IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 June 2023, accepted 20 July 2023, date of publication 26 July 2023, date of current version 1 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298821

METHODS

Multi-Feature Supervised Reinforcement Learning for Stock Trading

KUI FU, YIDONG YU, AND BING LI
School of Economics, Wuhan University of Technology, Wuhan, Hubei 430070, China
Corresponding author: Kui Fu (fukui@whut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 72004174.

ABSTRACT Deep reinforcement learning (DRL) algorithm is often used to find the best trading strategy in algorithmic trading. However, the classical DRL model is difficult to achieve rapid convergence, and the features extracted from the market data are relatively simple, resulting in incomplete DRL learning information. In this paper, we propose a supervised reinforcement learning method, a hybrid optimal investment strategy formation method consisting of long short-term memory neural network (LSTM) and deep deterministic policy gradient (DDPG). By participating in reinforcement learning in the early stage of supervised learning, agents can obtain guiding prior experience, thus reducing the cost of agent learning and accelerating convergence. In addition, multi-feature state input is added to the model to optimize the agent's learning of the environment. Compared with DDPG algorithm, LSTM-DDPG algorithm achieves convergence faster. Experiments on three regional stock markets in China, the United States and Europe show that LSTM-DDPG algorithm has higher profit and lower risk than B&H, MACD and LSTM trading strategies.

INDEX TERMS Supervised reinforcement learning, finance and operations, reinforcement learning, deep deterministic policy gradient, long short-term memory.

I. INTRODUCTION

The stock market plays an important role today. It is a direct reflection of the financial status of a company or even a country and reflects the economic health and development prospects of a company or a country. As a high-yield wealth management product, people expect to make profits from stock trading. However, the stock market is a typical nonlinear complex system, the uncertainty of the stock price trend and traders' emotions are the factors that affect the stock market trading [1], [2].

Using an electronic platform, we can enter trading orders containing algorithms to execute pre-defined trading strategies. This trading process is called algorithmic trading (AT). AT removes human emotions as well as the time required to make trading decisions and execute actual trades [3], [4]. AT can connect traders and the market organically to reduce the friction between them and reduce the impact of trading on the market to a certain extent.

In stock trading, a trading strategy is a rule that determines when to sell, buy, or hold a stock. For example, Pairs trading [5]. With a trading strategy, you can control the frequency of trading and not fall into the trap of trading too much and losing yourself. If the trading strategy can be given more accurately, it will have a strong reference significance and value for both the company and individual shareholders. The goal of algorithmic trading is to balance the impact cost and time risk, by taking the two as the objective to optimize, the result is the optimal strategy.

In the financial markets, investors usually want to know the return on their investment, the future trend of the stock market, and the optimal trading strategy to choose during the investment process. Therefore, stock trading decision model has always been an important research object in the field of financial investment. In recent years, researchers have begun to apply deep reinforcement learning-related technologies to the field of financial investment. RL is an adaptive model,

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

and dynamic self-improving trading strategies can be developed by using RL [6], [7], [8]. Some literatures have studied the application of RL methods in stock trading. Li et al. [9] proposed an RL scheme for short-term stock price movement prediction based on the actor-critic and critics-only RL methods respectively. Jia et al. [10] used LSTM neural network to extract market state characteristics and DDPG framework to judge trading decisions, and thus proposed a deep reinforcement learning model named LSTM-DDPG to make trading decisions. Li et al. [11] proposed PPO enhancement strategy to modify the signal of stock trading strategy rather than directly predict the direction of stock price, and the experiment proved that the proposed PPO enhancement strategy was superior to the benchmark test. In many applied research of stock trading, more and more scholars have found the superiority of RL. Li and Liao [12] effectively overcomes the limitations of supervised learning methods by combining the ''prediction'' step of setting prices and the ''allocation'' step of portfolios in a unified process through the DRL model to generate profitable trades in the stock market. Demonstrated the advantages of DRL in the financial market over other types of machine learning. Tang [13] proposed a stock market trading model based on deep reinforcement learning, which is suitable for predicting volatility of stock trading prices and stock trading. Experiments show that the model is superior to SVM and other supervised learning methods in several indicators. Watkins et al. [14] extended the algorithms of Deep Q Network and Asynchronous Advantage Actor Critic to better adapt to the trading market. Experimental results show that the proposed trading agent is better than CNN, DNN, SVM and other supervised learning algorithms in the stock market. And achieved a stable risk-adjusted return. Compared with supervised learning, reinforcement learning has two advantages in stock trading: defining the model requires fewer constraints, which reduces the difficulty of defining the transaction problem; The overall return from a sequence of trading actions is more important than the consistency of a single move. The deep reinforcement learning algorithm model can make use of repeated learning in real-time financial trading scenarios, optimize its own model to adjust network parameters, seize fleeting trading opportunities, and quickly make buying and selling decisions.

Through the above-mentioned literature research, we find that the stock market information has a lot of noise and uncertainty [15], [16], [17], and the stock trading model based on RL is difficult to converge due to the large space of stock trading decision action, and the quality of features extracted in the stock trading model will directly affect the performance of the learned trading strategy. Good trading systems rely on reliable input characteristics, and finally, due to the lack of prior experience, the stock trading model based on RL performs poorly in the early training phase. To overcome these limitations, we propose a new stock trading model that considers input characteristics, convergence, and pre-training performance.

Based on the existing analysis and techniques of financial markets, this paper aims to establish a quantitative trading algorithm combining supervised learning and reinforcement learning techniques. This research mainly focuses on seeking an optimal trading strategy and stock trading model training from numerous financial historical data, improving training performance and reducing trading costs and risks for investors. In this regard, our study has two main contributions, which fill gaps in the existing literature. First, we consider multi-category features as state inputs for training and provide a multi-feature set about stock trading. In addition to frequently used stock trading data and technical indicators, features formed according to wave theory and inverted K-line form are also added, to explore the regularity of stock price fluctuations and stock trend reversal and explore the deep market rules of the stock market. Thus, it can enhance the agent's exploration ability and obtain better trading strategies. Secondly, we propose a supervised reinforcement learning algorithm and apply it to stock trading. This method combines supervisor supervision and evaluation with active exploration of reinforcement learning. First, it ''simulates'' the reaction behavior in the supervised data to obtain prior experience, thereby reducing the difficulty of the Agent in the early learning and exploration process as the guidance of the teacher. Second, it continuously entices the Agent's experience through active exploration of reinforcement learning. Finally, the optimal control and fast convergence of the system are realized. In this study, we redesigned the action update strategy to achieve the above purpose, that is, the action decision should consider the guidance of supervised learning and the exploration of reinforcement learning. In this study, we used LSTM neural network as the action acquisition method of supervised learning and DDPG algorithm [18], [19] as the action acquisition method of reinforcement learning.

The rest of the paper is organized as follows. Section II introduces the background research of LSTM neural networks and RL. Section III illustrates the model for predicting the coached actions in stock trading and the framework of the deep reinforcement learning algorithm for stock trading. Finally, we discuss our proposed trading decision method of supervised reinforcement learning. Section IV discusses the experimental results. Section VI describes the conclusions and prospects.

II. METHODOLOGY

A. LONG SHORT-TERM MEMORY

Long Short-Term Memory artificial neural network (LSTM) has been applied in various fields and disciplines. It is a kind of time-cycle neural network, which can effectively preserve long-term memory through a unique ''gate'' structure and cell state update. It has the characteristics of parameter sharing, so it is suitable for dealing with the problem of long-cycle time series prediction, and the prediction speed is fast, and the accuracy is high. Therefore, LSTM forecasting method

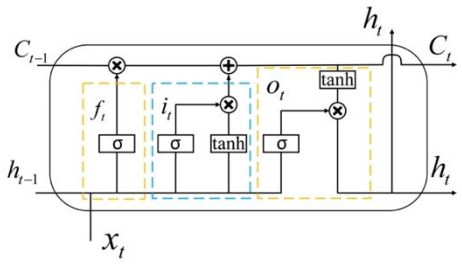


FIGURE 1. The architecture of LSTM. Picture inspired by ⟨⟨Deep Learning for Natural Language Processing⟩⟩ (Li Deng, Yang Liu).

has been widely used in many fields such as weather forecasting, stock forecasting, and behavior forecasting [20], [21]. In the aspect of stock forecasting, many scholars have confirmed the superiority of LSTM. Nabipour et al. [22] selected four stock market groups of the Tehran Stock Exchange for an experimental valuation. Nine machine learning models (decision trees, random forests, Adaptive enhancement (Adaboost), Limit gradient enhancement (XGBoost), support vector classifiers (SVC), Bayes, K-nearest Neighbor (KNN), Logistic regression and artificial neural networks (ANN) and two powerful deep learning methods (Recursion God) are compared by network (RNN) and Long Short-term memory (LSTM). The evaluation results show that LSTM is superior to other prediction models for continuous data. Ji et al. [23] studied various most advanced deep learning methods, such as deep neural network, convolutional neural network, and deep residual network, in bitcoin price prediction, and the experimental results showed that the prediction model based on LSTM was slightly better than other models in bitcoin price prediction (regression). Wang et al. [24] established BP neural network model and LSTM model to predict stock prices. Experimental comparison shows that LSTM model is more accurate.

