IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 6, 2020, accepted March 19, 2020, date of publication March 23, 2020, date of current version April 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982662

An Adaptive Financial Trading System Using Deep Reinforcement Learning With Candlestick Decomposing Features

DING FENGQIAN[1] **AND LUO CHAO**[1,2,3]

[1]School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China
[2]Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology, Jinan 250014, China
[3]Institute of Data Science and Technology, Shandong Normal University, Jinan 250014, China

Corresponding author: Luo Chao (cluo79@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61402267 and in part by the Shandong Provincial Natural Science Foundation under Grant ZR2014FQ004.

ABSTRACT When applying artificial intelligence technology to quantitative trading, high noise and unpredictability of market environment are the first practical problems to be considered. Therefore, how to select the learning features of the market based on rapidly changing financial data is particularly important. In this paper, the real time financial data are first processed by K-line theory, which uses candlesticks as a generalization of price movements over a period of time, so this process can play the role of de-noising. Then, the candlesticks are decomposed into different subparts by mean of a specified spatio-temporal relationship, based on which cluster analysis of the subparts to get the learning features. Further, the learning features that are clustered by the above K-lines are put into the model, and the online adaptive control of the parameters in the unknown environment is realized by the deep reinforcement learning method, so as to realize the high frequency transaction strategy. In order to verify the performance of the model, the data on different financial derivatives transactions such as stocks, financial futures and commodity futures are used. The proposal approach is compared with other methods which are based on price, fuzzified price and K-lines for features learning. In order to verify the accuracy of the proposal approach, prediction-based methods such as recurrent neural network and fuzzy neural network are used for comparison. Experimental results show that the proposed method has higher robustness and prediction accuracy.

INDEX TERMS Deep reinforcement learning, K-line decomposing, clustering, trading system.

I. INTRODUCTION

It is a common phenomenon in many developed countries to realize automated trading through computers. The advantage of replacing human trading with computers is that computers can find the laws and phenomena that people are difficult or unable to capture from the vast historical data [1]. On the other hand, it can greatly reduce the influence of traders' mood swings and avoid making irrational decisions in extreme market situations [2]. With the rapid development of artificial intelligence, it is the dream of every financial trader and strategy researcher to train an automated trading model through artificial intelligence technology. Although machine learning techniques have made many achievements in areas such as medicine and biology [3], [4], object recognition [5], time series prediction [6], [7] and industrial production [8]–[10], the application of machine learning techniques to the automated trading is not an easy thing [11]. Unlike common supervised learning methods, automated trading has no labels to learn, so it needs to continuously explore through an unknown environment and continuously update and optimize the decision through the feedback of the environment, and that is also the way to reinforcement learning [12].

Mnih *et al.* [13] published a paper in 2015, which showed computer trained by Google's DeepMind team to play the Atari game, with the model trained to play seven different games, three of which even surpass the level of the human being. Two years later, they improved the model, using the model to play 49 different games, half of which exceeded the human level. By early 2017, Silver *et al.* [14] developed

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

AlphaGo to beat the world's top go player by using deep reinforcement learning technique. Reinforcement learning [15] as a branch of machine learning, it is mainly used to solve the problems of Markov decision type [16]. Reinforcement learning is usually divided into two categories [15], one based on policy learning and one based on value learning. The common methods based on value Learning include q-learning [17], td-learning [18] and sarsa [19]. These methods usually learn to obtain a value function, which can be used to get the value of actions. Value-based methods are always applied to solve the optimization problems defined in a discrete space, and they can achieve good effect [20].

The value-based reinforcement learning solves a lot of problems, but it is difficult to apply it directly to financial market transactions [21]. Value-based methods, like q-learning are learned based on discrete states, while the environment of the financial market is so complex that it cannot be approximated as a discrete space. In order to deal with the continuous data, policy-based method is used, which learns the action distribution directly from the continuous sensory data, so it is suitable for continuous financial data [22]. The policy-based method selects the optimal action by estimating the distribution of the action, and in this paper deep neural network is used to estimate the distribution of the action to select the optimal action, and the object function can be guided by the reward of environmental feedback to realize parameters updating.

In recent years, there have been many theoretical achievements and practical applications in applying intelligent techniques to financial transactions. Rather *et al.* proposed [23] a model constituted of linear model and recurrent neural network (RNN), which achieve an outstanding prediction performance in the time series prediction problems. Jin *et al.* [24] proposed a deep learning-based stock market prediction model, which combined long short-term memory (LSTM) and the analysis of investors' emotional tendency, and the experiment results show that the LSTM model can improve the prediction results. Fuzzy neural network (FNN) is also used for financial time series prediction recently, and Zaiyi *et al.* [25] proposed a novel FNN architecture for stock market prediction, which exploited the price percentage oscillator for input preprocessing, and numerical experiments conducted on real-life stock data confirmed the validity of the model. In order to solve the problem of intrinsic unpredictability of the market, Lee *et al.* [26] proposed using rough set to develop an efficient real-time rule-based trading system (RRTS), and the profitability of the model was examined through an empirical study.

Reinforcement learning has also been proved its effectiveness in quantitative trading in recent years, and many achievements have emerged. Moody *et al.* [27] first applied reinforcement learning to the financial transactions, and they proposed a trading system based on recurrent reinforcement learning (RRL), where price was directly feed into the model as learning features for training. Although the price contains all the information, there are many uncertainties in the financial market, such as the changes in economic policy, the false information of the corporation. The uncertainties in the market will affect the direction of the price, so it is important to reduce the noise from the inputs. Deng *et al.* [28] proposed a trading system based on fuzzy deep direct reinforcement (FDDR), which applied the fuzzy learning to the denoising of the price. The results on both the stock-index and commodity future contracts demonstrated the effectiveness of the trading system in the various market condition. Through the fuzziness of price, the noise can be reduced to a certain extent, but the fuzzy processing will increase the state space, that made it harder to generalize about current market trends. Gabrielsson and Johansson [29] combined features based on Japanese candlesticks [30] with recurrent reinforcement learning to produce a high-frequency algorithmic trading system, and empirical study showed a statistically significant increase in both return and Sharpe ratio compared to relevant benchmarks. Li *et al.* [2] utilized the stacked denoising autoencoders and LSTM (SDAEs-LSTM) to propose a novel trading agent, which can autonomously make trading decisions and gain profits in the dynamic financial markets. In order to extract features from the high-noise market, it utilized the stacked denoising autoencoders and LSTM as the function approximator, which can achieve stable risk-adjusted returns in both the stock and the futures markets.

