

Received 10 July 2022, accepted 30 July 2022, date of publication 8 August 2022, date of current version 15 February 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3197165

APPLIED RESEARCH

Reinforcement Learning for Stock Prediction and High-Frequency Trading With T+1 Rules

WEIPENG ZHANG¹, TAO YIN¹, YUNAN ZHAO², BING HAN², AND HUANXI LIU¹ 

¹Shanghai Jiao Tong University, Shanghai 200240, China

²MYbank, Ant Group, Shanghai 310000, China

Corresponding author: Huanxi Liu (lhxsjtu@sjtu.edu.cn)


This work was supported in part by the Interdisciplinary Program of Shanghai Jiao Tong University under Grant ZH2018QNB12 and Grant YG2022QN011; and in part by the Ant Group, China.

ABSTRACT The high-frequency trading framework for the price trend prediction model and trading strategy has been a popular approach for T+0 trading in the stock market. The prediction model is used to predict price trends, and the trading strategy is used to determine the price and volume of the order. Most trading strategies consist of multiple trading logic associated with certain tuning parameters. These parameters significantly affect the profitability of high-frequency trading frameworks. There are two main disadvantages of this framework: 1) the price trend prediction model can not adapt to the current market data distribution, and 2) the trading strategy can not adapt to the current market conditions automatically. Thus, the framework cannot always maintain positive revenue. To address this problem, we propose a novel dynamic parameter optimization algorithm based on reinforcement learning for stock prediction and trading, and to generate an adaptive trading framework. First, we use a rolling model training method for stock price trend prediction. Second, we regard each set of strategy parameters as action and devise an inverse reinforcement learning algorithm for the reward function to accurately estimate the reward of each action. Because of the T+1 trading rules of the Chinese stock market, we consider the constraint of limited short position in the reward function. Finally, a reward-enhanced upper confidence bound (UCB) selection algorithm is proposed to automatically optimize the parameters of the trading logic in real-time trading. The experimental results show that our method achieves competitive performance in the Chinese stock market.

INDEX TERMS High-frequency trading, inverse reinforcement learning, multi-armed bandit learning.

I. INTRODUCTION

High-frequency trading (HFT) has developed deeply in each part, including price indicators, machine learning models, and trading strategies. From an academic viewpoint, HFT is an online decision-making problem that should take action at each trading time. The action is to send a trading order to the exchange, or do nothing, and an order is composed of three elements: *direction* (i.e., buy or sell), *price*, and *volume*. There are many technical indicators to address the order direction, such as moving average convergence and divergence (MACD), *K & D* line (KDJ), Williams %R (WR), relative strength index (RSI), and stop and reversal system (SAR).

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

With the popularity of machine learning algorithms, most of the quantitative traders have integrated the machine learning model into the trading strategy, which profoundly improves the profitability of the trading strategy with the high accuracy of machine learning models to predict price movement direction. Owing to the successful integration of the machine learning model, the new generation HFT framework consists of two modules at the production level, 1) a machine learning model, aims to determine the order direction by predicting the future price trend. 2) the trading strategy, provides the order price and order volume, which are usually designed by human expert, such as active policy or market making policy.

In the Chinese stock market, the stock that has bought today can be sold only on the next trading day, which is called the T+1 trading rule. If we want to run the HFT strategy in

the Chinese stock market, there is only one way, and that we have bought stock positions. Short positions are usually hedged with stock index futures, such as IC, IF, and IH. Thus, the entire portfolio has no relationship with the fluctuation of the market. In this stock portfolio, we can run the HFT (also called T+0) strategy on the stock positions. While there are some limitations to applying the T+0 strategy, 1) the stock position that can be short is limited, and 2) the stock position must be kept the same as the initial position at the end of the trading day.

As in [1], we also transform the decision-making problem of the trading strategy in the Chinese stock market as reinforcement learning problem. In the HFT stock framework, a supervised learning model is trained to predict the price trend of each stock, and all stocks use the same model. For each stock, there is only one parameter to control the threshold of the model prediction value, if this parameter becomes smaller, the more signals will be generated by the trading strategy, meanwhile, the larger the parameter, the fewer the signals. The trading strategy determines the price and volume of the order based on the threshold of the model prediction value and limit order book.

In the Chinese stock trading system, we use the same trading policy for all stocks, whereas the threshold of the model prediction value for each stock is different. Different stocks have different market data distributions, thus, the optimal threshold of the model prediction value can yield more profit. Using the supervised learning model to provide a prediction value, a complete trading policy can be determined by a set of thresholds. To this end, we propose a reinforcement learning (RL) framework for learning an automatic and intelligent high-frequency trading strategy.

In the RL framework, the action is that a set of all stock thresholds in the portfolio, and the state is the current position and market data of each stock. Unlike the work [1], the action space is large, because there are N stocks in the stock portfolio, and the value of N is usually approximately 200. We proposed a dynamic action set approach to reduce the action space by using historical trading data for each trading tick. For the design of the reward function, we can not only consider the long and short expectations of the trading order, since the Chinese stock applies T+1 trading rules with the above limits. We should also consider the limits of the short position in the design of the reward function, particularly the limited position that can be short.

In the experimental part, we designed abundant comparison experiments to evaluate our reinforcement learning framework on the tick level market data of the Chinese stock market, which included three parts, 1) the supervised learning model for predicting stock's price trend, 2) the reward function, and 3) the entire HFT framework for trading strategy. Model expectation and accuracy are used to evaluate the price trend model, while the evaluation metrics of the reward function and trading strategy are profit and the commission, respectively. The results of the above experiments demonstrate the competitive performance of our

reinforcement learning framework for high-frequency trading in the Chinese stock market.

The main contributions of our work as follows:

1) We transform the high-frequency trading strategy optimization problem in the Chinese stock market using the T+1 rules as a reinforcement learning problem.

2) We propose a rolling training approach for the price trend prediction model, which can generate an adaptive model for new market conditions.

3) We develop a novel method to consider the limits of T+1 trading rules, which considers the constraint of limited short position into the design of the reward function.

4) We propose a novel approach to dynamically reduce the action space for each trading tick.

5) We deploy the high-frequency trading framework into a production level system and greatly increase the profit.

In the following section, we describe the organization of this paper. Section 2 describes the related works on high-frequency trading and reinforcement learning. Section 3 introduces the details of our reinforcement learning framework for high-frequency trading in the Chinese stock market, including the supervised learning model and its rolling training method, learning algorithm for the reward function, and dynamic action space reduction approach. The details of the experimental information and results are described in Section 4. Finally, Section 5 concludes the paper.

II. RELATED WORK

We review the related work in the following aspects.

A. HIGH-FREQUENCY TRADING

High-frequency trading (HFT) [2] is a method of using algorithms to send orders to exchange automatically with a very short holding time. It can be widely used by any secondary market with the T+0 trading rules, meaning that market players can buy and sell stocks or futures within one day. The limit order book (LOB) plays the most significant roles in HFT, which represents a collection of buyers and sellers, ordered by price and time. The price buyers are prepared to buy is called the *bidprice*, and the highest *bidprice* is called *bestbidprice*. Similarly, the lowest *askprice* is called *bestaskprice*. There are two types strategy are commonly used in HFT, aggressive strategy and market-making strategy. The aggressive strategy aims at causing rapid up and down of securities price movements, which is relied on technical indicators or supervised learning model to provide prediction value for price trend. Market-making strategies help to increase the liquidity in the market, and reduce the price volatility which leads to fair pricing of the asset.

