equations (6)–(8), a is the learning rate. When $Q \approx Q^*$, $\pi_Q(s) = \mathrm{arg}_a \mathrm{max} Q(s,a)$ can approximately represent the optimal strategy. According to equation (6), the predicted Q value and the target Q value use the same parameter model. When the predicted Q value increases, the target Q value also increases, which increases the risk of model fluctuation and divergence to a certain extent. Hence, to solve this problem, the DQN algorithm employs two different learning networks: the prediction network can evaluate the cost function of the current state-action pair, and the target network can generate the target Q value. Meanwhile, the loss function can update the parameters of the prediction network model. After multiple iterations, the parameters in the prediction network are copied to the parameters in the target network. Significantly, DQN introduces the target network to stabilize

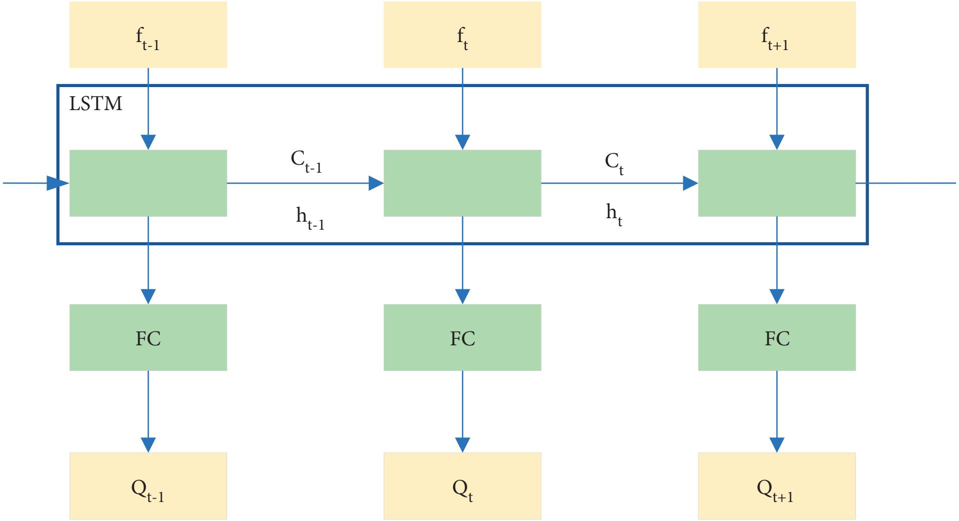


Figure 7: Transaction network structure based on LSTM.

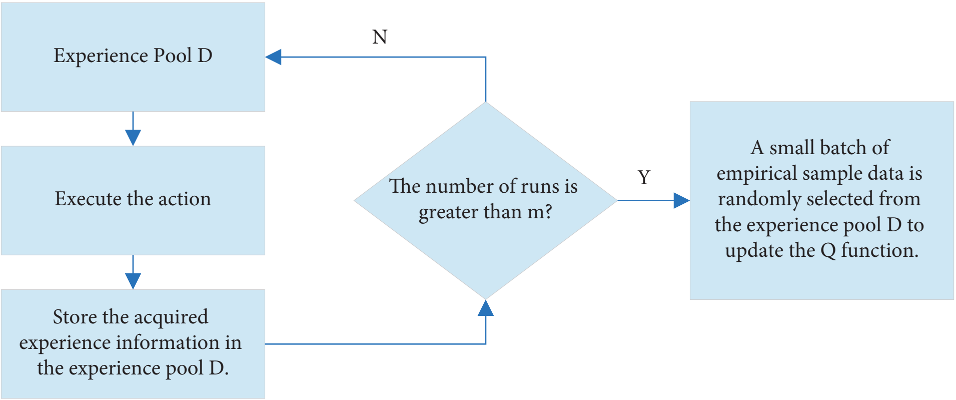

Experience Pool D
Execute the action
Store the acquired experience information in the experience pool D.
N
The number of runs is greater than m?
Y
A small batch of empirical sample data is randomly selected from the experience pool D to update the Q function.