Even though many approaches related to the financial trading system have been proposed in recent years, some problems are still open. On the one hand, high noise and unpredictability of market environment are the inevitable problems to be considered [31]. Therefore, how to select features of the market based on rapidly changing financial data for learning is critical when building models. In addition, it is also a question to consider how to adapt the trading strategy dynamically in the changing market [28].

In order to solve the difficulties mentioned above, this paper proposes a deep reinforcement learning model based on the decomposition of K-line [30] and clustering for online trading. The K-line uses candlesticks as a generalization of price movements over a specified period, where the process can have the initial effect of denoising. Moreover, a combination of multiple K-lines can also be used as a signal for price changes. To further extract the features of high robustness, the K-line is decomposed, and the results of each sub-part of K-line are clustered. Clustering is discussed using K-means [32], fuzzy c-means clustering method (FCM) [33] and an online clustering method based on data density [34]. The corresponding cluster centers obtained by clustering each part of the K-line are used as the input state of the model. Furthermore, the learning features are transformed through the deep neural network, and reinforcement learning is used to optimize the execution effect of the decision by constantly interacting with the environment. The main contributions are summarized as follows:

- A trading system based on deep reinforcement learning algorithm is proposed, which can achieve adaptive control in the unknown environment. Real data in stocks

and futures demonstrate the effectiveness of the proposal model in high-noise, unpredictable market environment.
- K-line theory is used to process the time series data of financial derivatives, which can be used as a generalization of prices movements over a specified period, where the process can play the role of de-noising. Furthermore, the combination of K-lines can also be used as a signal for market transactions.
- Three clustering methods are used to discuss the clustering of K-lines, and the clustering centers are used to construct the state space of deep reinforcement learning, which can extract learning features from the perspective of specified spatio-temporal relationship.

In the experimental part, real data from futures and stock markets are used. The proposal method is compared with other DRL models based on different learning features, and prediction-based models such as RNN are also used to compare the price prediction accuracy. The effects of different clustering numbers, time windows and closing time points are also discussed. Finally, the robustness of the trading system is verified by raising transaction charges.

The rest of the paper is as follows. Section II introduces the framework of the trading system and its detail implementation. Section III is the experimental part, where the performance of the trading system under real data are verified. Section IV is the summary and outlook of this paper.

II. MODEL DESCRIPTION

A. K-LINE PRESENTATION OF FINANCIAL DATA

The K-line [30], also is called candlestick, originated from the trading of rice market in feudal Japan, which was used to calculate the daily fluctuation of rice price. After 300 years of development, K-line has been widely applied to the securities markets, such as stock, futures and foreign exchange. The traders who have been engaged in relevant work for a long time have summarized some laws through some special areas or forms of the K-lines, and the patterns concluded by these laws have a great probability to predict the rise or fall of the price.

A K-line is used to represent the stock trading prices during a trading time period [35], which is shown in FIGURE 1. The first, last, highest and lowest transaction price during a particular period are called as the open, close, high and low price, respectively. A box used to make up the difference between the open price and the close price is called the body of the K-line. The thin line above the body is called the upper shadow line and the below is called the lower shadow line. When the close price is higher than the open price, the body color is red, otherwise the body is green.

When the input data is received, the data (High, Open, Low, Close) is first processed by K-line processing, and then the data after the K-line processing is calculated to get the length of the upper shadow line, the lower shadow line, the body and the color of the K-line body (0 represents red and 1 represents

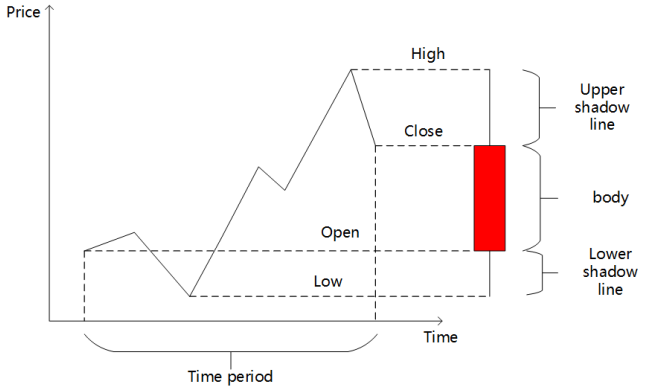

FIGURE 1. The price fluctuation represented by K-line.

green). Finally, the data at time t can be expressed as K_t:

$$K_t = (u_t, l_t, b_t, c_t) \tag{1}$$

Among them, u_t, l_t, b_t, c_t correspond to the upper shadow line length, the lower shadow line length, the body length and the body color in FIGURE 1 respectively.

B. DECOMPOSITION OF K-LINE AND CLUSTERING

After obtaining the upper shadow line, the lower shadow line and the length of the body, then clustering processing is needed. When clustering is carried out by K-means [32], the training data is divided into 5 classes (the section E of part III will discuss the number of clusters), and then the center of each class is obtained. Each input sample is represented by the cluster center which it belongs to.

FCM [33] is an algorithm that determines each data point belongs to a clustering center according to the membership degree, and this clustering algorithm is an improvement of traditional hard clustering algorithms [36]. Since FCM introduces membership function, fuzzy level and other parameters in the optimization function, the process of optimization iteration will be different from that in K-means. As a result of the final clustering center will also be different. Therefore, the FCM is also discussed. When clustering with FCM, it is also the default to divide the training data into 5 classes.

The online clustering method based on data density [34] is a clustering method that calculates data density by recursion and divides data by data density. It can automatically adjust the clustering center and update parameters online without the need for iterative training, and it can quickly adapt to new trends by learning from current data input while ignoring past data and retaining only important information. When using online clustering based on data density method, one need to calculate the sum of the distance between the current new input and the previous inputs in the sample space:

$$\pi_n(x_n) = \sum_{i=1}^{n-1} \|x_n - x_i\|^2 \tag{2}$$

where x_i represents the data of clustering, i.e., u_t, l_t and b_t, The density of the n^{th} newly input data is as follows:

$$D_n(x_n) = \frac{\sum_{i=0}^{n} \pi_n(x_i)}{2n\pi_n(x_n)} \tag{3}$$

But as more and more data enter online, it can be very difficult to calculate through the above methods, so a recursive form of calculation is used:

$$\pi_n(x_n) = n\left(\|x_n - u_n\|^2 + X_n - \|u_n\|^2\right) \tag{4}$$

$$\pi_n(x_i) = \pi_{n-1}(x_i) + \|x_i - u_n\|^2, \quad i = 1, 2, \ldots, n-1 \tag{5}$$

$$where\ u_n = \frac{n-1}{n}u_{n-1} + \frac{1}{n}x_n, \quad u_1 = x_1 \tag{6}$$

$$X_n = \frac{n-1}{n}X_{n-1} + \frac{1}{n}\|x_n\|^2, \quad X_1 = \|x_1\|^2 \tag{7}$$