Supervised learning methods are widely used in financial forecasting because of the high quality limit order book and market data, Researchers have formulated the price trend prediction problem as a regression task, and a set of classical machine learning algorithms are used for the regression tasks, such as linear regression [3], LASSO [4], elastic net [5], random forest [6], decision tree [7], support vector machine

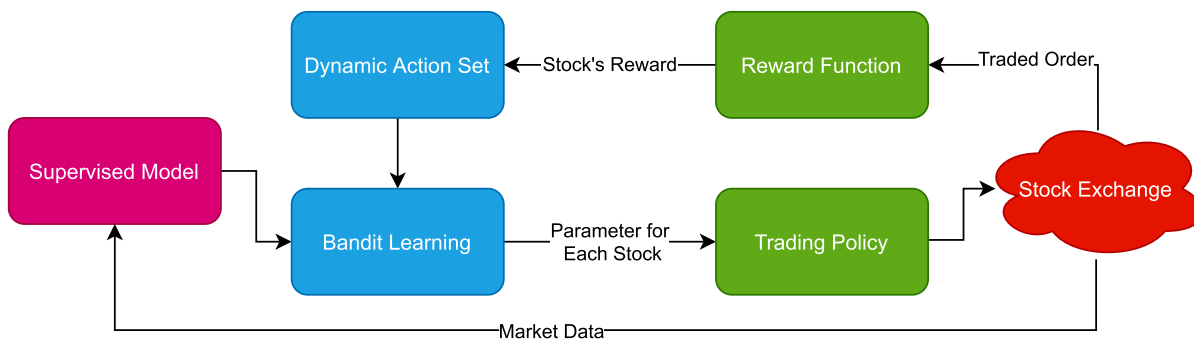


FIGURE 1. The proposed reinforcement learning framework for high-frequency trading in the Chinese Stock Market, with the following modules: 1) Supervised model (SM), 2) Bandit learning (BL), 3) Dynamic action set (DAS), 4) Reward function (RF), 5) Trading policy (TP), 6) Stock exchange (SE). SM receives market data from SE, and DAS receives all stock rewards which are computed by RF based on all traded orders. BL receives prediction value from SM and action set from DAS, then BL outputs the optimal action by hyperparameter optimization algorithm to TP. TP outputs order based on the optimal action to the stock exchange.

(SVM) [8] and LightGBM [9]. These non-linear algorithms usually outperform than linear models because they can learn the non-linear relationships between different features. With the development of deep learning models for computer vision, natural language, and speech recognition, some researchers have attempted to learn hidden relationship from features using deep neural networks (DNN) [10], recurrent neural networks (RNN) [11], long short-term memory (LSTM) [12], convolutional neural network (CNN). While at the production level of high-frequency trading, the fatal shortcoming of deep learning models is the time cost of prediction, which is usually millisecond, whereas the cost time of the high-frequency trading strategy is only 10 microseconds.

B. REINFORCEMENT LEARNING FOR HIGH-FREQUENCY TRADING

With the development of reinforcement learning (RL) in Robot Navigation and Game Playing [13] and even recently in chip design [14] and poster design [15], more and more researchers try to apply the RL algorithm into different domain, especially high-frequency trading. The work [16] is applied to automated financial trading programs. Most previous work focus on the stock market [17], [18], such as forecasting price fluctuations of stock market [19] and devising optimal stock trading strategies [20]. A set of methods dedicated to the study of financial portfolio optimization [21]. This work directly considers to learn an automatic and intelligent trading strategy that results in large-scale action space (e.g., considering each price or volume as an action), which is a hard-achieve-goals since the financial market changes rapidly. While in this paper we propose to simplify the task by optimizing the threshold in trading strategy, which can reduce the action space.

Inverse Reinforcement Learning (IRL) learns to extract the reward function given the observed behavior of an expert. The initial work on IRL was done by [22], which solved the inverse reinforcement learning problem for moderate-sized discrete and continuous domains. Using a probabilistic model

of a stochastic expert with a GP before reward values, the algorithm [23] presented can recover both a reward function and the hyper-parameters of a kernel function that describes the structure of the reward.

The optimization of model hyper-parameters has gradually evolved into an important research direction [24]. Specifically, in the field of financial transactions, financial trading programs contains a lot of control parameters, which are regarded as hyper-parameters of trading models. [25] discussed in detail how the parameterization choices are made according to the available historical data and the parameters are tuned to achieve optimal performance.

III. THE PROPOSED APPROACH

The proposed reinforcement learning framework for high-frequency trading in the Chinese stock market is shown in Figure 1, which consists of six modules, 1) SM (Supervised Model): a supervised model to predict stock price trend, 2) BL (Bandit Learning): bandit learning algorithm for selecting the optimal action, 3) RF (Reward Function): a reward function of action learned by the inverse reinforcement learning, 4) DAS (Dynamic Action Set): a dynamic algorithm for action set reduction, 5) TP (Trading Policy): trading policy to send the order to stock exchange, 6) SE (Stock Exchange): learning environment for high-frequency trading framework, stock exchange, while back-testing system can help us learn on history data. Based on current market data, the SM module provides the prediction value for all stocks, the BL module determines the threshold of the model prediction value for each stock, and the TP module sends all stocks' trading order to stock exchange by the threshold. If there are traded orders from stock exchange, RF module can compute the reward of each order, and can compute the reward of action based on each order's reward, then DAS can reduce the action space based on action's reward and market data. Finally, BL module can select the optimal action from above action space and determine the threshold of each stock.

The detailed information of above six components are demonstrated as follows.

A. PROBLEM DEFINITION

The aim of the high-frequency trading strategy is to send the proper order to the stock exchange, while in this study, the problem is to determine the proper threshold of the prediction value for each stock in the portfolio. The prediction value is provided by the supervised model, which is trained on historical data. If the threshold of prediction value is determined, trading strategy can give the price and volume of order based on prediction value and current stock’s state. As mentioned above, the problem can be treat as a reinforcement learning problem, where the action is the threshold of each stock.

Suppose p_i is the threshold of the i -th stock, and at each trading tick, we should determine all the values of p_i , where $i \in \{1, \dots, N\}$, and N is the number of stocks in the portfolio, which is usually approximately 200. Thus the action a_j is a threshold set $\{p_1, p_2, \dots, p_N\}$, which $j \in \{1, \dots, K\}$, K is the number of action in action space A . The problem is selecting the optimal action a_* from action space at each trading tick. Table 1 shows notation in this paper.

To summarize, the input of our problem is 1) market data, including tick data, transaction data, and order data, 2) trading strategies, which are written by expert experiences, including active strategy and market making strategy. 3) hyper-parameters, here are the threshold of prediction value of each stock. And the output is the optimal hyper-parameter at each tick. In addition, the constraint of the problem is that 1) limited position which can short, 2) keep the position as the fixed initial position.

$$\begin{aligned} \max_a \sum_{o \in O} \sum_{1 < i < M} w_i * E(o_i) \quad (1) \\ s.t. \quad \begin{cases} a = \{p_1, p_2, p_3, \dots, p_N\} \\ p_i \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\} \\ t_i \in \{10, 30, 90, 270, 810, \dots\} \\ w_i \in [0, 1] \\ \sum_{1 < i < M} w_i = 1 \end{cases} \end{aligned}$$

where $E(a)$ is the cumulative expectation of action a , N stands for the number of stock in portfolio, and M is the total period number for the order expectation.

B. SUPERVISED MODEL FOR PRICE TREND PREDICTION

In this section, we propose a novel approach for price trend prediction. The change rate of midprice (the average of best ask price and best bid price) is the label 2 of supervised learning model.

$$y = \frac{m^{t'+T}}{m^{t'}} - 1, \quad (2)$$

In contrast to CNN or LSTM algorithms, we compressed the history information into the current tick at the stage of X design. Table 2 shows detailed information about feature set,

which includes the abundant historical information of market data, such as spread of limit order book, the difference of mean average price and last price, the speed of price change, the speed of trade volume change, the accelerated speed of price change, the volatility of last price, the volatility of trade volume, etc. Then we use Gradient Boost Regression Tree (GBRT) [26], [27] algorithm to train the price trend model. And we use the Huber loss function 3,

$$Loss(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{for } |y - f(x)| < \sigma \\ \sigma * (|y - f(x)| - \frac{1}{2}\sigma), & \text{otherwise} \end{cases} \quad (3)$$

since huber loss function has better performance on overall data sample than MSE loss function, which performs better on extreme data samples.