The LSTM structure is shown in Fig. 1, and these gate structures update the cell state C_t by the following recursive equation.

$$\begin{cases} f_t = \sigma\left(W_f \cdot [h_{t-1}, X_t] + b_f\right) \\ i_t = \sigma\left(W_i \cdot [h_{t-1}, X_t] + b_i\right) \\ o_t = \sigma\left(W_o \cdot [h_{t-1}, X_t] + b_o\right) \\ \tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, X_t] + b_C\right) \\ C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \\ h_t = o_t * \tanh\left(C_t\right) \end{cases} \tag{1}$$

The forgetting gate f_t determines what information we will discard from the cell state. The input gate i_t determines how much new information to let into the cell state. The output gate o_t finally outputs the state of the neural unit. σ represents the activation function, W and b represent the weight matrix and the bias. h_{t-1} represents the output of the LSTM cell at time $t-1$, and X_t represents the input at time t.

B. MARKOV DECISION PROCESS

Markov Decision Process (MDP) [25] is used to define a sequential decision problem with uncertainty. It defines a problem that learns by interacting with the environment to find the goal state. All states in MDP follow the Markov property, that is, the next state s_{t+1} depends only on the current state s_t [19]. Stock trading can be viewed as a Markov decision process, where an agent interacts with the financial market by buying and selling stocks and receives the corresponding return from the market, denoted $M = (S, A, R, Z, \gamma)$, where S is the state space and a set of states can describe the environment [3]. A is the action space, where an agent can react to the environment at any moment with a set of actions. Z is the probability matrix of state transitions, $\gamma \in (0, 1)$ is the discount factor, and R is the direct reward.

When we give a state $s_t \in S$, the agent will take an action $a_t \in A$ according to the policy $\pi(a \mid s) = P(a \mid s)$, and the environment will reach a new state $s_{t+1} \in S$. At the same time, and the agent will get the corresponding reward R_t.

C. REINFOECEMENT LEARNING

In a Markov decision process, a policy is a state-to-action mapping function, it selects the optimal action for a state and uses the environment to update the reward of the action taken by the state. The optimal action is the one that provides the maximum reward among all possible actions. The goal of reinforcement learning is to obtain the optimal strategy and find the strategy that can get the maximum cumulative reward [26], [27].

As shown in Fig. 2, in the process of reinforcement learning, the Agent constantly interacts with the Environment. The agent obtains the current state from the environment and takes an action according to the current state. This action will lead to changes in the state of the environment, and the environment will give the agent a reward. The agent will continue to select actions according to the principle of maximizing the cumulative reward value, and finally achieve the goal [28], [29].

When an agent interacts with its environment, it obtains a continuous sequence of states and actions, which is called a trajectory $\tau = (s)_0, a_0, R_0, s_1, a_1, R_1, s_2, a_2, R_2, \cdots$. The objective function of this process is to maximize the cumulative payoff of the trajectory, that is, to seek the optimal strategy, denoted as

$$argmax_\pi E\left[\sum_{t=0}^{\infty} \gamma^t R_\pi\left(s_t, s_{t+1}\right)\right] \tag{2}$$

where $\pi(s_t)$ is the action defined by the policy π given the state s_t.

D. DEEP DETERMINISTIC POLICY GRADIENTS

Reinforcement learning algorithms can be generally summarized into three categories: (1) Value-based methods, which calculate the value of each action by learning the

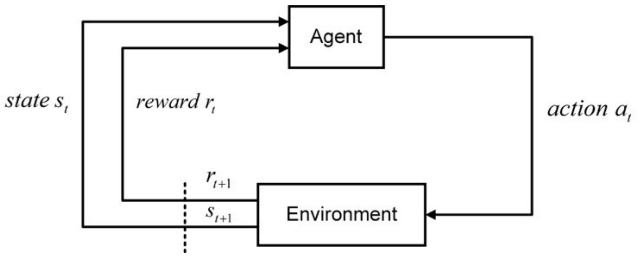


FIGURE 2. Typical Reinforcement Learning system. Picture inspired by ⟨⟨Reinforcement Learning⟩⟩ (Wei Zou et al.).

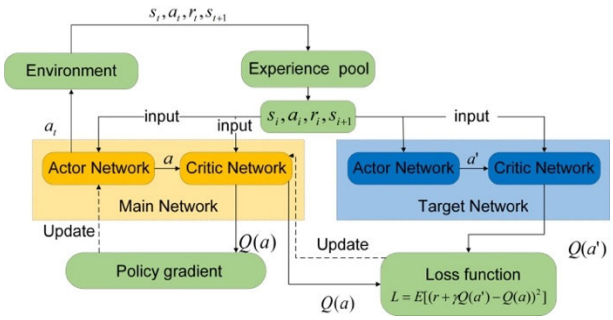


FIGURE 3. The architecture of DDPG algorithm.

optimal action value function, and finally select the action with the largest value, such as Q-learning; (2) Policy-based method, the Agent directly optimizes the policy, such as policy gradient (PG); (3) Actor-critic algorithm, which combines value-based methods and policy-based methods, such as Asynchronous Advantage Actor-critic (A3C). The strategy of the method based on value learning may lead to a big change in the value function when updating, which has a certain impact on the convergence. However, the method based on policy function requires the value function to fully converge before carrying out the step of strategy improvement, which has a certain impact on the learning rate. For the purpose of this article stock trading decisions which have high dimension or continuous state space problems, using the study after get cost function based on the value function, strategy, the value of the corresponding size need to compare various behavior, compared to a maximum value function the behavior of the process more difficult, using policy-based learning is easy to converge to a local optimum. However, the actor-critic algorithm takes advantage of the time-series differential method to update the value function in a single step and does not need to update the network parameters after an episode, which is faster than the value-based method and policy-based method. As mentioned above, DDPG algorithm is used to train reinforcement learning agents in this paper.

DDPG is an off-policy and Model-Free deep reinforcement learning algorithm based on Arctic-Critic for continuous motion space. This method combines the advantages of ''deterministic policy gradient'' (DPG) and Deep Q Network (DQN) [30], that is, DPG can handle tasks in continuous action space, and DQN can directly conduct end-to-end learning. DDPG introduces the experience playback mechanism of DQN algorithm, which solves the shortcoming that the Actor-Critic neural network cannot extract useful features due to the correlation before and after each parameter update.

DDPG algorithm uses two neural networks to approximate the value function. One neural network, called critic network, aims to learn a state-action value function. Another neural network, called an actor network, aims to learn policy functions that map states to deterministic actions.

Fig. 3 shows the flow of the DDPG algorithm. First initialize the parameters of the four networks. We use the main actor to interact with the environment, impose an action a on the environment, and the environment will return you the state s_{i+1} and reward r_i at the next moment. We use transition (s_i, a_i, r_i, s_{i+1}) to represent the reward r_i for action a_i in state s_i and the next state s_{i+1}. Put the transition in the experience pool. In this case, the Actor network has the function of interacting with the environment to obtain sample data (s_i, a_i, r_i, s_{i+1}). Then the network parameters are updated. Let's go back to the sampling process and update it. It's a circular process.

Critic network adopts TD error in DQN to update its parameters, and the loss function is the minimum mean square deviation:

$$L = \frac{1}{N}\sum_i \left(y_i - Q\left(s_i, a_i \mid \theta^Q\right)\right)^2 \tag{3}$$

where N is batch size, $Q\left(s_i, a_i \mid \theta^Q\right)$ represents a parameterized state-action value function, and y_i, s_i and a_i respectively represent ''TD target, state and action''. The actor network follows a deterministic policy to update its parameters in the direction of the action-value gradient. The gradient $\nabla_{\theta^\mu} J$ is defined as follows:

$$\nabla_{\theta^\mu} J = \frac{1}{N}\sum_i \nabla_a Q\left(s, a \mid \theta^Q\right)\big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu\left(s \mid \theta^\mu\right)\big|_{s_i} \tag{4}$$

where J and $\mu\left(s \mid \theta^\mu\right)$ is the cumulative reward and the parameterized policy function. The parameter vectors of the actor target network and the critic target network $\theta^{\mu'}$ and $\theta^{Q'}$ are recursively updated as follows:

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\,\theta^{\mu'} \tag{5}$$

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\,\theta^{Q'} \tag{6}$$

where, τ is the update factor affecting the update rate of the target network.

III. MODULE I: SUPERVISED LEARNING

For the first module of training, the main objective is to use supervised learning to predict coached actions in stock trading. Due to the supervision of the trading action directly predict a mentor in the learning process need to label, and in general trading data and can't see the trading expert action, so in the first stage, we mainly aim at the goal of predicting stock trading decisions, converts it to a return to task for

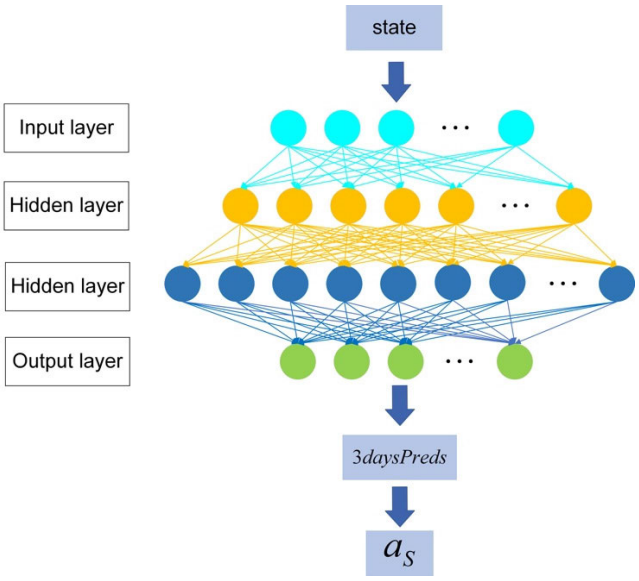


FIGURE 4. Predicting trading actions with LSTM.

processing, through forecasting the trend from stocks trading actions.