Additionally, the sum of π_n can also be calculated by a recursive method:

$$\sum_{i=1}^{n} \pi_n(x_i) = \sum_{i=1}^{n-1} \pi_{n-1}(x_i) + 2\pi_n(x_n) \tag{8}$$

Through the above methods, the density of each new input data can be calculated quickly and efficiently. When the new sample data is calculated to obtain its density, the following conditions are checked to determine whether it can form a new cluster center:

$$\begin{aligned} &if\ \left(D_n(x_n) > max_{i=1}^{N_n} D_n\left(x_i^*\right) or D_n(x_n) < min_{i=1}^{N_n}\left(D_n\left(x_i^*\right)\right)\right) \\ &\quad Then\ k = argmin_{i=1}^{N_n}\left(\left\|D_n(x_n) - D_n\left(x_i^*\right)\right\|\right) \\ &\qquad d_k = x_k^* \end{aligned} \tag{9}$$

where x_i^* is the i^{th} center point, and d_k is the k^{th} center point. If a new input data can form a new center, then it belongs to the new cluster center, otherwise the distance between the newly input data and the other cluster centers is calculated, and the minimum distance clustering center is the center of it.

When using K-means or FCM clustering, the number of clustering and the maximum number of iterations or fault tolerance should be determined, and online clustering based on data density requires the first data input to initialize the structure. After obtaining the upper shadow line, the lower shadow line and the length of the body of the K-line, the data are then clustered by the above three clustering methods.

C. DEEP REINFORCEMENT LEARNING

The results of K-line clustering through a sliding window constitute the input state of DRL model, where the state of each moment is:

$$s_t = \left(d_t^u, d_t^l, d_t^b, c_t\right) \tag{10}$$

where d_t^u, d_t^l, d_t^b represent the upper, lower and entity length clustering centers of the input data at time t, respectively. A standard reinforcement learning algorithm usually involves a process of exploration and exploitation [15]. Exploration can help the model fully understand the state space of its environment, and exploitation can help model to find the optimal sequence of actions. After the model receives the input state, the action will be selected by a $\varepsilon-$greedy strategy [37]. In this paper, the initial random parameter ε is set to 0.3, i.e., that means there's a 30% chance of randomly exploring the action space at the beginning, and it will gradually decrease with the increase of training epochs, and its lower limit is 0.05.

According to the learning rules of deep reinforcement learning, the initialized model will generate action based on the current state of the environment, then the action will react on the environment, and the model will updated according to the rewards of the environment feedback, so as to optimize decisions execution. To be specific, the selected action a_t is input into the environment, then the reward r_t from environmental feedback is derived, and the experience (s_t, a_t, r_t) is stored in the memory buffer of the DRL model. Continuous updates mean more time consumption, which is not suitable for real-time trading scenarios, so experience playback process with the data stored in the memory buffer can be used to update the current parameters every specific trading times to adapt to the new environment. Through this training method, it can adapt to the real-time trading scenario, i.e., realize online trading at trading time and offline training when the market is closed. If the model creates a trading action at the time t, the environment will feedback a real profit of closing position at the time $t+1$, and then the time t moves to $t+1$. If there is no trading action at time t, it's just t to $t+1$. The thing should be noticed is that the financial market is hard to be influenced by small trading behaviors, so an assumption shall be made is that the influence of the agent on environment is insignificant. This means that the trading actions do not affect the market state, but only the time will move from t to $t+1$. FIGURE 2 shows the overall framework of the system.

When building the objective function, supervised learning is usually accomplished by optimizing the sum of squared errors between the actual and target output. The method of reinforcement learning can guide the updating of parameters through the reward of environmental feedback. In the objective function constructed (11), the reward size r_t is an important factor which will guide the parameter training:

$$maximize_\theta L(\theta) = \sum_{t=1}^{T} log\varphi(a_t \mid s_t, \theta)\, r_t \tag{11}$$

In the process of optimization, the goal is to maximize the objective function for optimized trade execution. If a big reward from the environment is got under a condition which has a very small probability $\varphi(a_t|s_t)$, then increase the degree of update to it to increase the probability of making a profit in this state. Parameters training is through Adam algorithm, through which the training parameters are not easy to fall into the local optimal, and the update speed is fast. The specific algorithm flow can refer to Algorithm 1, where θ represents all the parameters in the deep neural network.

Increasing the depth and number of nodes of the deep neural network means more time consumption, so based on the experience in previous article [28], the structure of the five-layer neural network is chosen, in which each hidden layer

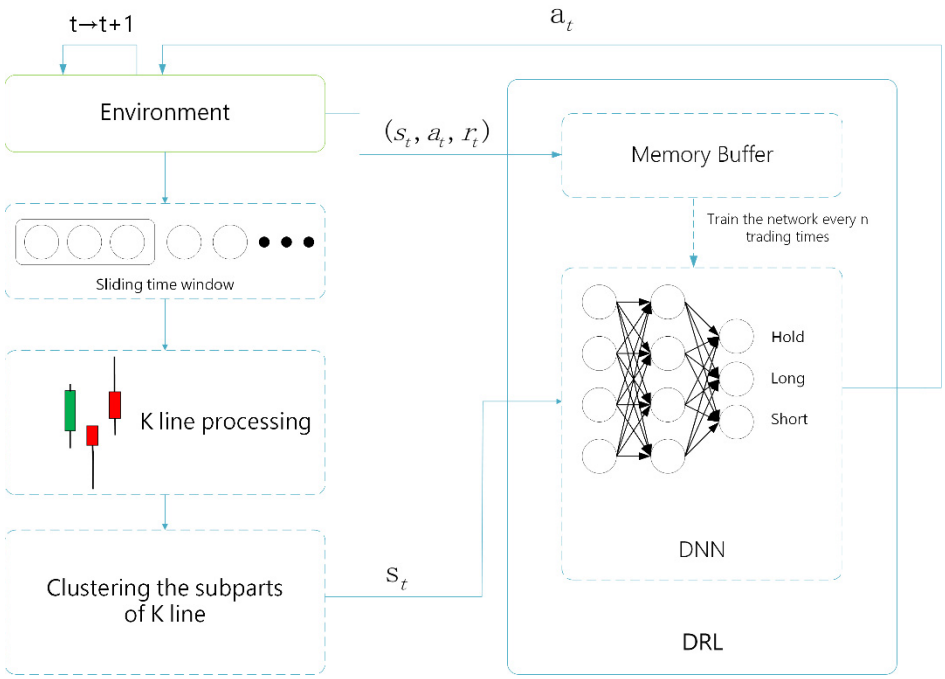

FIGURE 2. Framework of the proposal system.

has 128 nodes, the output layer has 3 nodes (it corresponds to three trading actions), and the input layer has 12 nodes. In the implementation of the DDN model, the python-based tensorflow framework developed by Google is used, and tensorflow provides a lot of modules for implementing neural network models.