C. ROLLING TRAINING FOR PRICE TREND PREDICTION MODEL

Even though we trained the price trend model on a huge training data set, usually three-month tick data, the effectiveness of the price trend model will decrease along with time, since the distribution of market data changes along with time. To address this problem, we deploy a rolling training approach for the price trend model. The model will be trained on the newest three-month data every month. If the expectation and accuracy of a new model are better than the old model, we can deploy a new model to a production system. If not, we can ensemble the prediction value of these two models and compare the expectation and accuracy, then select the better model.

D. EXPECTATION-BASED REWARD FUNCTION

The proper reward functional form can provide the ability to accurately evaluate the action in our reinforcement learning framework for each trading tick. We can use an experimental reward function via domain knowledge, such as the mean expectation of all traded orders in a fixed period. However the market data’s distribution changes rapidly in the stock market, and the reward function based on domain knowledge may become invalid after a short time. We propose a novel method based on inverse reinforcement learning to learn the optimal form of the reward function $R(a)$, which a represents for the action.

Now, we introduce how we compute the expectation of one order. For example, our trading framework sends an order $o(\text{price}, \text{volume}, \text{direction})$ with a price of 100.12, volume of 1000, and direction buy to the stock exchange. For example, if one order is traded at 14:10:30 at a price of 100, and after 20 ticks, the time is 14:11:30 while the price is 102, thus $E(T = 20) = 102/100 - 1 = 0.02$.

$$E(T) = \frac{m^{t'+T}}{m^{t'}} - 1, \quad (4)$$

where T is the number of ticks from the current tick to the future tick. In general, we will consider the short period and

TABLE 1. Notations set.

Notations	Description
P	Trading policy of trading agent
p_i	Parameter of threshold for $i - th$ stock
A	Action space
S	State
o	Order
O	Order Set
E	Order expectation: change rate of traded tick midprice and future midprice
T	Tick period
t	Minimum time unit
t^t	Tick number from the order traded tick
m_{price}	Middle price, means $(A_{price} + B_{price})/2$
A_{price}	Optimal sell price in limit order book
B_{price}	Optimal buy price in limit order book
R	Reward function
D	Data Set

TABLE 2. Important features of supervised learning model for price trend prediction.

Feature	Description
$AskPrice1 - BidPrice1$	Spread of current limit order book
$AskVolume1 - BidVolume1$	Difference of level-1 order volume
$sign>LastPrice) * Volume$	BSPDeal, larger value means strong price trend
$Diff>LastPrice)/Volume$	Larger value means that small volume can make large price change
$AvgPrice - AskPrice1, AvgPrice - BidPrice1$	The difference of trade average price with level-1 price
$ai * AskVolumei - bi * BidVolumei$	Balance of buying side and selling side power
$LastPrice - HighPrice(T)$	Difference of LastPrice with the highest price at past time
$LastPrice - LowPrice(T)$	Difference of LastPrice with the lowest price at past time
$LastPrice - MeanPrice(T)$	Difference of LastPrice with the mean price at past time
$Std>LastPrice)$	Volatility of the price
$StdVolume)$	Volatility of the trade volume
$Speed>LastPrice)$	Change speed of the price
$AccSpeed>LastPrice)$	Accelerated speed of the price

long period expectation together, and to evaluate the overall order signal. Here, we take the linear function to combine the different period expectation together, the reward function is as follows:

$$E(o) = \sum w_i * E_{T_i}, T_i \in \{10, 30, 90, \dots\}, \tag{5}$$

where $\sum w_i = 1, i \in \{1, 2, \dots, K\}$.

Because $T + 1$ trading rules are used in the Chinese stock market, thus we can apply $T0$ trading only if we have the yesterday's position of this stock. This leads to stocks having a limited short position on the next day. A constraint item is added in Equation 5 to 1) control the speed of trading in one day, and 2) maintain the same position at the end of the trading day.

$$E(o) = w * (1 - \frac{|CP - DP|}{DP}) + \sum w_i * E_{T_i}, T_i \in \{10, 30, 90, \dots\}, \tag{6}$$

where CP is the current position of the stock, DP is the target position of the stock.

To use the inverse reinforcement learning for learning the parameters of reward function, the reward function can be

written as $E(o) = w * u$, where u is the feature expectation,

$$u = [(1 - \frac{|CP - DP|}{DP}), E_{10}, E_{30}, E_{90}, \dots]^T \tag{7}$$

which E_{10} is followed by equation 5, and $T = 10$.

Algorithm 1 shows the detailed step for parameter learning of the reward function, which aims to find an optimal reward function to minimize the difference between reward function with the best reward function $E(w) - E(w^*)$. In addition, we describe the convergence analysis of the algorithm 1 as follows, From the equation (8), we can obtain the specific formulation of $error_j$ which is mentioned in algorithm 1. We can observe that the loss function $error_j$ is a linear function of the parameter w_j , and the Stochastic Gradient Descent (SGD) optimization method can be used to achieve the convergence point, which means $error_j < \epsilon$.

$$\begin{aligned} error_j &= E(w_j) - E(w^*) \\ &= \sum_{o \in O_j} E(o; w_j) - \sum_{o \in O^*} E(o; w^*) \\ &= \sum_{o \in O_j} w_j * u_j - \sum_{o \in O^*} w^* * u^* \end{aligned} \tag{8}$$

For the most high-frequency trading strategy, the short expectation plays more significant effect on the order's

Algorithm 1 Learning Algorithm for Reward Function

Input: Training set D , including LOB data, volume and turnover of each tick;

Output: Best parameter w_* of reward function;

- 1: Analysis back-testing result on training set D , then find the best trading action of each tick, get expert policy p^* ;
- 2: init a random parameter w_0 for reward function, compute the initial feature expectation u_0 ;
- 3: optimize parameter w_j to minimize the expectation difference $error_j$ between expert policy p^* and p_j ;
- 4: if $error_j < \epsilon$, then stop and get best parameter $w^* = w_j$;
- 5: based on reward function $R(w_j)$, get current optimal policy p_{j+1} by applying parameter optimization algorithm in next section Action Selection Algorithms;
- 6: compute the feature expectation u_{j+1} ;
- 7: set $j = j + 1$, loop again from step 3;

reward. Thus the overall formulation for the reward of one order are

$$E(o) = \sum_{k=1}^K \gamma^k E_T \quad (9)$$

where $T \in \{10, 30, 90, \dots\}$, $0 < \gamma < 1$.

E. DYNAMIC ALGORITHM FOR ACTION SET GENERATION

After we have one optimal reward function for evaluating action, the action set should be determined to select the optimal action at each tick. Usually, we have about three hundred stocks in the portfolio, and for each stock, we have a specific parameter for the model prediction value of this stock, such as 1.6 for SH600519, 1.2 for SZ000568, 0.8 for SZ300750, etc. And the possible values of each parameter are $\{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$, thus the action space is 6^{200} , which is significantly larger than the action space of Go. We can not iteratively compute the reward of each action, or the trading agent will not send any orders to the stock exchange.

Here we develop a dynamic algorithm to generate an action set at each tick based on current stock position and market data, 1) narrow the value range for the parameter of a specific stock, for example, the most valuable Liquor stock SH600519, if the parameter is high, then the prediction value can not be larger than this parameter, and can not send any orders to exchange. Thus the possible values of SH600519 are $\{1.0, 1.2\}$. In addition, we can narrow the value range based on the volume and volatility on that trading day. Higher volatility means we should use the large parameter, and a lower volume means we should use a small parameter. 2) cluster the stocks (e.g. four clusters from two aspects, price, and volatility), and use the same parameter for each cluster. Since the stocks in the same cluster have a similar distribution of market data. Finally, we can narrow the action set to 2^4 at each tick. And we will generate different action set at different tick.

F. ACTION SELECTION ALGORITHM BASED ON UCB

When the optimal reward function is given by the algorithm above, the expectation of each traded order can be computed accurately, then the reward of each action in the dynamic action space can be formulated as follows,

$$E(a) = \frac{\sum E_*(o_i)}{M}, \quad (10)$$

where M is the number of traded order in a fixed trading period, o_i stands for each traded order, and $i \in \{1, 2, \dots, M\}$.