We use the stock data of 20 consecutive days to predict the short-term stock prices in the next few days and use the return rate of the next three days to represent the stock's rise and fall trend, which is calculated by Equation (7) [31].

$$\begin{aligned} &3daysPred \\ &= 0.4 * \frac{closeprice\,(n+1) - closeprice\,(n)}{closeprice\,(n)} + 0.32 \\ &\quad * \frac{closeprice\,(n+2) - closeprice\,(n+1)}{closeprice\,(n+1)} \\ &\quad + 0.28 * \frac{closeprice\,(n+3) - closeprice\,(n+2)}{closeprice\,(n+2)} \end{aligned} \tag{7}$$

where $3daysPred$ represents the three-day yield, and $closeprice(n)$ represents the closing price on the day n. In stock forecasting, it is often more practical to predict whether a stock will rise or fall sharply than to only predict the rise and fall, so we do the rise and fall trend 3 classification. If the three-day return rate is greater than 1%, it means that the stock price will rise in the next three days, representing short-term investment value. If the three-day return rate is less than -1%, it means that the stock price will decline in the next three days, which means that there is no short-term investment value. Three-day returns between 1% and -1% are considered neither up nor down. Therefore, as is shown in Fig. 4, we according to the rise and fall predicted in the first 20 days, the three forecast results of big rise, no rise, no fall and big fall are defined as buying, holding and selling respectively.

IV. MODULE II: REINFORCEMENT LEARNING

A. STATE SPACE

In the construction of reinforcement learning model, how to abstract the state is one of the core issues. In the financial field, the state can be understood as a price position of a stock, and the most basic data describing a stock is the stock price. In addition, researchers have calculated some technical index factors according to the relevant knowledge of statistics. Therefore, this paper uses the stock price data and related technical indicators as the abstract of the daily status of the stock. On the other hand, the trend of stock prices is sequential to a certain extent, and the price of the day is also affected by the previous price to a certain extent. Therefore, this paper adds the characteristics based on the wave theory to abstract the influence of the previous state on the current stock state.

1) BASIC STOCK INFORMATION

The state of the environment observed by the agent should contain basic information such as prices. We select five basic information of stocks: Open, Close, High, Low and Volume.

2) TECHNICAL INDICATORS

Stock price information has a lot of noise. In order to reduce data noise and uncertainty, technical indicators are usually used to assist decision-making in traditional artificial methods. Technical indicators are a very common and effective analysis and prediction tool in the financial field, which can well find out some relevant laws from historical data and give investors buy or sell advice. There are hundreds of types of technical indicators, mainly divided into three categories: trend technical indicators, swing indicators and energy indicators of technical indicators, most of the indicators are targeted to analyze the characteristics of stock trading behavior. We selected six representative technical indicators (MA, MACD, KDJ, RSI, OBV, BOLL [3]) from many indicators. These six technical indicators are the most common in quantitative trading, among which MA and MACD belong to trend indicators, KDJ and RSI belong to swing indicators, OBV and BOLL belong to energy indicators.

A brief description of each indicator is given below.

a. Moving average (MA)

MA is the average of irregular changes in daily stock prices. Compared with daily stock price changes, MA can be used to predict the change trend of stock prices by analyzing the position, relationship and direction of moving averages.

b. Moving average convergence divergence (MACD)

MACD uses the aggregation and separation between the short-term index average index and the long-term index average index to represent the current bearish state and the possible development trend of stock prices.

c. KDJ

KDJ index is mainly used to judge the degree of stock deviation from the normal level, calculate the value of K, D and J through the highest and lowest prices in a specific cycle and the closing price in the last calculation cycle, and finally form a graph to judge the stock price trend.

d. Relative strength index (RSI)

RSI is a relative strength index calculated based on the ratio of the number of rising points over a certain period of time

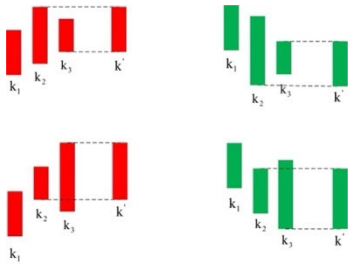


FIGURE 5. K-line merge.

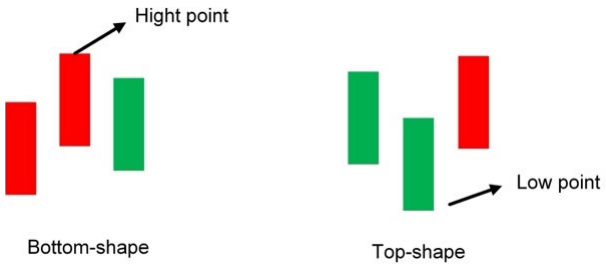


FIGURE 6. Bottom-shape and top-shape.

to the number of rising points and declining points over a specific period of time, describing the current and historical strength of a stock based on its closing price during the recent trading period.

e. On balance volume (OBV)

OBV is a technical indicator that reflects market sentiment, reflecting the investment activity of market investors through price and volume.

f. BOLL

The BOLL line is an auxiliary indicator used by investors to judge the stock price trend. It assesses the strength of the stock trend through the position of the stock price in the BOLL region.

3) FEATURES BASED ON WAVE THEORY

Elliott Wave Theory is a theory of stock technical analysis, which uses the correlation pattern of the stock market K line to analyze the trend of the stock market index and price. We score the K line according to the wave theory and input it into the trading decision system as a feature.

- K-line merge processing

Stock market volatility has randomness and instability, which affects the long-term stock price trend judgment. In this paper, the method thought of Chan Theory is used to standardize the K-line series and eliminate the inclusion relation in the financial time series.

Definition 1:

$$\begin{cases} k_{ih} < k_{(i+1)h} \text{ and } k_{il} > k_{(i+1)l} \\ k_{ih} > k_{(i+1)h} \text{ and } k_{il} < k_{(i+1)l} \end{cases} \tag{8}$$

Definition 1 illustrates the stock inclusion relationship, where k_{il}, k_{ih} is the low and high of the i-th K-line.

Definition 2:

$$\begin{cases} k'_l = \max\left(k_{il}, k_{(i+1)l}\right) \text{ and } k'_h = \max\left(k_{ih}, k_{(i+1)h}\right) \\ k'_l = \min\left(k_{il}, k_{(i+1)l}\right) \text{ and } k'_h = \min\left(k_{ih}, k_{(i+1)h}\right) \end{cases} \tag{9}$$

Definition 2 illustrates our principle of eliminating the stock inclusion relationship in the two trends of the K-line, where k'_l is the new K-line obtained after the merge processing. Fig 5 shows the whole k-line merging process, where the red line represents the merger treatment of the uptrend K-line, and the green line represents the merger treatment of the downtrend K-line.

- Bottom-shape and top-shape

After the k-line merger process, we divide the k-line into types, so that the wave theory can be added to judge the trend center.

Definition 3:

$$\begin{cases} k_{ih} < k_{(i+1)h} \text{ and } k_{(i+2)h} < k_{(i+1)h} \\ k_{il} < k_{(i+1)l} \text{ and } k_{(i+2)l} < k_{(i+1)l} \end{cases} \tag{10}$$

Definition 4:

$$\begin{cases} k_{ih} > k_{(i+1)h} \text{ and } k_{(i+2)h} > k_{(i+1)h} \\ k_{il} > k_{(i+1)l} \text{ and } k_{(i+2)l} > k_{(i+1)l} \end{cases} \tag{11}$$

Definition 3 and **Definition 4** illustrate the type of division of Bottom-shape and top-shape, respectively. Among the three adjacent K-lines, the division of top and bottom typing is clearly shown in Fig 6.

- Wave treatment

After dividing the top and bottom shape of K-line, we integrate them to better judge the following trend center. If two adjacent ones are top subtypes, and the top subtype high point in the back is larger than the top subtype high point in the front, only the top subtype high point in the back is retained. If two adjacent subtypes are bottom subtypes, and the bottom subtype low point is less than the previous bottom subtype low point, only the latter low point is retained. If the front one is the high point of the top classification and the back one is the low point of the bottom classification, and the high point is smaller than the low point, only the front high point is retained. If the front one is the bottom low and the back one is the top high, and the high is larger than the low, only the front low is retained.

The wave defined in this paper is the connection line of two adjacent bottom typing and top typing. Therefore, the high and low points of top typing and bottom typing are connected by line segments to form a wave. There are two types of waves. One type is the descending wave, which is the line connecting the top shape to the bottom shape, and the other type is the rising wave, which is the line connecting the bottom shape to the top shape.

- Trend center and scoring

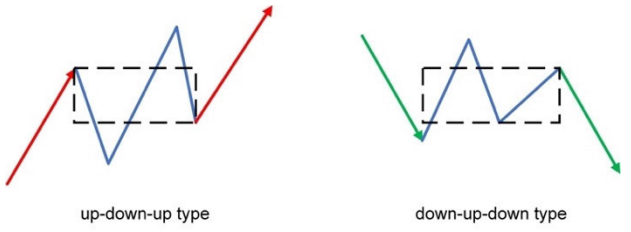


FIGURE 7. Movements in the central.

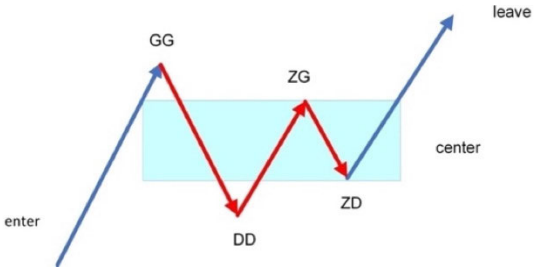


FIGURE 8. Central interval diagram.