III. EXPERIMENTAL VERIFICATION

A. EXPERIMENTAL SETUP

When train and test the model, the real data from financial markets are used, including commodity futures and stock index futures. For the commodity futures, the rebar (RB) contract and meal (M) contract is chosen, because they have high liquidity in the market. For the stock index futures, the IF 300 index is chosen. It is obtained from 300 large scale and good liquidity shares selected from the Shanghai and Shenzhen securities markets as samples. The IF 300 index sample covers about 60% of the market value of the Shanghai and Shenzhen market, and has a good market representation. All selected contracts allow both short and long operations. The long operation (respectively, short operation) means that one can makes profits when the market price goes higher (respectively, lower).

All the experimental data are real data, from the historical database which maintains the data crawled from each trading day and the data are also available conveniently through many historical data sources, such as yahoo finance and google finance [38]. In the experiment, the 1-minute data is used for commodity futures and stock index futures, that is, the interval between the two prices is one minute. Since the time of night market of the M contract futures is short and the IF 300 index contract has no night market, the time period chosen is longer than that of the RB contract.

From the FIGURE 3, one can see the price trend of the three contracts, the red is the training data part, and the blue is the test data part. Both the RB and the IF300 contracts have an upward trend, where the data in the training part are at a lower price, and the price of the test part of the data is significantly higher than the training part. The overall fluctuation of M contracts is relatively large, and the data of training and testing are relatively close. The profits and fees that these contracts generate in trading are measured in RMB, and each transaction is set in one hand. The Profit/point here means how much money can be gained every time the price changes one point. Take the RB contract as an example, when a long position (respectively, short position) is held, one can gain 10CNY when the price increases (respectively, decreases) one point. The handling fee shown in TABLE 1 is provided by the brokerage company. There are two ways of collecting fees, one is the percentage of the transaction amount, for example, when the RB contract is opened one hand at 3500, the calculation method of fee is $3500 \times 10 \times 0.001\%$. The other is a fixed price, for example, when dealing with one hand M contract, the fee is 1.5CNY.

TABLE 1. List of experimental contract properties.

CONTRACT	PERIODS	PROFIT/POINT	FEE
RB	2017/03-2017/08	10CNY	0.01%
M	2017/01-2017/08	10CNY	1.5CNY
IF300	2017/01-2017/08	`300CNY	0.0025%

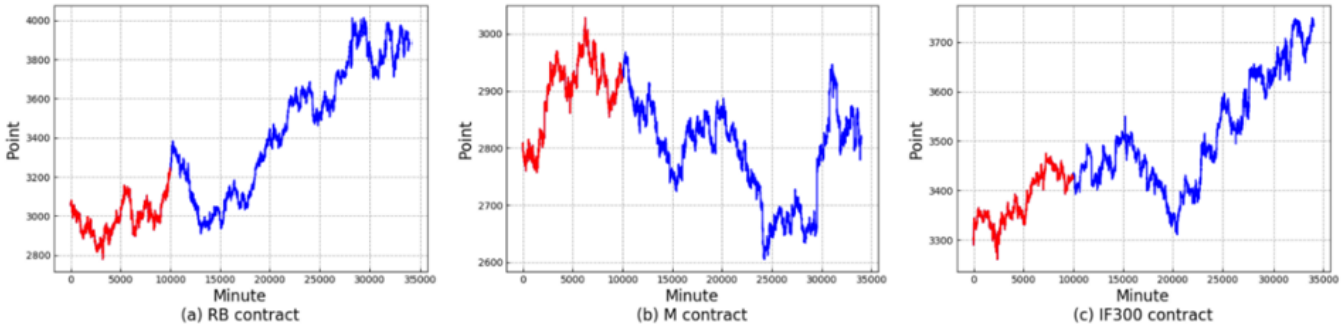

FIGURE 3. The price trend of three kinds of experimental data.

TABLE 2. Results of three clustering methods.

METHODS	CONTRACTS	CENTERS OF UPPER SHADOW LINE	CENTERS OF LOWER SHADOW LINE	CENTERS OF ENTITY
K-MEANS	RB	(0, 1.30, 3.59, 10.23, 25.12)	(0, 1.76, 2.27, 4.68, 14.81)	(0.67, 2.41, 4.90, 9.58, 22.64)
	M	(0, 1.00, 2.15, 8.85, 18.66)	(0, 1.54, 2.21, 6.10, 12.46)	(0, 1.02, 2.12, 3.35, 7.31)
	IF300	(0.07, 0.48, 0.93, 1.71, 3.96)	(0.07, 0.47, 0.97, 2.06, 5.04)	(0.22, 0.77, 1.51, 2.67, 5.13)
FCM	RB	(0.05, 1.00, 2.11, 4.24, 24.29)	(0, 1.01, 2.00, 3.13, 5.44)	(0.72, 2.24, 4.19, 6.95, 12.48)
	M	(0,1.00, 2.00, 3.25, 18.03)	(0, 1.00, 2.13, 3.32, 14.08)	(0, 1.00, 2.00, 3.13, 5.88)
	IF300	(0, 0.29, 0.68, 1.24, 2.53)	(0.01, 0.36, 0.75, 1.30, 2.29)	(0.22, 0.80, 1.54, 2.61, 4.70)
ONLINE CLUSTERING	RB	(0, 1.00, 3.00, 5.00, 7.00, 14.0, 24.0, 25.0)	(0, 1.00, 3.00, 4.00, 5.00, 6.00, 7.00, 13.00, 14.0, 24.0)	(0, 3.00, 4.00, 5.00, 11.00, 18.0, 21.0)
	M	(0, 1.00, 3.00, 4.00, 10.00, 21.00)	(0, 1.00, 2.00, 3.00, 6.00, 22.00)	(1.00, 2.00, 3.00 7.00, 8.00, 9.00, 10.00, 11.00, 13.00)
	IF300	(0, 0.20, 0.40, 0.50, 0.60, 0.80, 1.00, 1.20, 1.40, 2.40, 4.00, 4.20, 5.00, 5.60, 6.40)	(0, 0.20, 0.40, 0.60, 0.80, 1.00, 1.40, 1.60, 1.80, 2.40, 3.20, 4.80, 5.40)	(0.40, 0.80, 1.0, 1.20, 1.40, 1.60, 1.80, 2.00, 3.00, 4.20, 4.60, 7.80)