Figure 8: Experience playback mechanism.

the target Q value for some time, reduces the correlation between the predicted Q value and the target Q value, and improves the algorithm's stability.

The following experiment will use the ER mechanism for the DQN algorithm. First, it pools the experience sample data obtained through the agent-environment interaction at each time step (namely, the experience pool). Then, it randomly selects a small batch of data from the experience pool to train the network. DQN algorithm saves a certain amount of historical experience sample data. Each experience sample is stored in the form of (s, a, r, s', and T) quintuple (indicating that the agent performs the action (a) in the state (s) and reaches the new state (s') and obtains the corresponding reward r. Meanwhile, T is a Boolean value indicating whether the new state (s') is terminated). Thus, the ER mechanism effectively removes the sample correlation and dependence so that the DNN model can better learn the cost function in RL tasks. The principle of the experience playback mechanism is given in Figure 8.

The specific process is as follows:

Step 1: initialize experience pool D with a capacity of n to store training samples.

Step 2: the state function Q is set as the prediction network, and the weight parameter θ is initialized randomly.

Step 3: set the state value function Q' as the target network and initialize the weight parameters θ' to θ.

Step 4: set the maximum running times of the model as m.

Step 5: randomly select the action a_t or calculate the action Q-value corresponding to the current state according to the network. Select the action with the largest Q-value as the optimal action.

Step 6: the model performs the action (a_t) to obtain the reward signal (r_t) and state (s) from environmental feedback.

Step 7: store the obtained state transition parameters into experience pool D.