In the trend of K-line, the center is the analysis basis of the current trend. We can predict the next step of the trend according to the position and destruction of the center. Therefore, identifying the center is the key to capture the stock price information. The part of a trend type that is overlapped by at least three consecutive trend types is called the trend center. There are two kinds of trend center formation, one is the formation of the callback (down-up-down trend center); One is the formation of a rebound (up-down-up trend center), as shown in the Fig. 7.

The central interval is fixed by the overlap of the first three-line segments. The low point of the first two high points is taken as the central high (ZG), and the high point of the first two low points is taken as the central low (ZD). The interval formed by ZG-ZD is the central interval, as interval, as shown in the Fig. 8.

Correspondingly, we design the stock price into three ranges: below the central range, within the central range and above the central range. The three price ranges are marked with numbers 0,1 and 2 respectively and are used as the feature input in the stock trading decision system.

4) FEATURES OF K-LINE WITH INVERTED SIGNAL

The K-line pattern of reversal signal is one of the clues of trend change. We selected six special K-line patterns with reversal signal to form new features.

- Hammer and hanging man

As shown in the Fig. 9, the hanging man follows a sustained uptrend. It is characterized by lower shadow that is more than twice the height of the body, with no or almost no shadow line. The hanging man is a sign of a reversal at the top. If this line appears in the market, users need to be careful that the trend of the stock price will go straight down.

Hammer occurred in the downward trend, reflects the stock more internal forces in the game, the trading day, continue

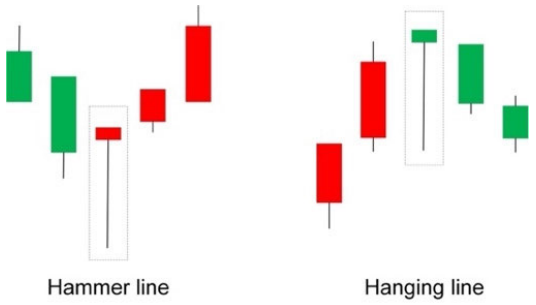


FIGURE 9. Hammer line and Hanging line.

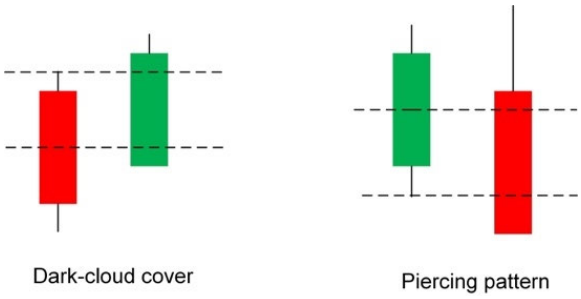


FIGURE 10. Dark-Cloud Cover and Piercing Pattern.

to stop selling continues to cause prices to fall but due to the buyer, prices started to rise, this suggests that persistent downtrend or will come to an end.

- Dark-cloud cover and piercing pattern

As shown in the Fig. 10, the dark-cloud cover pattern consists of two days of K-line, the first day K-line is a positive line, the second day is negative line, and the opening price of the negative line is higher than the high price of the previous day, and the closing price is in the lower part of the previous positive line. The dark-cloud cover pattern is a signal that prices are falling, and the market is about to go from good to bad.

The piercing pattern occurs in falling markets and is also made up of two K-lines. The first day's K-line is a negative line, the second day is a positive line, and the positive line opened lower than yesterday's low price and closed successfully through the previous trading day's K-line body midpoint. The piercing pattern is the opposite of the cloud top pattern and is a bullish signal.

- Doji

Doji is a special K-line shape with only the upper and lower shadow lines and no body. It indicates that the opening price and closing price within this period are the same, which is represented as a straight line in shape. The doji appears after the continuous rising or falling market, which means that the long and short forces reach equilibrium in several trading days and may change the late trend of a stock.

As shown in the Fig. 11, if the doji appears after a downward trend and a strong positive line's body forms behind the cross, and the positive line's body pierces the interior of

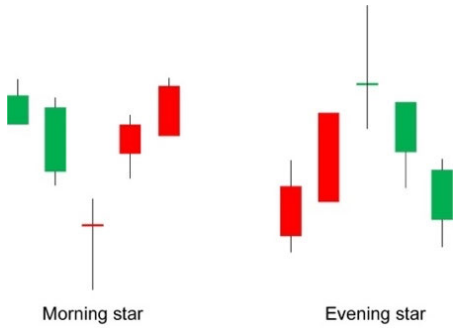


FIGURE 11. Morning star and evening star.

the previous negative line's body, then the cross is called the morning doji star. If the doji appears after an upward trend and a strong negative line's body forms behind the cross, and the negative line's body pierces the interior of the previous positive line's body, then the cross is called the evening doji star.

As for the characteristics of important turning points, we do the following operations: if there is a hammer, it is marked as 1; if there is a hanging man, it is marked as -1; if there is not, it is 0. The piercing pattern is marked as 1, the dark-cloud cover pattern is marked as -1, and the absence is marked as 0. The presence of the morning star is marked as 1, the evening star is marked as -1, and the absence is marked as 0.

B. ACTION SPACE

The Agent action is to output a certain value in a continuous action space, which contains the number of shares bought or sold. The action range designed in this paper is between -1 and 1, and the action space $A = [a, \omega]$: a defines a series of actions on the stock, which respectively represent buy, hold and sell, and ω represents the trading share.

C. REWARD

Agent reward value design is that the environment feeds back a corresponding reward value according to the action made by the Agent, and the agent adjusts its strategy according to this reward value and makes the next action according to the observed state received. We first define the payoff at time t:

Definition 5:

$$Earnings = Banlance + Volume \times P_t - initMoney \quad (12)$$

where *Banlance* represents the remaining capital after trading the stock, *Volume* represents the total amount of the current position, P_t represents the current price of the stock, and *initMoney* represents the initial capital.

The reward to the agent for acting at time t is

Definition 6:

$$reward = \begin{cases} \dfrac{Earnings}{initMoney}, & if\ Earnings > 0 \\ -0.1, & others \end{cases} \quad (13)$$

V. SUPERVISED LEARNING GUIDES REINFOECEMENT LEARNING ALGORITHMS

In the early stage of reinforcement learning, the agent can only learn by interacting with the environment without prior experience or higher-level guidance, which means that the agent needs to learn from scratch. Moreover, the action space of stock trading decision is too large, and the initial training of reinforcement learning is random exploration, so the convergence speed of reinforcement learning training without interference is slow or difficult to achieve convergence.

Based on the above problems, we propose a supervised reinforcement learning algorithm. It combines the guidance of the supervisor of supervised learning and the self-learning characteristics of reinforcement learning, introduces the experience of supervisor based on reinforcement learning, and adds the supervisor's supervision and guidance to the exploration process of the agent, endows the Agent with prior knowledge. Speed up the process of Agent finding the optimal solution.

This section describes a network-based model that is trained using a supervised stock dataset while leveraging RL to further improve the model when interacting with the environment. With the optimization of the agent network structure, the guiding role of supervised learning is weakened, and the trading strategy can be obtained only by the autonomous exploration of reinforcement learning. The advantage of this framework is that it can use both supervised learning and reinforcement learning to train a single model. Compared with directly using reinforcement learning to judge trading decisions, this method is easier to achieve convergence. The supervised reinforcement learning algorithm model is shown as Fig. 12.

In supervised reinforcement learning, we design an action update strategy to achieve the guidance of the supervised learning tutor. The action update strategy is as follows:

Definition 7:

$$a_\xi = (1 - \lambda)\, a_t^R + \lambda a_t^S \quad (14)$$

The supervised learning action a_t^S is obtained by LSTM neural network, and the action a_t^R is obtained by DDPG algorithm. a_ξ is calculated according to the formula, where the decay factor λ is a number between 0 and 1, which decreases with the number of training iterations, and then the formula is used to judge the action a_t. At the beginning of the experiment, if the decay factor λ is taken to be a large value close to 1, the more weight a_t^S occupies in the optimization objective. It shows that when the number of simulations is small, the model will refer to the supervised learning strategy network more. The extreme example is when λ is 1, there is no simulation experience to refer to, so the model is degraded to directly use the target of the supervised learning strategy network as the search target, and the action decision is mainly determined by the supervised learning. With the increase of the number of iterations, the attenuation factor lambda will reduce gradually, a_t^R the importance of the constantly improve, to the middle and later periods of the experiment,