Multi-armed bandit learning algorithm provides an appropriate way to select the optimal action from action space, that is to select the best threshold of prediction value for each stock in portfolio.

$$S(a) = W(a) + C * \sqrt{2 * \frac{\ln N}{n(a)}}, \quad (11)$$

where $S(a)$ stands for action's score, $W(a)$ stands for the average reward, visited time is recorded as $n(a)$, and a constant C .

In general, RL environment can generate the reward of only one action, then update $W(a)$. While in this section, we developed a novel method to compute the reward of all action in action space together at each trading tick. We develop a precise back-testing system, and integrate it into online trading system. The back-testing system can estimate the expectation of each traded order, and act as a learning environment.

Algorithm 2 demonstrates the running process for action selection algorithm. At each tick in online trading, the algorithm will run once to get the optimal action a_t^i based on previous UCB score, then run the back-testing system to update the reward score of all other actions. After that, the algorithm stops to step into the next tick.

Algorithm 2 Action Selection Algorithm Based on Reward Enhanced UCB

Input: Action set A , Reward function $R(a)$

Output: Optimal action a_t at each trading tick t ;

- 1: Init $W(a_i) = 0.0$, $n(a_i) = 0$;
- 2: Select action a_t^i with highest UCB score $S(a)$;
- 3: Trading with action a_t^i , then record the reward of each order;
- 4: Back-testing the action a_t^i in different fixed period (e.g. 30 minutes, 60 minutes) before current tick, then record the reward $w(a_i) * \lambda_t$, there t is the period;
- 5: Update $W(a_i)$ and $n(a_i)$;
- 6: Loop again from step 2;

IV. EXPERIMENTS

Level-2 market data for the Chinese stock market consists of the dataset. In the following sections, we will introduce the data set, evaluation metrics, experimental result of baseline and our algorithm.

TABLE 3. Price trend model: expectation result of the top prediction value. The unit of expectation value is 0.001.

Period/k%	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	2	5	10
1	1.59	1.38	1.18	1.06	0.98	0.93	0.89	0.86	0.83	0.81	0.78	0.64	0.48	0.38
2	2.16	1.87	1.59	1.44	1.34	1.28	1.22	1.18	1.13	1.10	1.07	0.89	0.67	0.53
4	2.9	2.48	2.10	1.89	1.76	1.67	1.60	1.53	1.48	1.44	1.40	1.16	0.88	0.69
10	3.65	3.05	2.55	2.29	2.14	2.01	1.92	1.84	1.77	1.72	1.67	1.39	1.07	0.86
20	4.13	3.35	2.77	2.43	2.26	2.13	2.01	1.91	1.84	1.78	1.73	1.43	1.11	0.90
40	3.90	3.20	2.63	2.32	2.17	2.05	1.95	1.86	1.78	1.72	1.68	1.40	1.09	0.88
80	3.70	3.11	2.58	2.30	2.18	2.07	1.97	1.89	1.81	1.75	1.70	1.42	1.10	0.89
120	3.47	3.08	2.56	2.29	2.18	2.08	2.00	1.91	1.83	1.77	1.72	1.42	1.11	0.90
240	2.50	2.27	1.92	1.78	1.76	1.71	1.65	1.61	1.56	1.52	1.51	1.27	1.03	0.86
480	2.97	2.49	1.99	1.80	1.71	1.68	1.63	1.58	1.51	1.49	1.45	1.21	0.99	0.83

TABLE 4. Price trend model: accuracy result of the top prediction value.

Period/k%	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	2	5	10
1	0.683	0.688	0.688	0.685	0.679	0.674	0.669	0.662	0.655	0.649	0.643	0.595	0.521	0.456
2	0.721	0.723	0.726	0.727	0.726	0.725	0.722	0.717	0.713	0.709	0.704	0.667	0.604	0.545
4	0.736	0.734	0.735	0.737	0.735	0.733	0.731	0.727	0.724	0.721	0.718	0.692	0.645	0.599
10	0.705	0.697	0.694	0.692	0.691	0.687	0.685	0.681	0.679	0.676	0.673	0.656	0.629	0.601
20	0.692	0.670	0.660	0.652	0.650	0.645	0.641	0.638	0.635	0.633	0.631	0.615	0.595	0.576
40	0.659	0.642	0.628	0.620	0.615	0.612	0.608	0.605	0.601	0.598	0.596	0.582	0.564	0.551
80	0.610	0.603	0.590	0.586	0.583	0.581	0.577	0.575	0.572	0.570	0.569	0.559	0.546	0.536
120	0.589	0.588	0.575	0.570	0.568	0.566	0.564	0.561	0.558	0.557	0.555	0.546	0.535	0.528
240	0.573	0.568	0.556	0.551	0.548	0.547	0.545	0.541	0.539	0.538	0.537	0.530	0.523	0.518
480	0.556	0.550	0.542	0.538	0.537	0.536	0.534	0.533	0.530	0.530	0.528	0.522	0.516	0.513

TABLE 5. Information of example stock.

Stock	Min price unit	Recent Price	Commission	Market Value	Exchange
SH600519	0.01	2040 rmb	0.00015 + 0.001	20000G rmb	Shanghai
SZ002594	0.01	187 rmb	0.00015 + 0.001	5200G rmb	Shenzhen

A. DATASET

Level-2 depth market data, which is the finest grain of stock data, is used in our experiment. It is also called tick data, which was published from the Chinese Stock Exchange. The format was TradingDay, UpdateTime, Volume, Turnover, BidPrice1-5, AskPrice1-5, BidVolume1-5, AskVolume1-5. From the viewpoint of the dataset, we may be the first to present our novel research results on such a high-quality dataset that runs in the real trading system.

We prepare datasets from a subset of all stocks in the Chinese stock market, except for ST stock and low trading volume. This subset is selected carefully to cover most of the stock types, including high price, low price, a different industry, value stock, and growth stock. In the Chinese stock market, Level-2 depth market data are published as market data snapshot every three seconds. The information of the final dataset is 1) 200 stocks, 2) the scope of the training set is 2020.03.01 - 2020.07.31, 3) the scope of the test set is 2020.08.01 - 2020.10.30, and 4) minimal time unit is 3 s. Table 5 shows some example stocks, including the minimum price unit, recent price, market value, and so on.

B. EVALUATION METRICS OF PRICE TREND MODEL

In a practical trading system, we typically use a large prediction value, which is either, positive or negative. Thus we will focus on the evaluation results of a large prediction value.

Two metrics are proposed to evaluate the price trend model, 1) Accuracy, which checks if the sign of the prediction is the same as the sign of the label. 2) Expectation, which checks that if we send a signal to exchange using the large prediction value, we can obtain the profit with the corresponding y .

$$Acc(Topk\%) = \frac{N(\text{sign}(\text{pred}) * \text{sign}(y) > 0)}{N(Topk\%)} \quad (12)$$

where pred is the prediction value, y is the corresponding label. We also have:

$$Exp(Topk\%) = \frac{\sum_{\text{sign}(\text{pred}) * \text{sign}(y) > 0} |y|}{N(Topk\%)} \quad (13)$$

where $N(Topk\%)$ stands for the number of prediction signal at the top $k\%$, and $k \in \{0.05, 0.1, 0.2, 0.3, \dots, 1, 2, 5, 10\}$. In addition $\text{sign}(y)$ is the sign of y , and its value can be 1 or -1.

C. EVALUATION METRICS OF TRADING FRAMEWORK

Trading profit and commission are used in the design of evaluation metrics. In general, we evaluate the trading policy from two viewpoints, 1) Absolute profit $AP = p - c$, 2) Multiple $MP = \frac{p}{c}$ of profit and commission, where p is trading profit, and c is trading commission.

Usually, most researchers use absolute profit AP as the metric, while if the absolute profit of one trading strategy is the same as another, Multiple MP will give another viewpoint for evaluation. The higher MP , the better trading strategy, and it shows that this trading strategy also has more capacity.