B. MODEL TRAINING

In this module, the model training is to discuss. Firstly, the training data set and test data set are divided. For commodity futures and stock index futures, the first 10000 points are used as training data, and the latter 24000 points are used for testing. In practice, in order to avoid overfitting, the first training points are divided into two sets as training set (first 8000 points) and validation set (the rest 2000 points). On the first 8000 points, the model is trained for 5 times, and the best one is selected on the performance on the rest 2000 points. When the training data is used to initialize the system, the training data must be processed with K-line, and then the results are used to train the K-means, FCM and the density based online clustering module. Then, the clustering results are continuously input into the DRL model, and the parameters are trained.

In the process of testing, since K-means and FCM algorithms cannot be updated online, they are used directly after training to get the results of clustering. Then the results are input into the trained DRL model. The online clustering based on density only needs to input the data from the environment, then data clustering and parameters updating are dynamically performed, and the clustering results are also input into the trained DRL model. TABLE 2 shows the results of clustering the training data with three clustering methods. Among them, the clustering number of K-means and FCM is fixed to five, while the number of online clustering changes with the arrival of data. It can be seen from the clustering results that the clustering results obtained by the three clustering methods are similar in value, while the number of clustering obtained by K-means and FCM is less than that by online clustering method, because online clustering will constantly update the clustering centers according to the distribution of current data.

C. COMPARING WITH OTHER DRL MODELS

This section is the contrast test part, where the comparison between the proposal DRL model and the DRL model of

Algorithm 1 The Overall Process for the Model

Input: Raw Price data (High, Open, Low, Close)
Initialization: Initialize the specific cluster method, greedy ε, the size of the sliding window w, training interval n, learning rate α and parameters β_1, β_2, ϵ for Adam.
repeat
  Update learning rate α and greedy parameter ε
  for $t = 1, 2, 3 \ldots$ **do**
    K-line processing to obtain the upper shadow line, lower shadow line, the length and the color of the body: $K_t = \left(u_t, l_t, b_t, c_t\right)$;
    Get the clustering centers of the sub-parts of K-line by clustering method to form the state of time t:
      $s_t = \left(d_t^u, d_t^l, d_t^b, c_t\right)$;
    if $t < w$ **then**
      continue
    Sliding window append s_t
    Input state of sliding window $(s_{t-w+1}, s_{t-w+2}, \ldots, s_t)$ into network;
    Choose action a_t based on the output of network and ε-greedy strategy;
    Perform action a_t and get reward r_t from the feedback of environment;
    Store (s_t, a_t, r_t) into the memory buffer.
    if $t \,\%\, n == 0$ **then**
      Update the parameters by the data in memory buffer using Adam algorithm:
      Initialize parameter τ, m, u
      while θ not converged **do**:
        $\tau \leftarrow \tau + 1$
        $g \leftarrow \nabla_\theta L\left(\theta\right)$
        $m \leftarrow \beta_1 m + \left(1 - \beta_1\right) g$
        $u \leftarrow \beta_2 u + \left(1 - \beta_2\right) g^2$
        $\hat{m} \leftarrow m / \left(1 - \beta_1^\tau\right)$
        $\hat{u} \leftarrow u / \left(1 - \beta_2^\tau\right)$
        $\theta \leftarrow \theta + \alpha \hat{m} / \left(\sqrt{\hat{u}} + \epsilon\right)$
      end while
  end
until reach training times

other features learning is performed. Here the indicators, including total profit (Profit), the winning rate (WR), the win to loss ratio (WLR), the times of long operation (LT) and the times of short operation (ST) are used for measurement. Among them, The WR is calculated by dividing the number of profitable trades with the total trading number, and the WLR is calculated by dividing the number of profitable trades with the number of lost trades. WR and WLR play an important role in evaluating a trading model or trading strategy. Sometimes even if the WR is relatively low, the WLR is relatively high, which can also be profitable. In practical applications, the WLR can also be raised by setting stop loss.

Firstly, from the profit curve of FIGURE 4, one can intuitively see that different features learning have different effects, and K-means DRL (KDRL), FCM DRL (FDRL) and online clustering DRL (ODRL) are much better than other DRL models. In addition, only K-line processing without clustering (KlDRL) can also get a moderate effect, because after K-line processing, the features also have a certain generalization ability and denoising effect. It is not good to use the price fuzzification as a feature (FzDRL), although the fuzzy processing can reduce the noise from price to some extent, it increases the state space at the same time, which makes it difficult to converge in the process of training. The effect of putting price directly into the DRL model (PDRL) is the worst, because it is difficult to predict the future trend from the price simply. In addition, the price contains a lot of noise, and it is difficult to extract the features effectively. The results of K-line processing and clustering can not only be used as a generalization of price trend, but also can remove part of the noise in the data, so the obtained features can be converge to the DRL model as soon as possible and achieve a good effect in the training process, thus getting a better performance at the time of testing.

Secondly, from the three contract trends, the M contract has a large fluctuation in the data trend, but both long and short operations are allowed in the future market, so no matter how the price trend is, there is opportunity to make a profit. In the test data of RB and IF300 contracts, both of them have an upward trend in the overall trend, so the times of long operation is more than the times of short operation in the statistics of the trade times of the K-line clustering DRL. Besides, from the statistics of the TABLE 3, one can see that when the price trend is more obvious, the WR is high, but when the volatility is large it is relatively low, so the model is more suitable for the trend trading. While LT, ST just means the number of two kinds of trading action, it is impossible to judge who wins and who loses. e.g., trading with a lot of times but a low return each time is likely to yield the same total return as trading with less times but a high return each time. So only Profit, WR and WLR are bold in the table.