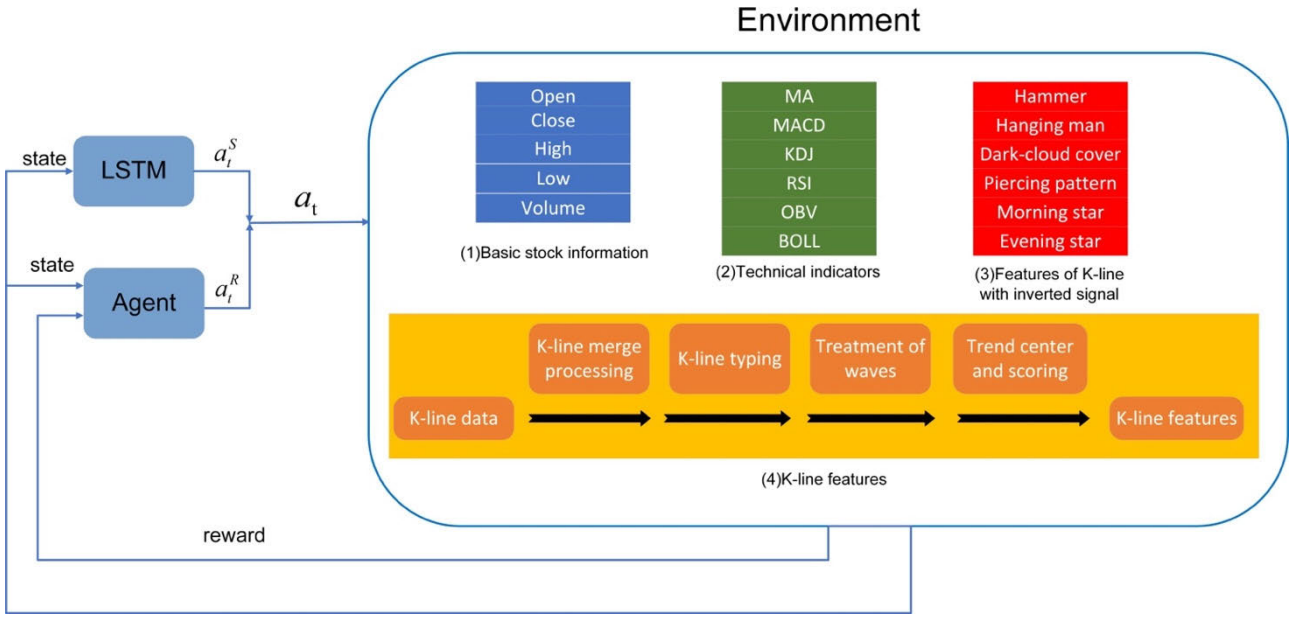


FIGURE 12. The architecture of the supervised reinforcement learning algorithm.

the reinforcement learning network has been very well learn good parameters, and the attenuation factor lambda reduce to a value close to zero, then the a_t^R almost full weight, action decision-making is mainly decided by reinforcement learning. In this process, guiding prior knowledge is introduced at the initial stage of training to increase the learning effect of the system. Meanwhile, the guiding role of supervised learning is gradually weakened, and the advantage of reinforcement learning to explore optimization is played, so that the agent can find the optimal decision as soon as possible.

The architecture of the algorithm is that LSTM provides early guidance for DDPG algorithm. In the learning process, LSTM data is also stored as training samples to update the predicted value, thus providing more effective guidance for DDPG strategy exploration. The detailed flow of the LSTM-DDPG algorithm is as follows.

(1) Initialize the DDPG algorithm strategy and set the state observation space to D and the state action space to A.

(2) In $t= 1, \cdots, T$, perform the following steps:

a. Action $a_0^R(S_t)$ of DDPG algorithm decision is generated, but not executed.

b. LSTM algorithm is used to get the guiding action $a_0^S(S_t)$.

c. Perform action $a_0(S_t) = (1 - \lambda)\, a_0^R(S_t) + \lambda a_0^S(S_t)$.

d. Get the reward r_i this time, switch to the next state is S_{t+1}, store the state action data $(S_t, a_0^S(S_t))$ to A, and store the state action transfer data $(S_t, a_0(S_t)@\text{comma}S_{t+1}, r_t)$ to D.

(3) Calculate the reward $\sum_{t=1}^{T} r_t$ of this iteration.

(4) Update D.

(5) The current iteration process is regarded as the first policy iteration, and the following cycle is carried out within the set number of policy iterations:

a. Through DDPG algorithm and based on the previous iteration reward, strategy $a_k^R(S_t)$ is generated for the k time.

b. A is obtained based on the data in $a'_k = a_0^S + \cdots + a_{k-1}^S$.

c. In $t= 1, \cdots, T$, repeat step (2) and add the component of $a'_k(S_t)$ to the strategy.

d. Repeat steps (3) and (4).

(6) The iteration ends when the termination condition is met.

VI. EXPERIMENT

A. EXPERIMENTAL DATASET

Putting our method into real world financial data, we select nine individual and index stocks in different markets as representatives for testing to verify the behavior of our learning agent in some real trading scenarios. Table 1 describes all data datasets, and Table 2 shows the division of the experimental and test datasets in training. All data used in this paper are available on the Yahoo Finance and Tushare Finance Data. Fig. 13 are the plot of the experimental index stocks and experimental dataset of six individual stocks.

B. EXPERIMENTAL SETTING AND SCHEME

In the Supervised learning effect experiment, we used the results predicted by the trained LSTM model to conduct the experiment, and the initial cash was set to \$50,000, £50,000 and ¥50,000 respectively. In the Algorithm convergence rate and Performance analysis experiments, the first eight years of data are used as the training set, and the remaining three years of data are used as the test set. The initial cash is set at \$100,000, £100,000 and ¥100,000 respectively. Transaction costs are set to zero during training. The main purpose of the LSTM model we use is to predict the closing price of the next three trading days based on the information of the previous

TABLE 1. Experimental datasets of stocks in three stock markets.

Market	Dataset	Number of data	Number of features	Total period
US	AAPL	2769	18	2010/01/01-2020/12/31
	IBM	2769	18	2010/01/01-2020/12/31
	DJIA	2769	18	2010/01/01-2020/12/31
	IXIC	2769	18	2010/01/01-2020/12/31
EUR	ULVR	2024	18	2014/01/01-2022/01/01
	BATS	2024	18	2014/01/01-2022/01/01
	FTSE	2769	18	2010/01/01-2020/12/31
	DAX	2769	18	2010/01/01-2020/12/31
CN	600519	2674	18	2010/01/01-2020/12/31
	601398	2674	18	2010/01/01-2020/12/31
	399300	2674	18	2010/01/01-2020/12/31
	000001	2769	18	2010/01/01-2020/12/31

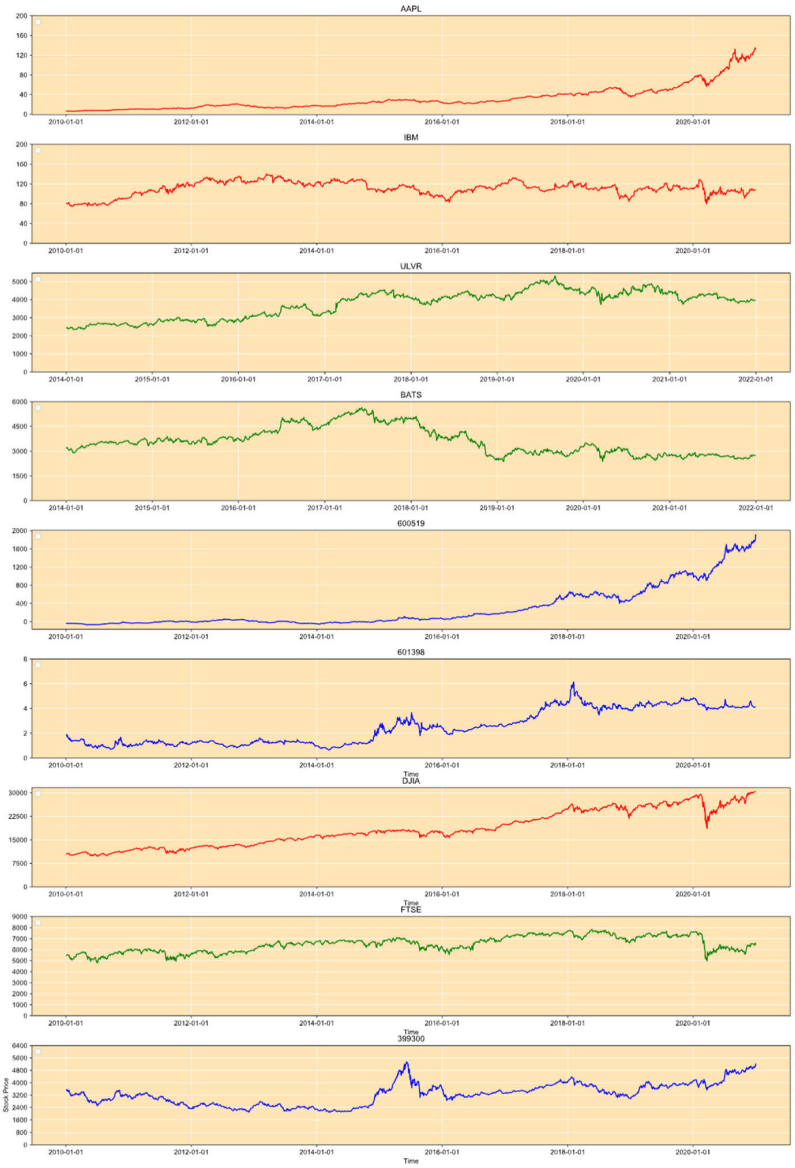


FIGURE 13. Experimental dataset of six Individual stocks and three Index stocks.

20 days, and finally calculate the return rate of the next three days to judge the trading action. First, we preprocess the original data, then build features according to the preprocessed data set, divide the processed data set into a training

TABLE 2. Training and testing set in experimental datasets.