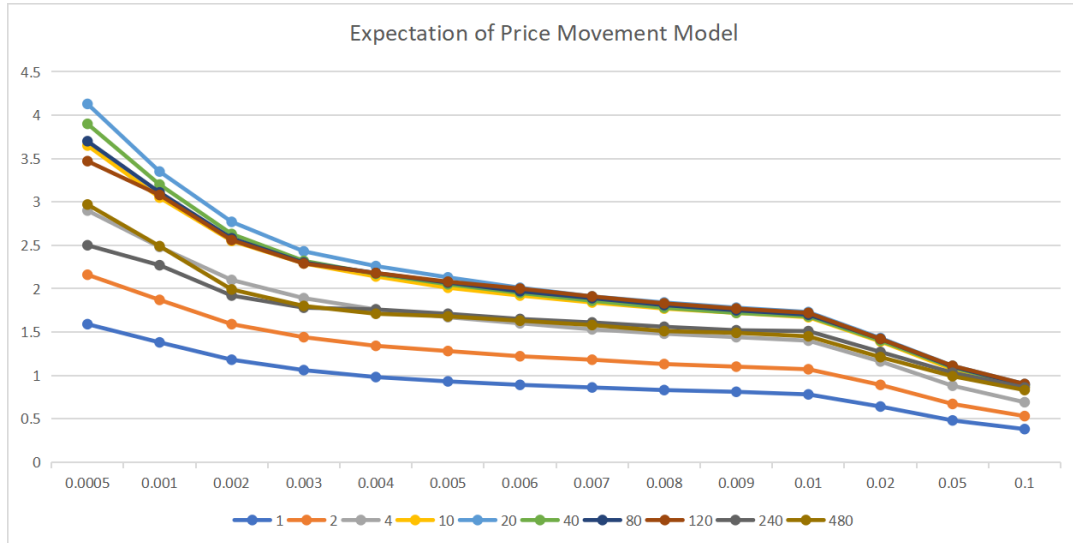


FIGURE 2. The expectation of each top ranking point of the price movement model. Different line stands for the different period with different color. X axis shows the ranking point, Y axis shows the value range of expectation, and the unit is 0.1%.

D. RESULT FOR PRICE TREND MODEL

We sorted the prediction value by its absolute value, and evaluate the accuracy and expectation for the top $k\%$ prediction value, where k takes the value from $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 5, 10\}$. In addition, we evaluate the distribution on different period label, here we use T stands for period, means that there are T ticks in each period, where T takes the value from $\{1, 2, 4, 10, 20, 40, 80, 120, 240, 480\}$. An excellent model can take high expectations on short periods, also on long periods.

Table 3 shows the expectation results for the top prediction value. From the expectation result, we observed that, 1) the larger the prediction value, the higher the expectation. This implies that the model has high monotonicity. If we want to obtain a sharper strategy, we can trade only with the highest signal. 2) The expectation distribution from the perspective of the period is peak-like, where the highest point is $T = 20$, which also be proved in figure 2. From $T = 1$ to $T = 20$, the expectation increases very quickly. While from $T = 20$ to $T = 480$, the expectation decreases slowly. In addition, we can observe that the first 1 tick can obtain a nearly 33% expectation of 20 ticks, which means that we should send the signal as soon as possible.

Table 4 shows the accuracy results of the price trend model. From the accuracy result, we can observe that, 1) The higher the prediction value, the higher the accuracy of the model prediction. 2) The accuracy distribution from the viewpoint of the period is peak-like, where the highest point is $T = 4$ and be showed in figure 3, which differs from the highest point ($T = 20$) in the expectation distribution. This means that the accuracy of $T = 20$ is lower than that of $T = 4$, but some signals from $T = 20$ are much higher than

those from $T = 4$, thus the expectation is higher. 3) Even for the period $T = 480$, which means 12 minutes, the model also has an accuracy greater than 50%.

E. RESULT FOR OUR REINFORCEMENT LEARNING FRAMEWORK OF HFT

Existing reinforcement learning work on trading strategy focuses on predicting action (determining whether to send a trading order to exchange) directly, honestly, this work usually has little effect on real trading. This author has no experience of high-frequency trading at the production level, thus we can not run any comparison experiment based on their work, although we will design some typical baselines.

Here we prepare some different types of baselines, 1) The traditional technical indicator *MACD* [28], and many quantitative agents use technical indicators to send the order signal. 2) A supervised learning model [29] with a simple taking strategy, which sends orders only depend on prediction value, and use market order type to trade. 3) The first public and complete trading strategy, Way of the Turtle, which brought huge profit for the strategy author in real trading.

We also propose some baselines by simplifying one or more modules of our reinforcement learning framework, 4) FT: Fixed threshold for each stock. 5) F(Time): Determine the threshold just by time, for example, set the threshold as 1.8 during morning trading session, and set the threshold as 1.4 during the afternoon session. 6) F(Volatility): Determine the threshold just by volatility, for example, the price of stock SZ300750 has high volatility, we can set the threshold as 2.0, and the price of stock SH600519 has low volatility, we can set the threshold as 1.4. 7) F(Time, Volatility): Determine the threshold by time and volatility.

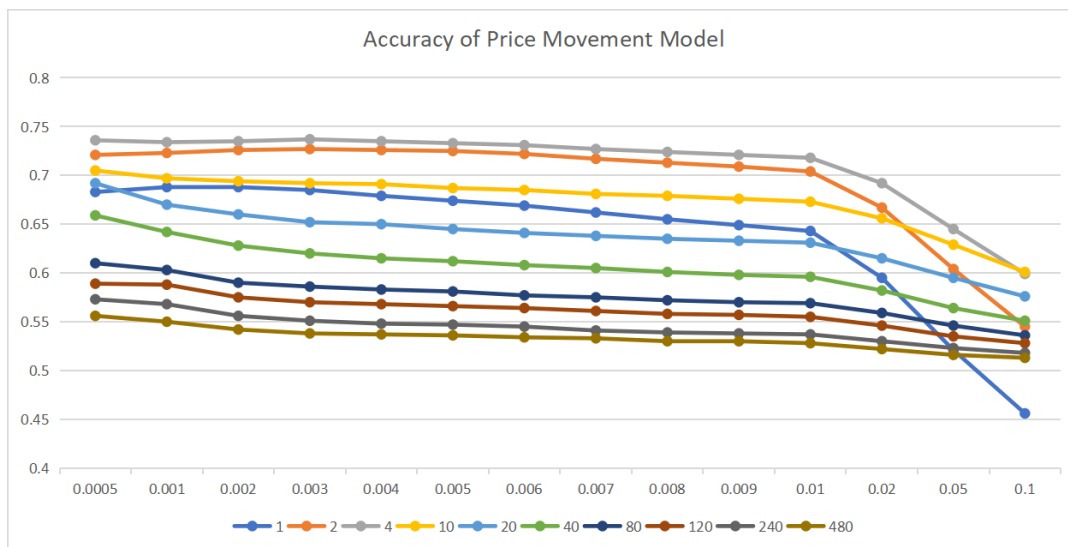


FIGURE 3. The accuracy of each top ranking point of the price movement model. Different line stands for the different period with different color. X axis shows the ranking point, Y axis shows the value range of expectation.

TABLE 6. HFTStock: performance result of proposed method.

Method	Experiment Setting	Average Profit	Average Commission
Baseline1	MACD	2235	13890
Baseline2	Supervised Learning Model	15045	13680
Baseline3	Way of the Turtle	11289	13720
Baseline4	Fixed Threshold	16384	13294
Baseline5	F(Time)	15824	13378
Baseline6	F(Volatility)	16295	12976
Baseline7	F(Time, Volatility)	16347	13143
SL-TP-UCBE	Time Priority, UCB enhanced	19356	13647
SL-IRL-UCBE	Inverse RL, UCB enhanced	23347	13131

Our proposed framework is called SL-IRL-UCBE, which refers to the supervised learning model, inverse reinforcement learning for learning the reward function, UCB enhanced selection algorithm. And we design one more baseline is called SL-TP-UCBE, which means the reward function use time priority (TP) method, and is same as the equation 9, we set the $\lambda = 0.7$. Thus the expectation of short period is more important.