Thirdly, the three types of proposal DRL models show little difference in performance. In the experiment, the clustering number of K-means and FCM is fixed as 5, and the experiment (TABLE5 in section E) indicates that too many or too few clustering numbers will affect the final effect of the model. Although online clustering can dynamically increase the cluster center in the process of clustering, it will eventually be maintained on a fixed number of clusters, so from the point of view of clustering, the difference between the three models is not significant. On the other hand, K-means and FCM divide the data from the macro level in the process of training, so the data partition is more focused on the whole data. While the online clustering method determines whether data belongs to the current center or is formed a new center by calculating the data density online, so it is more focused on local data when dividing the data. In addition, the online clustering is faster and can be adapted to real-time clustering.

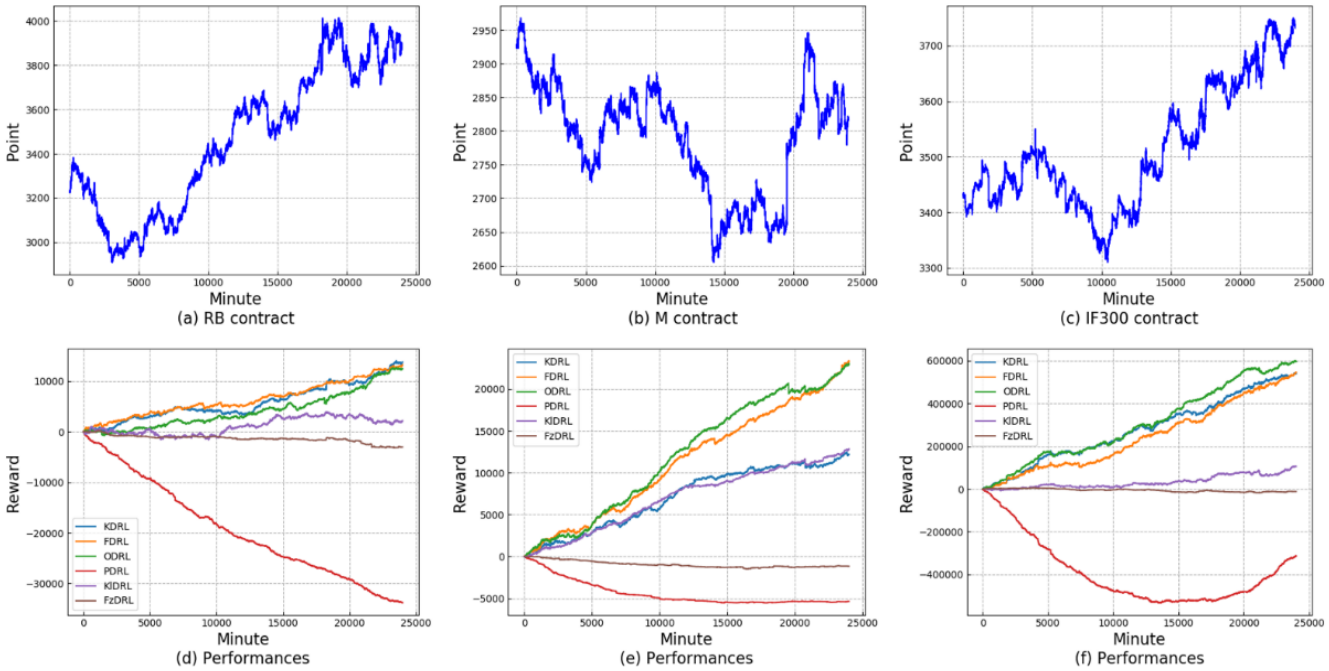

FIGURE 4. The performances of DRL trading system with different features of learning.

TABLE 3. Results of different trading systems.

	RB					M					IF300				
	PROFIT	WR	WLR	LT	ST	PROFIT	WR	WLR	LT	ST	PROFIT	WR	WLR	LT	ST
KDRL	13809.6	0.53	1.00	4479	3338	12220.5	0.45	1.49	7110	4663	539956.5	0.51	**1.13**	10882	8027
FDRL	13271.4	0.54	1.01	2817	2783	**23370.0**	0.46	1.56	7364	7816	545854.5	0.51	**1.13**	11446	8131
ODRL	**14507.9**	**0.55**	1.03	3230	2969	23051.0	0.45	**1.57**	6767	9019	**596752.5**	**0.52**	1.11	10250	7675
PDRL	-33793.4	0.38	0.81	1200	3042	-5351.0	0.31	0.91	930	118	-313563.4	0.46	0.94	2403	6021
KLDRL	2192.7	0.44	**1.28**	3881	5978	12850.5	**0.48**	1.52	2558	3235	106421.5	**0.52**	0.96	3855	3432
FZDRL	-2997.7	0.41	1.08	379	369	-1162.0	0.37	1.29	354	387	-11755.0	0.48	0.98	369	344

In the actual application, take the delay of the Internet and the high frequency of the transaction into account, and the results can be calculated faster using online clustering based on density.

Finally, Through the performance of these DRL models with different features, one can see that no matter whether the market is volatile or relatively stable, there are opportunities for profit, and the DRL model by K-line clustering is more profitable for various market conditions than other models. At the same time, although the DRL model can have a good performance in the market trading, it needs to choose the appropriate features to learn from the market.

In order to further demonstrate the effectiveness of the trading system, the first 500 points in the test data are selected from the three contracts to demonstrate the trading behavior. Then the test with ODRL is performed, since the effect of ODRL in the TABLE 3 is better. What kind of trading actions ODRL produces during the periods can be clearly seen in FIGURE 5. When price increases (respectively, decreases), it is profitable to perform long operation (respectively, short operation), so the action of long operation (respectively, short operation) generally exists in the rising (respectively, downward) trend. In addition, the trading frequency of contract is also different, where the trading frequency of M contract is higher than that of RB and IF300.