Market	Stock Name	Training period	Testing period
US	AAPL	2010/01/01-2017/12/31	2018/01/01-2020/12/31
	IBM	2010/01/01-2017/12/31	2018/01/01-2020/12/31
	DJIA	2010/01/01-2017/12/31	2018/01/01-2020/12/31
	IXIC	2010/01/01-2017/12/31	2018/01/01-2020/12/31
EUR	ULVR	2014/01/01-2019/12/31	2020/01/01-2022/01/01
	BATS	2014/01/01-2019/12/31	2020/01/01-2022/01/01
	FTSE	2010/01/01-2017/12/31	2018/01/01-2020/12/31
	DAX	2010/01/01-2017/12/31	2018/01/01-2020/12/31
CN	600519	2010/01/01-2017/12/31	2018/01/01-2020/12/31
	601398	2010/01/01-2017/12/31	2018/01/01-2020/12/31
	399300	2010/01/01-2017/12/31	2018/01/01-2020/12/31
	000001	2010/01/01-2017/12/31	2018/01/01-2020/12/31

set and a test set, select relevant input features, and establish model training. Finally, the prediction results of the model are compared with the real measurement results, the error between the predicted value and the real value is measured by the loss function MSE, and the model is constantly adjusted and optimized. We use the grid search method to find a good combination of hyperparameters. A large amount of empirical evidence has demonstrated the effectiveness of grid search in terms of optimizing parameters [32]. Grid search samples the following hyperparameters: (1) The number of LSTM layers, ranging from 1 to 5; (2) the number of epochs, from 10 to 100, and (3) the neuronal activation function; (4) batch size, from 16 to 64; (5) The number of neurons per hidden layer, from 2 to 200. Finally, the topology of LSTM network is determined. We set 18 features and 20 timesteps in input layer. And in LSTM layer, we set 60 hidden neurons and 0.1 for dropout rate. In dense layer, we apply 16 neurons and relu activation function.

C. PERFORMANCE EVALUATION METRICS

We choose Annual Return, Cumulative Return, Maximum Drawdown, Sharpe Ratio, Sortino Ratio, Calmar Ratio, Longest drawdown (DD) days is used as a performance evaluation metrics to compare the performance of the model.

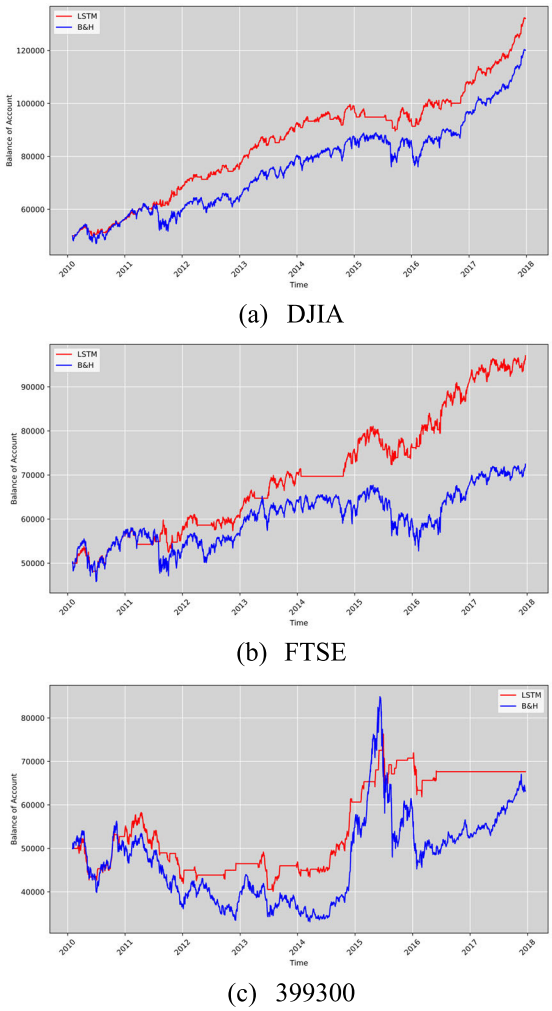
The Annual Return is the cumulative returns over the total trading years.

$$AnnualReturn = \frac{CurrentEarnings}{Principal} \times \frac{yeardays}{totaldays} - 1 \quad (15)$$

where year days is the number of days during the year that a stock is open for trading, total days is number of days invested. Cumulative Return is defined as the change between the initial amount of investment and the final amount of investment.

$$CumulativeReturn = \frac{Amountatlast - Amountatstart}{Amountatstart} \times 100 \quad (16)$$

For all two return values, the larger the better. The Maximum Drawdown refers to the largest amount of money a fund has lost over a period. Sharpe Ratio, Sortino Ratio, Calmar Ratio and Longest drawdown days are all used to measure the risk under the return of trading strategy. The lower the Sharpe Ratio, Sortino Ratio and Calmar Ratio are, the greater the risk is. The smaller the Longest drawdown (DD) days, the less risk there is.



(a) DJIA

(b) FTSE

(c) 399300

FIGURE 14. The LSTM trend forecasting strategy and B&H account net worth changes are compared on the index stock dataset.

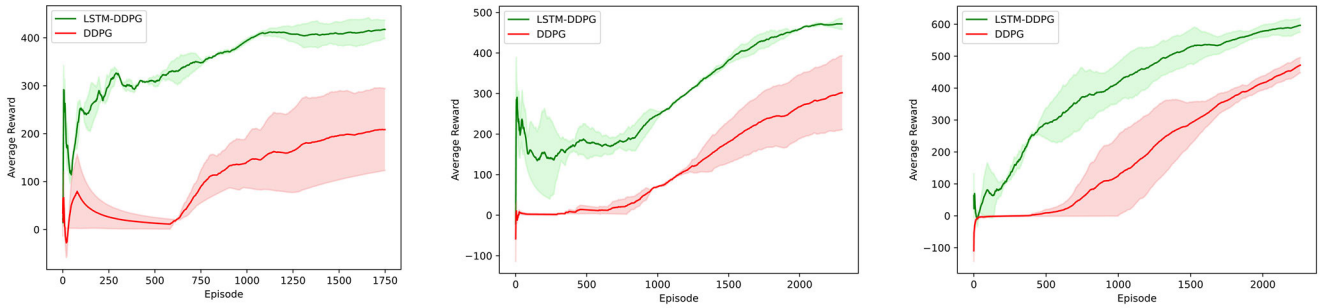


FIGURE 15. Average rewards of the LSTM-DDPG and DDPG algorithms on the index stock dataset.

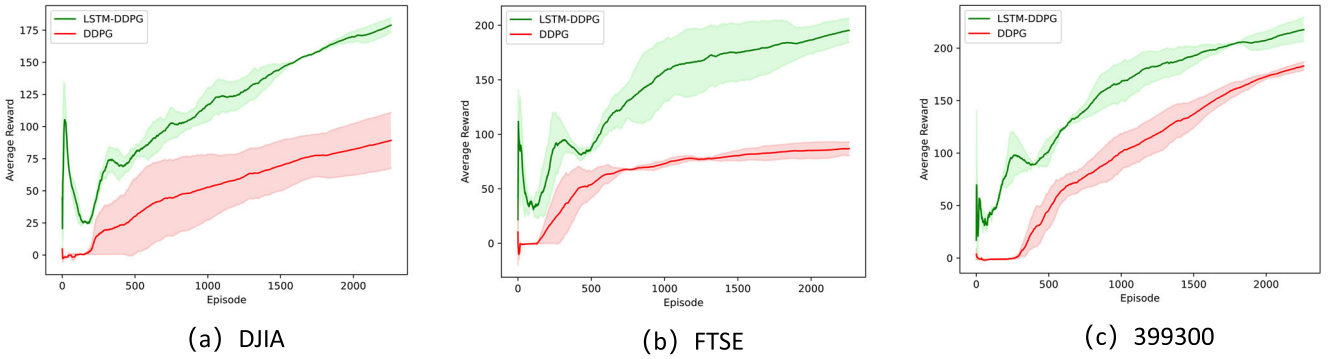


FIGURE 16. Average rewards of the LSTM-DDPG and DDPG algorithms on the index stock dataset.

D. BASELINE METHODS

1) **Buy and Hold (B&H)** [33] is a common investment strategy in which investors purchase shares and hold them for a long time without operating during the whole investment process.

2) **Moving Average Convergence and Divergence (MACD)** trading strategy is an evolution from the index moving average and was proposed by Gerald Appel in 1979. This strategy uses the positional relationship between long- and short-term averages to make trading decisions. The short-term average above the long-term average is a long signal, the short-term average below the long-term average above the long-term average is a short signal.

VII. RESULTS

A. SUPERVISED LEARNING EFFECT

Under the mechanism of supervised learning guidance, the agent first uses the LSTM neural network to predict the stock price, and then judges the buying and selling action. Let 3daysPred be the judgment index of the short-term stock price trend, then when 3daysPred$>1\%$, perform the long operation. Specifically, use all funds to open long positions. If you hold a short position, you first close the short position. When 3daysPred$<-1\%$, the shorting operation is performed. If the long position is held, the long position is first deflated. When $-1\%<$3daysPred$<1\%$, no action is performed.

Fig. 14 records the comparison between LSTM's prediction of short-term stock price up-and-down trend and its execution of buying and selling actions and the performance of benchmark model B&H on three index stocks. Specifically, the initial cash is set as \$50,000, £50,000 and ¥50,000 respectively. Through the change of total assets compared with the benchmark model in Fig. 14, we can see that on all three index stock data sets, the performance of our guidance model as a trading strategy is better than that of the benchmark model. Therefore, we can conclude that, LSTM prediction model can provide valuable reference for reinforcement learning agents, that is, it can be used as a guide to participate in the interaction between agents and environment.