When we tested the performance of these algorithms, the test set was set as the last one month dataset. Every day one result is generated, which includes profit and commission cost. In table 6, we show the average profit and average commission in the one month test set by day.

From the result of table 6, we can find that our reinforcement learning framework for high-frequency trading in the Chinese stock market, SL-IRL-UCBE achieves the most competitive performance among all these methods. SL-TP-UCBE performs better than other baselines, but the time priority method for reward function is not the optimal selection. Inverse reinforcement learning helps to obtain the optimal form of reward function. Among the baselines, the technical indicator *MACD* has the worst performance, that is because one single indicator can not adapt to the stock market now.

F. RESULT FOR DIFFERENT REWARD FUNCTION OF ORDER

We proposed several baselines to evaluate the performance of a reward function of one order. 1) $E(T=10)$: Just use only one period expectation as the reward of order, the future 10 tick after trading. 2) $E(New)$: Only one period expectation as the reward, but the reward changes at each tick, which is the expectation between the traded tick and the newest tick. 3) $E(Linear)$: Fixed parameter of different period expectation, here we set as 0.5, 0.4, 0.3, ..., which is the linear combination. Similar to the above comparison experiment, all these methods were tested on the same test set, that is, the last one month market data. Table 7 presents the results of the different reward functions.

From the comparison results, we can observe that, our proposed method for learning the form of the reward function outperforms all other methods, where SL-IRL-UCBE achieves the highest profit, which means that the reward function is more accurate for evaluating action. Baseline $E(Linear)$ performs better than the other baselines, which demonstrates that the expectation of different periods is important for order evaluation.

TABLE 7. Reward Function: performance result of proposed method.

Method	Experiment Setting	Average Profit	Average Committee
Baseline1	E(T=10)	16897	13532
Baseline2	E(New)	14694	13187
Baseline3	E(Linear)	17647	13857
SL-TP-UCBE	Time Priority, UCB enhanced	19356	13647
SL-IRL-UCBE	Inverse RL, UCB enhanced	23347	13131

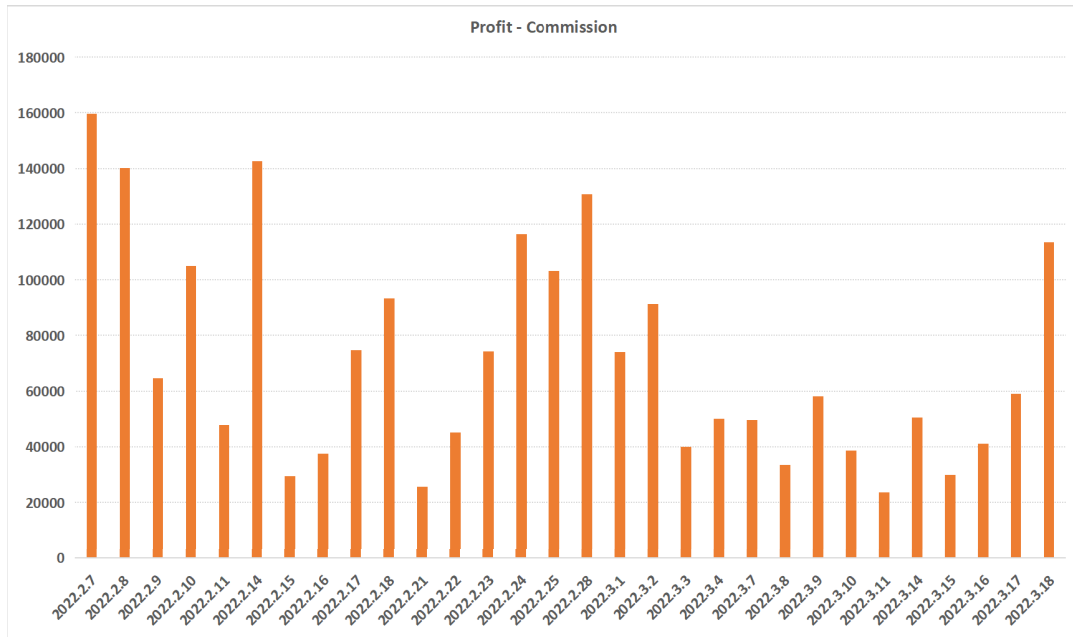


FIGURE 4. Absolute profit of the SL-IRL-UCBE approach in real trading.

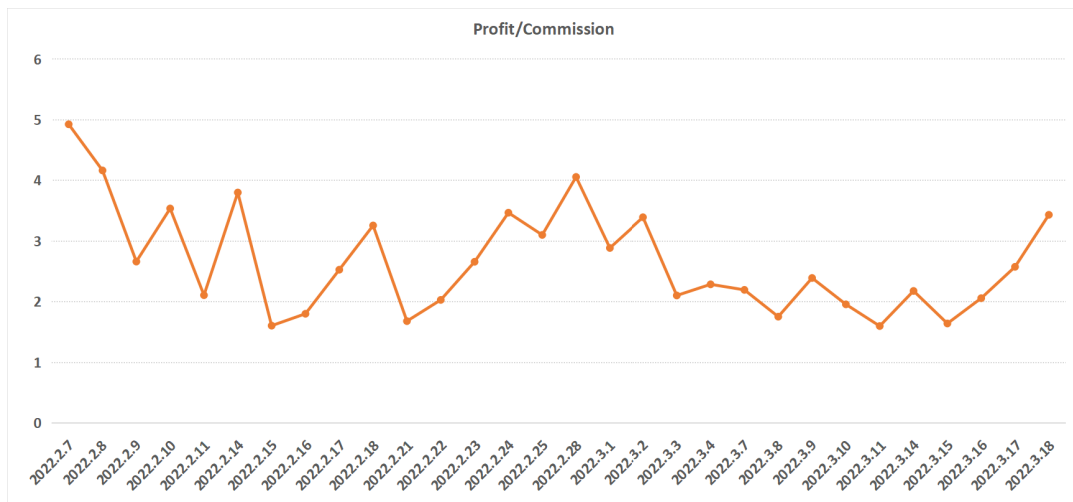


FIGURE 5. Profit multiplier of the SL-IRL-UCBE approach in real trading.

G. STATISTIC RESULT OF REAL TRADING IN CHINESE STOCK MARKET

We run the T0-IRL-UCBS approach on the real stock account in the Chinese stock market from Feb 7th, 2022 to Mar 18th, 2022, where the stock position value is about 50 billion RMB,

and there are 300 stocks in the portfolio. Figure 4 shows the absolute daily profit, the difference of profit and commission. Figure 5 shows the earning multiplier for each trading day, the division of profit and commission. The result of real trading shows the competitive performance in the T0 strategy of the

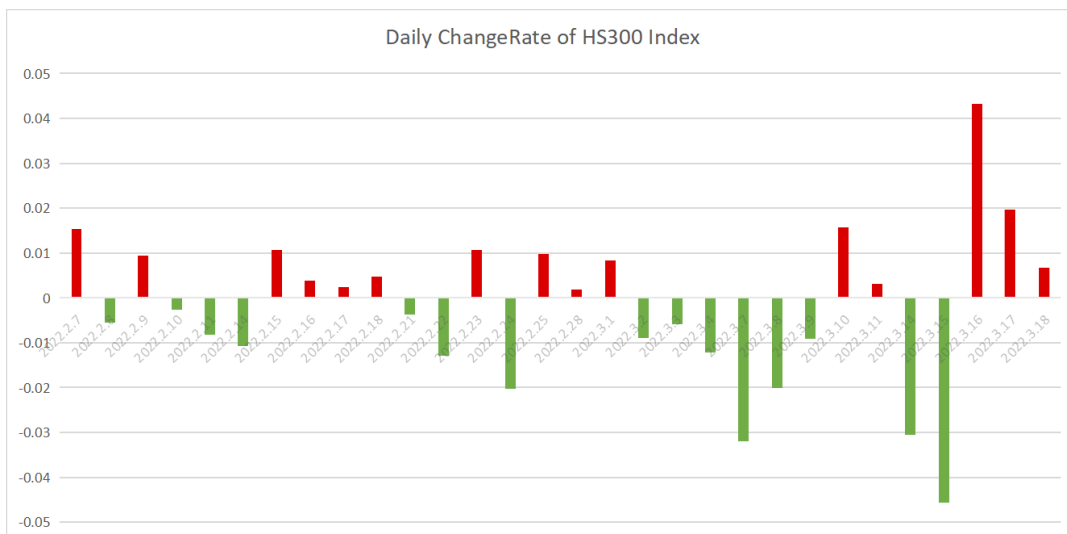


FIGURE 6. Daily change rate of HS300 stock index. red color means the up trend, while green means the down trend.