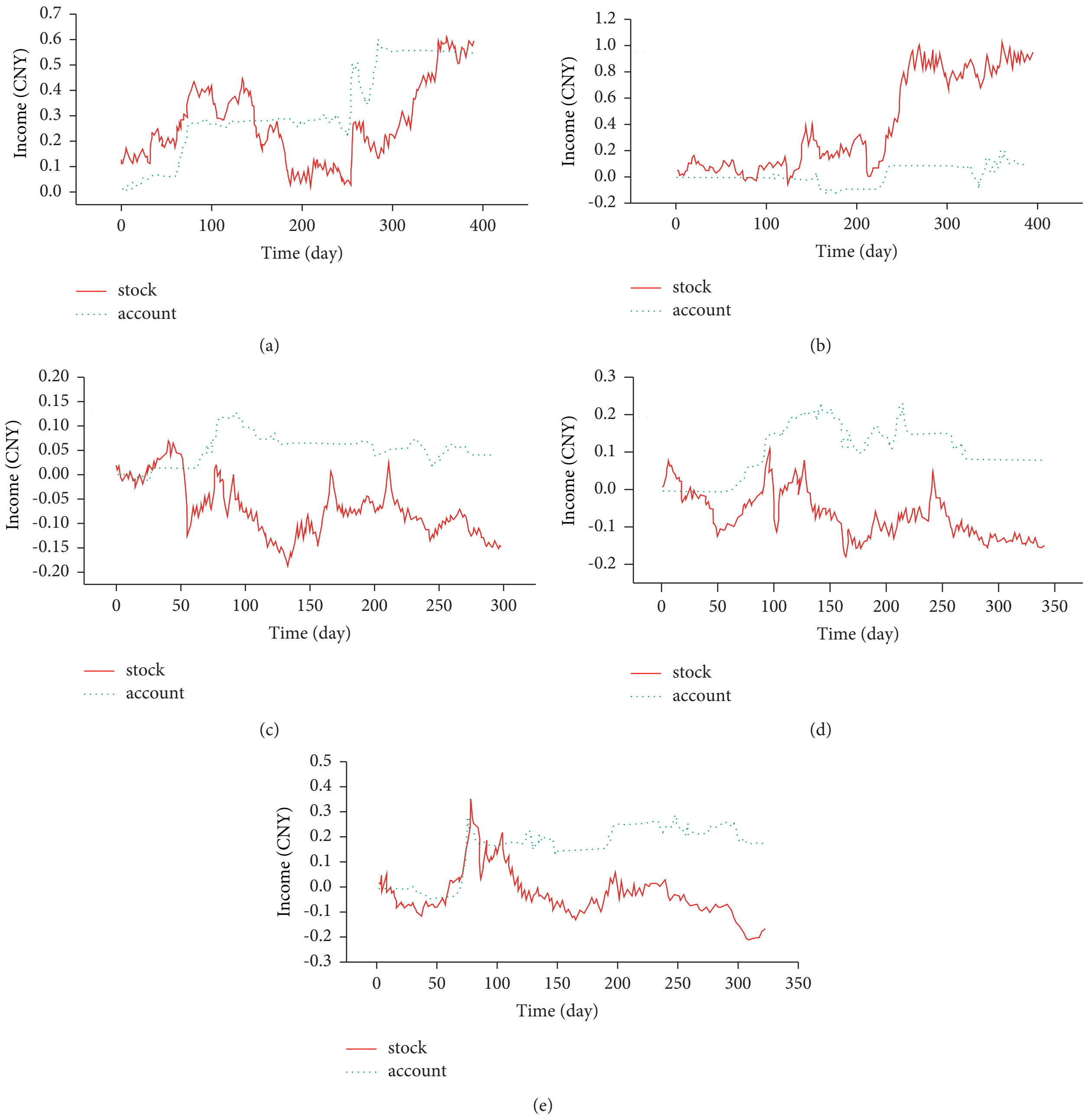


Figure 9: Trading behavior and account income of the (a-e) stock model.

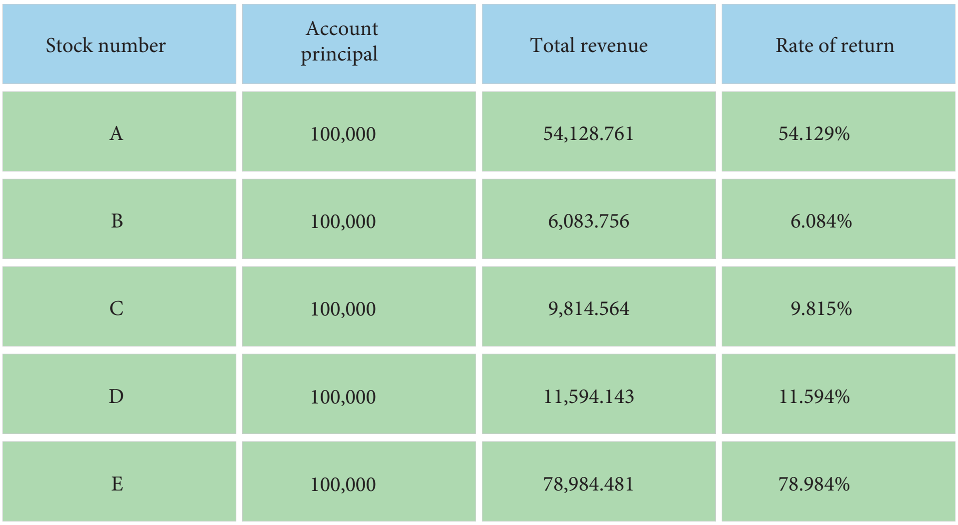
Step 8: the model randomly takes out the relevant information of calibration batch status from experience pool D.

Step 9: calculate the target value of each state, and the model updates the Q value through the reward r after the target network Q' performs the action.

Step 10: update the weight parameter θ of the Q network based on small-batch samples using equations (6)–(8).