B. ALGORITHM CONVERGENCE RATE

We conduct experiments on DJIA, FTSE and 399300 index stocks to compare the convergence speed and performance of the model. When there is no significant change in the average reward obtained by the deep reinforcement learning strategy under an episode during training, we can consider DRL training to converge approximately. We report the learning curve on the training dataset in Fig. 15 and the learning curve on the test dataset in Fig. 16, and LSTM-DDPG always has a higher score than DDPG. As shown in Fig. 15 and Fig. 16, LSTM-DDPG is significantly superior to DDPG. In the DJIA training dataset, LSTM-DDPG roughly converges to

TABLE 3. Training and testing set in experimental datasets.

Stock	Method	Annual Return	Accumulated Return	MDD	Sharpe Ratio	Sortino ratio	Calmar Ratio	Longest DD days
AAPL	B&H	47.39%	220.18%	-38.51%	1.28	1.88	1.23	906
	MACD	45.20%	204.32%	**-15.92%**	**1.81**	**2.81**	**2.84**	**237**
	LSTM	21.41%	78.96%	-39.27%	0.90	1.25	0.55	1021
	LSTM-DDPG	**58.13%**	**295.37%**	-46.61%	1.35	2.04	1.25	1089
IBM	B&H	-2.01%	-5.92%	-38.98%	0.08	0.11	-0.05	892
	MACD	0.69%	2.09%	**-24.65%**	0.13	0.19	0.03	**650**
	LSTM	-12.50%	-4.35%	-33.21%	-0.11	-0.15	-0.13	1046
	LSTM-DDPG	**5.86%**	**18.65%**	-37.18%	**0.37**	**0.52**	**0.16**	972
ULVR	B&H	-4.69%	-9.22%	-23.69%	-0.09	-0.14	-0.20	373
	MACD	-17.87%	-32.49%	-39.01%	-1.15	-1.54	-0.46	906
	LSTM	-11.81%	-6.06%	**-20.93%**	-0.28	-0.39	-0.29	545
	LSTM-DDPG	**1.37%**	**2.78%**	-28.31%	**0.18**	**0.26**	**0.05**	**328**
BATS	B&H	-8.44%	-16.26%	-32.07%	-0.18	-0.24	-0.26	853
	MACD	-27.97%	-48.05%	-49.59%	-1.72	-2.14	-0.56	1218
	LSTM	2.65%	1.31%	**-22.46%**	0.17	0.23	0.06	**217**
	LSTM-DDPG	**9.75%**	**20.59%**	-35.36%	**0.44**	**0.64**	**0.28**	1089
600519	B&H	45.37%	195.57%	-33.50%	1.33	2.04	1.35	951
	MACD	18.60%	63.49%	-25.41%	0.79	1.19	0.73	452
	LSTM	40.62%	168.44%	-24.62%	0.59	0.41	0.89	698
	LSTM-DDPG	**64.18%**	**320.50%**	**-22.04%**	**1.89**	**3.09**	**2.91**	**327**
601398	B&H	-2.61%	-7.38%	-34.45%	-0.03	-0.05	-0.08	1037
	MACD	-12.57%	-32.09%	-41.58%	-0.69	-1.01	-0.30	1045
	LSTM	0.75%	2.19%	**-11.42%**	0.13	0.19	0.07	**187**
	LSTM-DDPG	**5.66%**	**17.31%**	-17.09%	**0.43**	**0.33**	**0.72**	262
IXIC	B&H	25.56%	97.94%	-33.84%	0.99	1.39	0.85	904
	MACD	-19.82%	-48.45%	-54.21%	-0.73	-0.82	-0.37	943
	LSTM	6.70%	21.47%	-36.46%	0.45	0.57	0.18	870
	LSTM-DDPG	**28.79%**	**113.63%**	**-28.03%**	**1.06**	**1.49**	**0.91**	**554**
DAX	B&H	2.15%	6.58%	-40.92%	0.21	0.28	0.03	1123
	MACD	-52.88%	-89.53%	-90.47%	-1.54	-1.56	-0.58	1575
	LSTM	1.77%	5.42%	-40.56%	0.19	0.26	0.04	1096
	LSTM-DDPG	**2.93%**	**9.04%**	**-38.78%**	**0.82**	**0.39**	**0.06**	**761**
000001	B&H	1.16%	3.39%	-30.76%	**0.16**	0.22	0.04	986
	MACD	-24.00%	-54.85%	-54.85%	-1.66	-2.03	-0.44	1206
	LSTM	0.60%	1.74%	-32.74%	0.13	0.18	0.02	839
	LSTM-DDPG	**1.27%**	**3.73%**	**-24.82%**	0.15	**0.22**	**0.05**	**487**

the 1100 episode, while DDPG roughly converges to the 1500 episode. In the FTSE training dataset, LSTM-DDPG roughly converges to the 2000 episode, while DDPG still has an upward trend after 2250 episodes. On the 399300 training dataset, LSTM-DDPG still has an upward trend when it reaches the 1500 episode, but the upward trend gradually flattens out, while DDPG has no obvious convergence trend after 2250 episodes. In Fig. 16, we can see that LSTM-DDPG still has an upward trend after 2000 episodes in the training data set, but the average reward is always higher than DDPG. Moreover, it can be clearly seen that in the training process of the three index stock data sets, the average reward obtained by LSTM-DDPG training is generally greater than that obtained by DDPG training, and the training effect is significant. Among them, the average reward of LSTM-DDPG in the training process has a trend of rising in the early stage, then falling sharply, and then rising slowly. This is because under the guidance of supervised learning in the early stage, the agent can quickly find the target. As the guidance of supervised learning weakens, reinforcement learning continues to explore and trial and error, and the training results continue to rise after large fluctuations. From this, we conclude that our proposed supervised learning-style reinforcement learning model learns better policies than reinforcement learning-based methods in similar training steps.

C. PERFORMANCE ANALYSIS

In Table 2, we compare the experimental results of AAPL, IBM, BATS, ULVR, 301398, 300519, IXIC, DAX and 000001 data sets. The results show that the LSTM-DDPG algorithm is significantly better than other baseline algorithms on the data of cumulative returns. Especially for the two data sets of AAPL and 600519, the cumulative return rate of LSTM-DDPG algorithm increases to 295.37% and 320.50%, respectively. The Sharpe ratio, Sortino and Calmar ratios returned by the LSTM-DDPG algorithm are all the highest on the IBM, BATS, ULVR, 301398, 300519, IXIC, DAX and 000001 data sets. The Longest drawdown

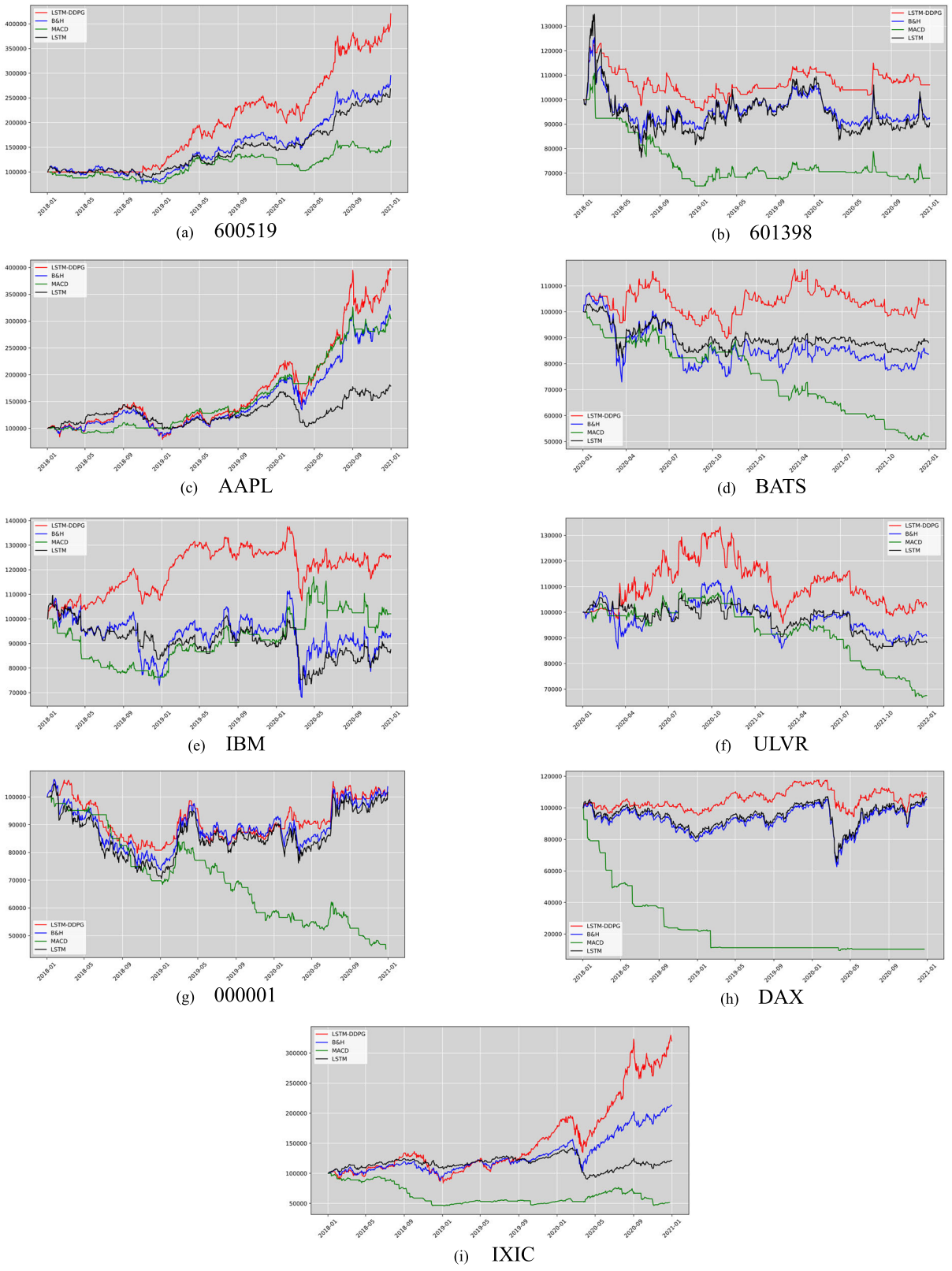


FIGURE 17. Net asset change curves of LSTM-DDPG and benchmark models on different individual stock datasets.

days were the smallest on ULVR, 600519, IXIC, DAX, and 000001 data sets. This means that the LSTM-DDPG algorithm has the minimum risk under the premise of obtaining the same return. In addition, we found that the MDD value

of LSTM-DDPG algorithm was not optimal in the six data sets of AAPL, IBM, BATS, ULVR, IXIC and DAX, but the cumulative Return, Annual Return, Sharpe ratio, and Sortino and Calmar ratio were all optimal. We annotated the optimal results returned by the experiment of each data set. Overall, the performance of LSTM-DDPG algorithm was the best, which showed the effectiveness of LSTM-DDPG algorithm.