D. COMPARING WITH PREDICTION-BASED MODELS

This part compares the proposal approach with several common prediction-based models, by predicting whether the next time point is suitable for opening the position. In this part of the experiment, the S&P 500 index is used. The S&P

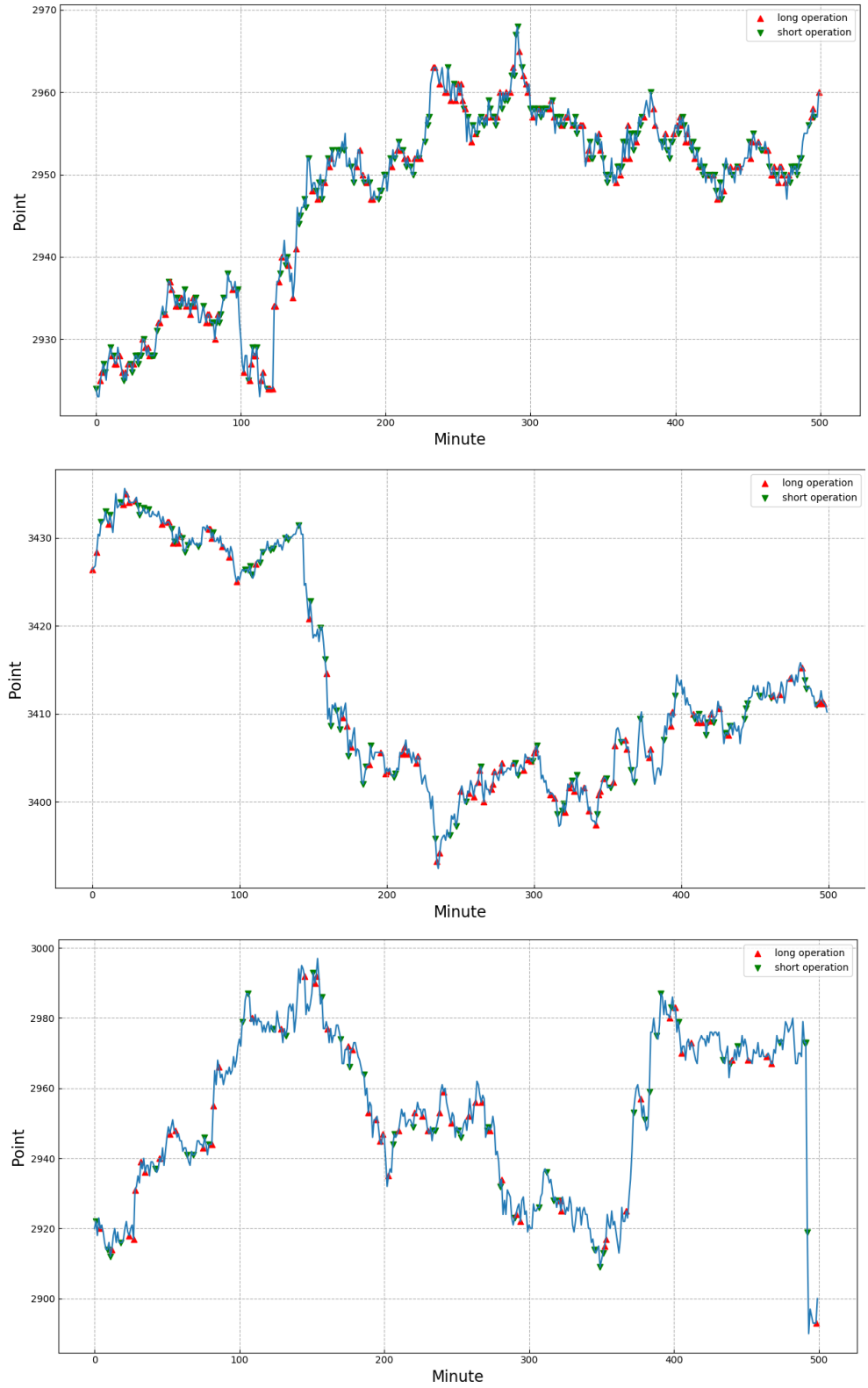

FIGURE 5. The trading points in the RB contract, M contract and IF300 contract, respectively.

500 index is a stock index that records 500 listed companies in the United States. It has the characteristics of wide sampling, high representativeness, high accuracy, good continuity and so on. Therefore, it is generally considered as the standard of an ideal stock index contract. S&P index is often used as the benchmark data set for various financial data prediction. The data used in the experiment are the daily data obtained from Yahoo Finance from January 2000 to August 2016.

TABLE 4. Comparison with neural network model.

	ACCURACY	LT
KDRL	0.538	1052
FDRL	0.547	974
ODRL	**0.551**	979
LSTM	0.528	1146
RNN	0.511	1104
FNN	0.509	1149

In the implementation of the contrast models, the tensorflow framework is used, which has many modules for model implementation. Through these modules, the prediction-based models such as RNN [23], LSTM [24], FNN [25] and other machine learning methods can be implemented conveniently. According to the structure of fuzzy inference system, FNN can be divided into Mamdani type and TSK type correspondingly, and in view of many literatures, it is shown that the identification accuracy of TSK fuzzy neural network is higher than that of Mamdani fuzzy neural network [39], therefore, TSK fuzzy neural network is used in designing contrast experiments. In the structure of TSK fuzzy neural network, the forward network is used for fuzzy inference, and the back network is used for parameter learning. The FNN contains an input layer, a fuzzy inference layer, a dense layer with 128 hidden neurons and a soft-max layer with three outputs for the prediction. The architecture of RNN and LSTM are designed in reference to the paper [28], in which the same architectures of network are used for comparison. The RNN contains an input layer, a dense layer with 128 hidden neurons, a recurrent layer, and a soft-max layer with three outputs. The structure of LSTM and RNN is basically similar, but the units used in the recursive layer are LSTM cell and RNN cell respectively. The training epochs of the neural network models are set to 50, and the time window size is set to 20. The price of the next day is predicted by the input price. Because short operation is not allowed in the stock market, the K-line clustering DRL model only keeps the actions of long and hold when training.

From the results of TABLE 4, one can see that all the prediction accuracy is slightly more than 50%, which indicates the difficulty of forecasting the market trend. Nevertheless, the accuracy of K-line clustering DRL trading system is still higher than that of the other prediction-based models. The prediction-based models are based on historical data, but the complex market rarely repeats. In addition, the price contains a lot of noise, so it is difficult to learn from the data without reducing noise. The proposed trading system denoises by K-line and clustering and it also consider the historical data and the current trend when predicting. Therefore, it can be seen from the TABLE 4 that the times of long operation of the proposed system is less than the other models, indicating that it is more considerate in the decision making.