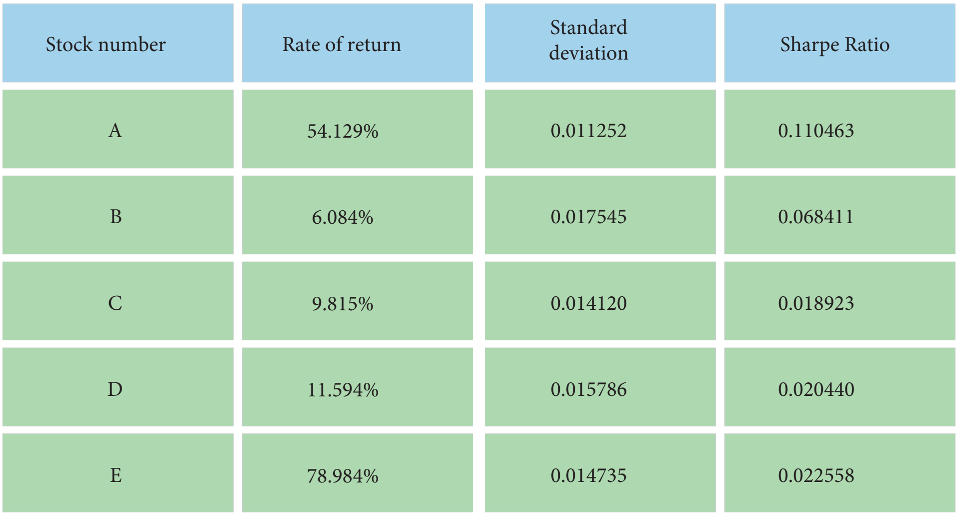
Step 11: every time, after c iterations, update the network parameter θ' of the target action cost function Q' as a parameter θ of the prediction network.

This section selects five stocks from the current Chinese A-shares, numbered A-E, to form an asset pool. Then, the initial 500,000 amount of the account is divided into five equal shares, and five stocks are invested and traded, respectively. Each stock is trained by indexes, including the opening price (open), closing price (close), highest price (high), lowest price (low), transaction volume (volume), MA10, MA21, RSI, momentum (MOM), and exponential moving average (EMA). Then, the open, close, high, and low are obtained through the Tushare financial data interface, while MA10, MA21, RSI, MOM, and EMA are obtained by calling TA-lib.



Stock number	Account principal	Total revenue	Rate of return
A	100,000	54,128.761	54.129%
B	100,000	6,083.756	6.084%
C	100,000	9,814.564	9.815%
D	100,000	11,594.143	11.594%
E	100,000	78,984.481	78.984%

Figure 10: Model back-test results.



Stock number	Rate of return	Standard deviation	Sharpe Ratio
A	54.129%	0.011252	0.110463
B	6.084%	0.017545	0.068411
C	9.815%	0.014120	0.018923
D	11.594%	0.015786	0.020440
E	78.984%	0.014735	0.022558

Figure 11: Comparison of SD and SR of each stock.

5. Experimental Results and Analysis

5.1. Experimental Result. Next, the five stocks' data are divided into two parts: the data from January 1, 2012, to May 29, 2019 (used for model training), and the data from May 30, 2019, to December 31, 2021 (used for back-test inspection). Figure 9 charts the inspection results.

Figures 9(a)–9(e) display the trading behavior and account income of the A-E stock model, respectively.

In Figure 9, the abscissa and the ordinate are the trading time and stock price increase. The red line indicates the stock price trend during the back-test, while the green line represents the trend of proxy account value during the back-test. The specific back-test results of each stock are summarized in Figure 10

In Figure 10, the total return is the account's market value on the last day of the back-test period minus the initial principal. The rate of return (RoR) is the percentage of the quotient of the total return and the account's principal. Overall, the RoR of the agent account during the back-test period is 32.12%, and the performance is relatively ideal. The Shanghai Stock Index (SSI) increase in the same period is 19.157%. Thus, the model's performance has exceeded the increase of the SSI in the same period. The five stocks all have achieved positive returns during the back-test period in terms of individual stocks. E stock achieved the highest yield, 78.984%, while the B stock has achieved the lowest, 6.084%.

5.2. Index Analysis. The back-test results are further analyzed by sharp ratio (SR). Specifically, the calculation of SR takes each trading day as the cycle. Generally speaking, the smaller the cycle is, the higher the calculation frequency is, and the greater the return volatility is. Since it is more difficult to make profits every trading day than every year, the calculation method with high calculation frequency can more comprehensively and genuinely reflect the risk-return status of account investment. The SD and SR of each stock are illuminated in Figure 11.