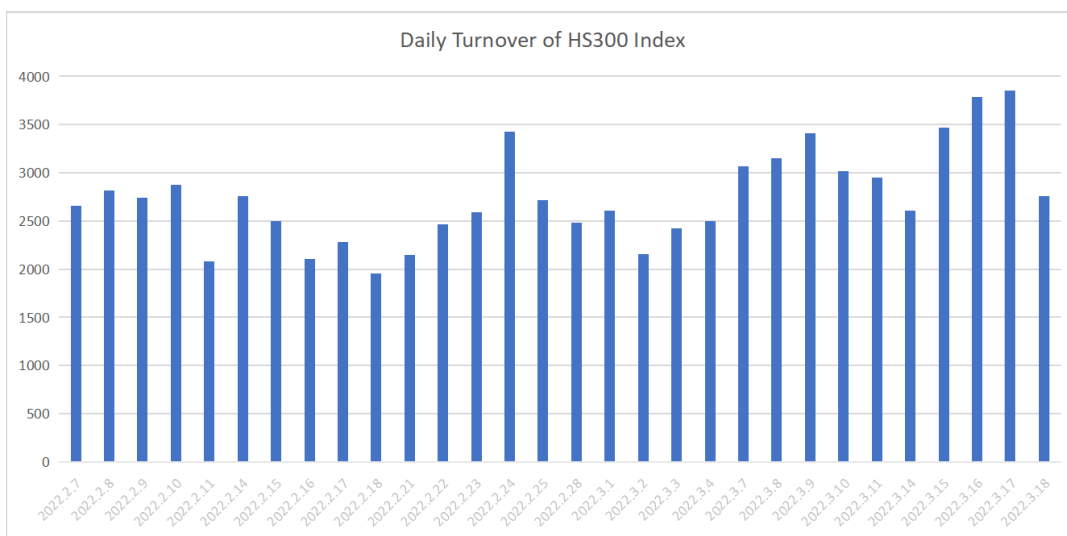


FIGURE 7. Daily turnover rate of HS300 stock index. The unit of turnover value is hundred million.

Chinese stock market. In addition, we can find that the result in real trading from 2022.2.7 to 2022.3.18 is much better than the result at the test set, especially at Feb 7th, we achieve about five times profit of transaction commission. There are two reasons for this: 1) **High volatility**: The Chinese stock market is much more larger fluctuation than before, and the HFT strategy usually has perfect performance in the market with high volatility. In the figure 6 and 7, the value and volume of the transaction of HS300 stock index are showed the high relationship between volatility and transaction volume. In the trading period from from Feb 7th, 2022 to Mar 18th, 2022, the value of transaction is higher than the market with low volatility, and the percent of transaction value can achieve about 70% of the value of the whole portfolio. Thus the daily transaction value is about 35 billion RMB on the single side,

and 70 billion RMB on the double side (buy and sell). Due to the transaction fee rate is 0.07% in Chinese stock market on the single side, the daily transaction commission is about 49 thousands. 2) **Intelligent Algorithm**: Our RL framework has much more optimization space in such volatile markets, and other market participants have no such better adaptability.

V. CONCLUSION

We have presented a novel and effective framework for price trend prediction model and parameter online optimization of a high-frequency trading strategy in the Chinese stock market. An abundant feature set is designed for price trend model training, and a rolling trading method is applied for the self-adaptation of the price trend model. Inverse reinforcement learning based algorithm is proposed for the parameter

learning of reward function, and the constraint item of Chinese stock $T + 1$ rules is considered in the equation of reward function. In addition, a precise back-testing system was developed to evaluate the reward for each action during real-time trading. All these experiments result on the subset of all stocks show that our proposed algorithm achieves competitive performance on Chinese Stock Market Data. Finally, we run our proposed framework at the production level to evaluate the effectiveness in real trading. Daily profit shows the promised profitability. In the future, we will upgrade the framework to make it more suitable for use in any secondary market.

VI. OUTLOOK FOR FUTURE WORK

There are many potential directions for future work, as the high-frequency trading with advanced machine learning is still relatively in its early stage, especially seeing the fast development of machine learning.

First, we aim to explore more advanced temporal models both for time series learning, in terms of anomaly detection [30], [31] and forecasting [32], as well as continuous time event sequence modeling, especially for the so-called temporal point process (TPP) [33]. Specifically, the TPP model can either be used for relation mining [34] also for prediction [35], [36], rule mining [37] and clustering [38], [39].

Another promising direction is how to incorporate more information into the decision making pipeline, which can be encoded by graph neural networks (GNNs) [40] or other more efficient embedding methods [41], [42]. Meanwhile, machine learning for combinatorial optimization is also worth further study. One immediate way is to incorporate the knowledge graph [43].

Finally, putting the decision making in a multi-agent system perspective, it would be also interesting to consider the relation and constraint among the agents for trading, whereby graph learning [44], and especially graph matching [45] can be a potential tool to advance this topic.