In addition, we show the change in net assets of the LSTM-DDPG algorithm and other baseline algorithms in the six individual stocks and 3 index stocks test data sets in Fig 17. The initial cash is set at $100,000, £100,000 and $100,000 respectively. As can be seen from Figure 17, the blue line representing LSTM-DDPG algorithm is higher than other curves in almost all the tests. In all data sets, with the fluctuation of price, the return curve of the baseline algorithm will drop sharply when the price drops, and LSTM-DDPG algorithm will be affected by the price to a certain point, but the return curve is always above other baseline algorithms. When the price rises, LSTM-DDPG algorithm can obviously obtain higher profits and perform better than other baseline algorithms. Experimental results further verify the effectiveness and stability of LSTM-DDPG algorithm.

VIII. CONCLUSION

With the increasing application of algorithmic trading in stock trading, it has become a problem for stock traders to find the optimal trading strategy using algorithmic trading. We try to find the optimal trading strategy using the reinforcement learning. Classical reinforcement learning lacks prior experience in the early training process, the model performance is poor, and the large trading action space may lead to non-convergence of the algorithm. Considering the characteristics of reinforcement learning algorithms and stock trading data, we propose a new algorithmic trading idea called supervised reinforcement learning method. We use LSTM as the supervised learning module and DDPG as the reinforcement learning module. By participating in reinforcement learning in the early stage of supervised learning to interact with the environment, agents can obtain guiding prior experience, so that the exploration process can quickly proceed to the direction of optimal trading strategy. In addition, we consider multi-class features as state input for training, including wave theory and stock trend reversal. The experimental results on the US, UK and Chinese stock markets show that the proposed LSTM-DDPG algorithm obtains more profits and takes less risks than the baseline algorithm in all market environments.

This paper develops a model of automated investment trading decisions in stocks, which also has some limitations. First, as prices fluctuate, the model results exhibit the same volatility impact as prices and do not show a steady increase in profits regardless of price rises or falls. Therefore, the robustness of the algorithm to the market needs to be improved. Secondly, the model only shows good performance in the field of stock trading, and its application in futures, funds and foreign exchange markets needs further research. In addition, the application of the model to portfolio management and venture capital management is also the direction of our future research.

REFERENCES

[1] M. Kim and L. M. McAlister, "Stock market reaction to unexpected growth in marketing expenditure: Negative for sales force, contingent on spending level for advertising," *J. Marketing*, vol. 75, no. 4, pp. 68–85, Jul. 2011.
[2] J. Y. Campbell and Y. Hamao, "Predictable stock returns in the United States and Japan: A study of long-term capital market integration," *J. Finance*, vol. 47, no. 1, pp. 43–69, Mar. 1992.
[3] C. Ma, J. Zhang, J. Liu, L. Ji, and F. Gao, "A parallel multi-module deep reinforcement learning algorithm for stock trading," *Neurocomputing*, vol. 449, pp. 290–302, Aug. 2021.
[4] L. Weng, X. Sun, M. Xia, J. Liu, and Y. Xu, "Portfolio trading system of digital currencies: A deep reinforcement learning with multidimensional attention gating mechanism," *Neurocomputing*, vol. 402, pp. 171–182, Aug. 2020.
[5] R. J. Elliott, J. Van Der Hoek, and W. P. Malcolm, "Pairs trading," *Quant. Finance*, vol. 5, no. 3, pp. 271–276, 2005.
[6] K. Kim, "Enhancing reinforcement learning performance in delayed reward system using DQN and heuristics," *IEEE Access*, vol. 10, pp. 50641–50650, 2022.
[7] Y. Kim and H. Lim, "Multi-agent reinforcement learning-based resource management for end-to-end network slicing," *IEEE Access*, vol. 9, pp. 56178–56190, 2021.
[8] F. Gallego, C. Martin, M. Díaz, and D. Garrido, "Maintaining flexibility in smart grid consumption through deep learning and deep reinforcement learning," *Energy AI*, vol. 13, Jul. 2023, Art. no. 100241.
[9] H. Li, C. H. Dagli, and D. Enke, "Short-term stock market timing prediction under reinforcement learning schemes," in *Proc. IEEE Int. Symp. Approx. Dyn. Program. Reinforcement Learn.*, Apr. 2007, pp. 233–240.
[10] Z. Jia, Q. Gao, and X. Peng, "LSTM-DDPG for trading with variable positions," *Sensors*, vol. 21, no. 19, p. 6571, Sep. 2021.
[11] Y. Li and Y. Chen, "Enhancing a stock timing strategy by reinforcement learning," *IAENG Internaitonal J. Comput. Sci.*, vol. 48, pp. 1–10, Dec. 2021.
[12] W. Li and J. Liao, "A comparative study on trend forecasting approach for stock price time series," in *Proc. 11th IEEE Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Oct. 2017, pp. 74–78.
[13] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn.*, 1993, pp. 330–337.
[14] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, May 1992.
[15] X. Ji, J. Wang, and Z. Yan, "A stock price prediction method based on deep learning technology," *Int. J. Crowd Sci.*, vol. 5, no. 1, pp. 55–72, Apr. 2021.
[16] J. Liu, C.-M. Lin, and F. Chao, "Gradient boost with convolution neural network for stock forecast," 2019, *arXiv:1909.09563*.
[17] R. Zhang, Z. Wu, and S. Wang, "Prediction of stock based on convolution neural network," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 3175–3178.
[18] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 387–395.
[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[20] B. Gülmez, "Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm," *Expert Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120346.
[21] N. Yadav Vanguri, S. Pazhanirajan, and T. Anil Kumar, "Extraction of technical indicators and data augmentation-based stock market prediction using deep LSTM integrated competitive swarm feedback algorithm," *Int. J. Inf. Technol. Decis. Making*, pp. 1–27, Feb. 2023.
[22] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150199–150212, 2020.
[23] S. Ji, J. Kim, and H. Im, "A comparative study of Bitcoin price prediction using deep learning," *Mathematics*, vol. 7, no. 10, p. 898, 2019.

[24] Y. Wang, Y. Liu, M. Wang, and R. Liu, "LSTM model optimization on stock price forecasting," in *Proc. 17th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci. (DCABES)*, Oct. 2018, pp. 173–177.
[25] L. A. Baxter, "Markov decision processes: Discrete stochastic dynamic programming," *Technometrics*, vol. 37, no. 3, p. 353, Aug. 1995.
[26] L. Oelschläger and T. Adam, "Detecting bearish and bullish markets in financial time series using hierarchical hidden Markov models," *Stat. Model.*, vol. 23, no. 2, pp. 107–126, 2023.
[27] Z. Wen, D. O'Neill, and H. Maei, "Optimal demand response using device-based reinforcement learning," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2312–2324, Sep. 2015.
[28] P. Ladosz, L. Weng, M. Kim, and H. Oh, "Exploration in deep reinforcement learning: A survey," *Inf. Fusion.*, vol. 85, pp. 1–22, Sep. 2022.
[29] A. Gosavi, "Reinforcement learning: A tutorial survey and recent advances," *INFORMS J. Comput.*, vol. 21, no. 2, pp. 178–192, May 2009.
[30] Y. Song, J. W. Lee, and J. Lee, "A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction," *Int. J. Speech Technol.*, vol. 49, no. 3, pp. 897–911, Mar. 2019.
[31] V. Mnih, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[32] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–22232, Oct. 2017.
[33] B. G. Malkiel, "A random walk down Wall Street: Including a life-cycle guide to personal investing," *Ww Norton Company*, vol. 40, no. 17, p. 1566, 1990.

KUI FU received the Ph.D. degree from the Wuhan University of Technology, China, in 2007. He is currently an Associate Professor with the School of Economics, Wuhan University of Technology. His research interests include artificial intelligence and quantitative trading, business intelligence and big data analytics, and fintech.

YIDONG YU received the B.S. degree from the Hubei University of Technology, China, in 2021. He is currently pursuing the master's degree with the Wuhan University of Technology. His research interests include machine learning, artificial intelligence, and quantitative trading.

BING LI received the Ph.D. degree from the City University of Hong Kong, in 2013. She is currently an Associate Professor with the School of Economics, Wuhan University of Technology, China. Her research interests include digital economy, emergency management, digital trade, data analytics, and intelligent networks.