As corroborated by Figure 11, the SR of the five stocks in the account is positive, indicating that the average NGR of the stocks in the measurement period exceeded the risk-free interest rate. The larger the SR is, the higher the risk return is. Therefore, the proposed financial SIM model based on the DL algorithm is scientific and feasible.

6. Conclusions

At present, most trading strategies combined with DL focus on portfolio screening and rise and fall prediction. However, there are still few studies on reducing trading risk effectively. This paper mainly studies the financial SIM based on DL and obtains the excess return by exploring a new model. Firstly, the DQN algorithm is integrated into financial SIM, and then the model based on LSTM is constructed. Secondly, the model is trained by historical data, the training results are normalized, and GDM optimizes the model. Finally, the trained model is tested. As a result, five stocks of Chinese A-shares are selected to form an asset pool. The initial 500,000 amount of the account is divided into five equal shares, and the five stocks are invested and traded, respectively. The test results corroborate that the overall RoR of the model account during the back-test is 32.12%, and the maximum RoR falls on E stock with 78.984%. The minimum RoR goes to B stock with 6.084%, which is a positive return, and the SR of each stock is positive. Therefore, the proposed financial SIM model based on the DL algorithm is scientific and feasible. Regarding research limitations, this paper only used the basic stock market data and derived technical indicators without considering other factors. The lack of such data might determine the upper limit of the model. Later, other features affecting the stock price will be added to the data set to enhance the generalization and robustness of the model. The research content has certain reference significance for the DL-based financial SIM.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. C. Huang, “Assessing the financial performance of airlines in the Asia-Pacific region,” *Investment Management and Financial Innovations*, vol. 18, no. 2, pp. 234–244, 2021.

[2] S. Prime, “Forecasting the changes in daily stock prices in Shanghai stock exchange using neural network and ordinary Least Squares regression,” *Investment Management and Financial Innovations*, vol. 17, no. 3, pp. 292–307, 2020.

[3] Y. Rongjun, Y. Lin, Z. Yuanjun et al., “Big data analytics for financial Market volatility forecast based on support vector machine - ScienceDirect,” *International Journal of Information Management*, vol. 50, pp. 452–462, 2020.

[4] H. Guo, M. Sandra, R. Savickas, and R. Wood, “Market liquidityand conditional equit,” *Financial Management*, vol. 46, no. 3, pp. 743–766, 2017.

[5] R. Arnott and C. Harvey, “Backtesting protocol in the era of machine learning,” *SSRN Electronic Journal*, vol. 14, no. 2, pp. 23–29, 2018.

[6] J. Simonian and C. Wu, “New foundations for quant-macro investing,” *The Journal of Financial Data Science*, vol. 1, no. 2, pp. 94–110, 2019.

[7] V. Tripathi and P. Singh, “Fama-French five-factor asset pricing model: empirical evidence from Indian stock market,” *International Journal of Business and Globalisation*, vol. 27, no. 1, pp. 70–91, 2021.

[8] S. A. Bhutto, R. R. Ahmed, D. Streimikiene, S. Shaikh, and J. Streimikis, “Portfolio investment diversification at global stock market: a cointegration analysis of emerging brics(P) group,” *Acta Montanistica Slovaca*, vol. 25, no. 1, pp. 57–69, 2020.

[9] G. Andreas and U. N. Nurjannah, “The effect of announcement as the host of XVIII Asian Games on the Indonesian stock market,” *Investment Management and Financial Innovations*, vol. 17, no. 1, pp. 109–118, 2020.

[10] A. Bhandari and K. Bandi, “Time-varying nature of stock market interdependence a global perspective,” *Economic and Political Weekly*, vol. 55, no. 13, pp. 62–71, 2020.

[11] A. V. Koren, T. E. Danilovskikh, and A. A. Pustovarov, “Improvement of tax regulation arrangements for the Russian stock market,” *REICE: Revista Electrónica Iberoamericana sobre Calidad, Eficacia y Cambio en Educación: Revista Electrónica de Investigación en Ciencias Económicas*, vol. 8, no. 16, pp. 556–570, 2020.