REFERENCES

- [1] W. Zhang, L. Wang, L. Xie, K. Feng, and X. Liu, "TradeBot: Bandit learning for hyper-parameters optimization of high frequency trading strategy," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108490.
- [2] R. Savani, "High-frequency trading: The faster, the better?" *IEEE Intell. Syst.*, vol. 27, no. 4, pp. 70–73, Jul./Aug. 2012.
- [3] C.-H. Hsieh, B. R. Barmish, and J. A. Gubner, "The impact of execution delay on kelly-based stock trading: high-frequency versus buy and hold," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Nice, France, Dec. 2019, pp. 2580–2585.
- [4] J. Liu, Q. Fu, and O. Yilmaz, "Prediction of high frequency trading financial data using stacked LSTMs for algorithmic trading," in *Fuzzy Systems and Data Mining (Frontiers in Artificial Intelligence and Applications)*, vol. 320. Amsterdam, The Netherlands: IOS Press, 2019, pp. 1064–1070.
- [5] H. Han, J. Teng, J. Xia, Y. Wang, Z. Guo, and D. Li, "Locally linear embedding for high-frequency trading marker discovery," in *Proc. IDMB*, in Communications in Computer and Information Science, vol. 1099. Springer, 2019, pp. 3–17.
- [6] G. P. M. Virgilio, "Absolute vs. relative speed in high-frequency trading," *Algorithmic Finance*, vol. 7, nos. 3–4, pp. 71–86, Apr. 2019.
- [7] A.-I. Stan, "Computational speed and high-frequency trading profitability: An ecological perspective," *Electron. Markets*, vol. 28, no. 3, pp. 381–395, Aug. 2018.
- [8] L. Delaney, "Investment in high-frequency trading technology: A real options approach," *Eur. J. Oper. Res.*, vol. 270, no. 1, pp. 375–385, Oct. 2018.
- [9] B. Crawford, R. Soto, M. A. S. Martín, H. de la Fuente Mella, C. Castro, and F. Paredes, "Automatic high-frequency trading: An application to emerging Chilean stock market," *Sci. Program.*, vol. 2018, pp. 8721246:1–8721246:12, Sep. 2018.
- [10] W. Currie, J. J. M. Seddon, R. Cooper, and B. V. Vliet, "Theories for analysing innovation and technology in emerging financial markets: The case of algorithmic and high frequency trading," in *Proc. AMCIS*. Atlanta, GA, USA: Association for Information Systems, 2018.
- [11] N. Reznik and L. Pankratova, "High-frequency trade as a component of algorithmic trading: Market consequences," in *Proc. CEUR Workshop ICT*, vol. 2104, 2018, pp. 73–83. [Online]. Available: <https://CEUR-WS.org>
- [12] S. A. Alves, W. Caarls, and P. M. V. Lima, "Weightless neural network for high frequency trading," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–7.
- [13] Y. Hua, X. Wang, B. Jin, W. Li, J. Yan, X. He, and H. Zha, "HMRL: Hypermeta learning for sparse reward reinforcement learning problem," in *Proc. SIGKDD*, Aug. 2021, pp. 637–645.
- [14] R. Chen and J. Yan, "On joint learning for solving placement and routing in chip design," in *Proc. NeurIPS*, 2021, pp. 16508–16519.
- [15] G. Wang, Z. Qin, J. Yan, and L. Jiang, "Learning to select elements for graphic design," in *Proc. ICMR*, Jun. 2020, pp. 91–99.
- [16] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," *CoRR*, vol. abs/1706.10059, pp. 1–31, Jun. 2017.
- [17] Z. Xiong, X.-Y. Liu, S. Zhong, H. Yang, and A. Walid, "Practical deep reinforcement learning approach for stock trading," *CoRR*, vol. abs/1811.07522, pp. 1–7, Nov. 2018.
- [18] X. Liang, D. Cheng, F. Yang, Y. Luo, W. Qian, and A. Zhou, "F-HMTC: Detecting financial events for investment decisions based on neural hierarchical multi-label text classification," in *Proc. IJCAI*, Jul. 2020, pp. 4490–4496.
- [19] J. W. Lee, "Stock price prediction using reinforcement learning," in *Proc. ISIE*, vol. 1, 2001, pp. 690–695.
- [20] J. W. Lee, J. Park, O. Jangmin, J. Lee, and E. Hong, "A multiagent approach to Q-learning for daily stock trading," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 37, no. 6, pp. 864–877, Nov. 2007.
- [21] Y. Sato, "Model-free reinforcement learning for financial portfolios: A brief survey," 2019, *arXiv:1904.04973*.
- [22] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. ICML*, vol. 1, 2000, pp. 663–670.
- [23] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with Gaussian processes," in *Proc. NIPS*, 2011, pp. 19–27.
- [24] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyperparameter optimization," in *Proc. NIPS*, 2011, pp. 2546–2554.
- [25] O. V. Pictet, M. M. Dacorogna, B. Chopard, M. Oussaidene, R. Schirru, and M. Tomassini, "Using genetic algorithms for robust optimization in financial applications," *Neural Netw. World*, vol. 5, no. 4, pp. 573–587, 1995.
- [26] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
- [27] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. NIPS*, 2017, pp. 3146–3154.
- [28] G. Durantin, S. Scannella, T. Gateau, A. Delorme, and F. Dehais, "Moving average convergence divergence filter preprocessing for real-time event-related peak activity onset detection : Application to fNIRS signals," in *Proc. EMBC*, Aug. 2014, pp. 2107–2110.
- [29] E. A. Gerlein, M. McGinnity, A. Belatreche, and S. Coleman, "Evaluating machine learning classification for financial trading: An empirical approach," *Expert Syst. Appl.*, vol. 54, pp. 193–207, Jul. 2016.
- [30] L. Li, J. Yan, H. Wang, and Y. Jin, "Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1177–1191, Mar. 2020.
- [31] L. Li, J. Yan, Q. Wen, Y. Jin, and X. Yang, "Learning robust deep state space for unsupervised anomaly detection in contaminated time-series," *IEEE Trans. Knowl. Data Eng.*, early access, May 12, 2022, doi: [10.1109/TKDE.2022.3171562](https://doi.org/10.1109/TKDE.2022.3171562).
- [32] L. Li, J. Zhang, J. Yan, Y. Jin, Y. Zhang, Y. Duan, and G. Tian, "Synergetic learning of heterogeneous temporal sequences for multi-horizon probabilistic forecasting," in *Proc. AAAI*, 2021, pp. 8420–8428.

[33] J. Yan, H. Xu, and L. Li, "Modeling and applications for temporal point processes," in *Proc. SIGKDD*, Jul. 2019, pp. 3227–3228.

[34] Y. Zhang and J. Yan, "Neural relation inference for multi-dimensional temporal point processes via message passing graph," in *Proc. IJCAI*, Aug. 2021, pp. 3406–3412.

[35] J. Yan, Y. Wang, K. Zhou, J. Huang, C. Tian, H. Zha, and W. Dong, "Towards effective prioritizing water pipe replacement and rehabilitation," in *Proc. IJCAI*, 2013, pp. 2931–2937.

[36] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, "Learning time series associated event sequences with recurrent point process networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3124–3136, Oct. 2019.

[37] S. Li, M. Feng, L. Wang, A. Essofi, Y. Cao, J. Yan, and L. Song, "Explaining point processes by learning interpretable temporal logic rules," in *Proc. ICLR*, 2022, pp. 1–23.

[38] W. Wu, J. Yan, X. Yang, and H. Zha, "Discovering temporal patterns for event sequence clustering via policy mixture model," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 573–586, Feb. 2022.

[39] Y. Zhang, J. Yan, X. Zhang, J. Zhou, and X. Yang, "Learning mixture of neural temporal point processes for multi-dimensional event sequence clustering," in *Proc. IJCAI*, Jul. 2022.

[40] H. Zhang, Q. Wu, J. Yan, D. Wipf, and P. S. Yu, "From canonical correlation analysis to self-supervised graph neural networks," in *Proc. NeurIPS*, 2021, pp. 76–89.

[41] X. Du, J. Yan, R. Zhang, and H. Zha, "Cross-network skip-gram embedding for joint network alignment and link prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1080–1095, Mar. 2022.

[42] H. Xiong and J. Yan, "BTWalk: Branching tree random walk for multi-order structured network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3611–3628, Aug. 2022.

[43] S. Yang, J. Tian, H. Zhang, J. Yan, H. He, and Y. Jin, "TransMS: Knowledge graph embedding for complex relations by multidirectional semantics," in *Proc. IJCAI*, Aug. 2019, pp. 1935–1942.

[44] Q. Wu, C. Yang, and J. Yan, "Towards open-world feature extrapolation: An inductive graph learning approach," in *Proc. NeurIPS*, 2021, pp. 19435–19447.

[45] R. Wang, J. Yan, and X. Yang, "Combinatorial learning of robust deep graph matching: An embedding based approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 29, 2020, doi: 10.1109/TPAMI.2020.3005590.



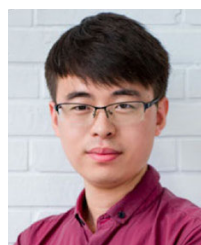
TAO YIN received the B.E. and M.S. degrees from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2011 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. Before that, he worked as the Director and the Founder of a hedge fund for high-frequency trading. His current research interests include quantitative trading and data mining.



YUNAN ZHAO received the B.E. and M.E. degrees in the major of automation from Shanghai Jiao Tong University, Shanghai, China, in 2013 and 2016, respectively. He is currently a Staff Researcher with MYbank, Ant Group, Shanghai. Once, he was an Engineer with China Merchants Bank, Shanghai. His research interests include machine learning and computer vision.



BING HAN is currently a Senior Staff Engineer and the Head of the Intelligent Engine Department, MYbank, Ant Group. Her research interests include machine learning and data intelligence, especially in recommender systems and financial technology.



WEIPENG ZHANG received the bachelor's and master's degrees from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2012 and 2015, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. Before that, he worked as the Manager of hedge fund to integrate machine learning and reinforcement learning algorithm into high frequency trading. He is currently focusing

on the research and application of reinforcement learning algorithm in quantitative trading.



HUANXI LIU was born in Hunan, China, in 1982. He received the master's and Doctor degrees in pattern recognition and intelligent system from Shanghai Jiao Tong University, China, in 2007 and 2010, respectively. He is currently a Senior Engineer with Shanghai Jiao Tong University. His research interests include pattern recognition, computer vision, and machine learning.

...