[12] J. Hu, Y. Sui, and F. Ma, “The measurement method of investor sentiment and its relationship with stock market,” *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 6672677, 11 pages, 2021.

[13] Q. Yang and X. Wang, “Simulation of stock market investor behavior based on bayesian learning and complex network,” *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 2, pp. 2481–2491, 2021.

[14] A. Emamian and N. S. Mazlan, “Monetary-Fiscal policies and stock market performance: evidence from linear ARDL framework,” *GATR Journal of Business and Economics Review*, vol. 5, no. 4, pp. 69–80, 2021.

[15] Y. Antoniuk and T. Leirvik, “Climate change events and stock market returns,” *Journal of Sustainable Finance & Investment*, vol. 1, pp. 1–26, 2021.

[16] M. Kashif, S. Chamadia, R. R. Ahmed, O. A. Kalugina, and V. Havrysh, “Identifying investment strategies using accounting information from stock market,” *Transformations in Business and Economics*, vol. 3, no. 51, pp. 235–260, 2020.

[17] S. Thakolsri, “Modeling the relationships among gold price, oil price, foreign exchange, and the stock market index in Thailand,” *Investment Management and Financial Innovations*, vol. 18, no. 2, pp. 261–272, 2021.

[18] S. Banerjee, N. Dabeeru, and R. Lavanya, “Stock market prediction,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 9, pp. 506–510, 2020.

[19] J. Shu, D. Zhao, X. Zheng, Y. Li, and Y. Zhang, “Bridge temperature prediction model based on long-short term memory neural network,” *Journal of Physics: Conference Series*, vol. 1966, no. 1, p. 6, 2021.

[20] A. Altan, S. Karasu, and E. Zio, "A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer," *Applied Soft Computing*, vol. 100, no. 4, Article ID 106996, 2021.

[21] F. Knig, J. Marheineke, G. Jacobs, C. Sous, and Z. Tian, "Data-driven wear monitoring for sliding bearings using acoustic emission signals and long short-term memory neural networks," *Wear*, vol. 476, no. 3, Article ID 203616, 2021.

[22] J. Sun and J. Kim, "Joint prediction of next location and travel time from urban vehicle trajectories using long short-term memory neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 128, no. 5, Article ID 103114, 2021.

[23] H. Su, T. Zhang, M. Lin, W. Lu, and X. H. Yan, "Predicting subsurface thermohaline structure from remote sensing data based on long short-term memory neural networks," *Remote Sensing of Environment*, vol. 260, no. 8, Article ID 112465, 2021.

[24] L. Ma, C. Hu, and F. Cheng, "State of charge and state of energy estimation for lithium-ion batteries based on a long short-term memory neural network," *Journal of Energy Storage*, vol. 37, Article ID 102440, 2021.

[25] J. Liu, C. Ma, H. Gui, and S. Wang, "Thermally-induced error compensation of spindle system based on long short term memory neural networks," *Applied Soft Computing*, vol. 102, no. 2, Article ID 107094, 2021.

[26] R. Bhardwaj and A. Bangia, "Assessment of stock prices variation using intelligent machine learning techniques for the prediction of BSE," *Numerical Optimization in Engineering and Sciences*, vol. 979, pp. 159–166, Springer, Berlin Germany, 2020.

[27] O. B. Sezer, U. Gudelek, and M. Ozbayoglu, "Financial time series forecasting with deep learning: a systematic literature review: 2005-2019," *Applied Soft Computing*, vol. 90, Article ID 106181, 2020.

[28] J. Zhou, M. Yang, F. Wang, and C. Wang, "Research on financial aid model of colleges and universities based on artificial neural network," *Journal of Physics: Conference Series*, vol. 1575, no. 1, Article ID 012100, 2020.

[29] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," *Neural Computing & Applications*, vol. 32, no. 6, pp. 1609–1628, 2020.

[30] X. Yan, W. Weihan, and M. Chang, "Research on financial assets transaction prediction model based on LSTM neural network," *Neural Computing & Applications*, vol. 33, no. 1, pp. 257–270, 2021.