In addition, supervised learning is usually accomplished by optimizing the sum of squared errors between the actual and target output. This scheme requires data labeled with the true target values in order to train the model. To avoid a time-consuming labeling procedure, a reinforcement learning scheme can be used instead, where the concept of a temporal difference is employed [29]. For one form of it, temporal difference learning uses the reward at time $t + 1$ as the target value for the output at time t. Furthermore, supervised learning usually treats market transactions as a binary classification problem, and it is difficult to take the magnitude of returns into account in parameters training. However, the method of reinforcement learning can guide the updating of parameters through benefits of environmental feedback. Therefore, reinforcement learning is better at optimizing the execution of trading decisions

E. COMPARISON WITH OTHER VALUES OF HYPERPARAMETERS

In this part, the hyperparameters used in the model is to discuss, including the number of clusters in K-means, FCM, time window size and closing time. The data used in the experiment is RB contract data from March 2017 to August 2017. If the number of clustering centers is set too much, the state space will be so large that the model will be hard to converge, and the setting of time window will also face the same problem. Based on the experience in the previous articles, the range of clustering number is discussed from 3 to 6, and the time window size range is discussed from 1 to 5. The later the closing time, the less current information is used, so the time of the close position is discussed from t+1 to t+3, and the experimental results are in TABLE 5, TABLE 6 and TABLE 7, respectively.

From the results of TABLE 5, one can see that if the number of clusters is too few or too much, it will reduce the effect of the model to a certain extent. This is because the setting of clustering number will affect the size of the state space. If the state space is too small, the model could not effectively extract the features from the data, and too large state space will affect the convergence of the model to some extent. Therefore, the number of clusters is fixed 5 according to the performance in TABLE 5.

When exploring the performance of different time window size, it can be found from TABLE 6 that with the increase of size, the results are getting worse. The reason is that when the window is set too large, the convergence of the model is more difficult, but when the time window is set too small, it will also affect the effect because of the too few features for learning.

When changing the closing time, it can be clearly observed from TABLE 7 that with the delay of closing time, the result is getting worse. This is because the relationship between t and $t + 1$ time is the most closely linked in price changes. The more time goes on, the more distant the relationship between price and current moment is. Therefore, the $t + 1$ time is used as the closing time when designing the model.

TABLE 5. Comparisons of different clustering number.

	Clusters number											
	c=3			c=4			c=5			c=6		
	Profit	WR	WLR	Profit	WR	WLR	Profit	WR	WLR	Profit	WR	WLR
KDRL	3671.1	0.51	1.02	12447.6	**0.53**	1.02	**13809.6**	0.53	1.00	9216.1	0.51	**1.05**
FDRL	-2415.5	0.43	1.07	12811.2	0.54	**1.02**	**13271.4**	**0.54**	1.01	12490.4	0.53	1.01

TABLE 6. Comparisons of different window size.

	Window size														
	w=1			w=2			w=3			w=4			w=5		
	Profit	WR	WLR	Profit	WR	WLR	Profit	WR	WLR	Profit	WR	WLR	Profit	WR	WLR
KDRL	1777.5	0.50	0.97	13809.6	**0.55**	0.98	**17518.2**	0.53	1.00	13978.0	0.52	1.01	10493.9	0.52	**1.02**
FDRL	8303.9	0.51	1.02	13271.4	**0.55**	1.01	**24510.8**	**0.55**	**1.05**	11647.0	0.53	1.04	9173.3	0.51	1.04
ODRL	8998.5	0.52	1.00	12507.9	0.53	1.02	**20894.8**	0.54	**1.03**	12997.3	0.52	1.00	-4196.1	0.44	**1.03**

TABLE 7. Comparisons of different closing time.

	Closing time								
	s=t+1			s=t+2			s=t+3		
	Profit	WR	WLR	Profit	WR	WLR	Profit	WR	WLR
KDRL	**13809.6**	**0.53**	1.01	5692.5	0.50	**1.03**	-2869.7	0.48	1.00
FDRL	**13271.4**	**0.54**	1.02	7293.9	0.50	**1.06**	3502.2	0.49	1.06
ODRL	**12507.9**	**0.53**	**1.03**	6066.0	0.50	1.03	1946.4	0.48	1.05

TABLE 8. The performances after raisng fees.

	f=1.5		f=2		f=2.25		f=2.5	
	Profit	TT	Profit	TT	Profit	TT	Profit	TT
KDRL	**12220.5**	11773	3400.0	7745	218.2	4223	-3237.5	3951
FDRL	**23370.0**	15180	14786.0	13162	14024.7	12869	12100	11864
ODRL	**23051.0**	15786	17218.0	13766	13790	13480	11337.5	12705

F. ROBUSTNESS VERIFICATIONS

In the real trading environment, due to Internet delays, transaction slip points and other risk factors, there will be differences in the performance of historical testing and actual trading. Therefore, in this part the robustness of the trading system is verified by raising the handling fee. The data used in the experiment are the M contract data from January to August 2017. The handling fee for M contract is 1.5CNY, and in the experiment, the handling fee is raised to 2 CNY (raised by 33%), 2.25 CNY (raised by 50%) and 2.5 CNY (raised by 66%).

From the results in TABLE 8, one can see that when the fee increase, there is less trading times (TT) for the three models. But some among them can still gain good profits, which indicates that when raising the fees, the model is more likely to execute the decision that is expected to be more profitable. These trading behaviors help the model to maintain a relative low trading frequency to avoid fees.

IV. CONCLUSION

In the face of high noise and unpredictability of financial markets, this paper proposes a deep reinforcement learning model with K-line clustering based on the actual situation of the market. With the processing of K-line, one can get the rules from history and summarize current trend. The K-line processing can denoise data preliminarily, then clustering can increase the representation of data and further reduce the noise. The results obtained by this method can be used as the

input state for the deep reinforcement learning model, where the real-time data can be better used to update the decision and the adaptive optimization of parameters.

Experimental analysis is performed on real data in the commodity futures, stock index futures and stock market. The results indicates that the proposed K-line clustering DRL model can achieve higher total profit with higher winning rate and win to loss rate than the other DRL models that are based on other features, which indicates that the proposed method is more profitable for various market conditions. Furthermore, by comparing the parameters of the model, one can get the best effect when the clustering number of FCM and K-means is set to 5, the closing time is set to T+1 and the sliding window is set to 3. Additionally, comparative experimentation using prediction-based models like RNN, LSTM and FNN, demonstrate that the proposed method can achieve higher accuracy. In the follow-up study, the framework will be extended to extract features from different spatial-temporal perspectives and combine with multiple technologies. Further study on strategy with multi commodity multi period trading will also be carried out.

REFERENCES

[1] M. R. Alimoradi and A. Husseinzadeh Kashan, "A league championship algorithm equipped with network structure and backward Q-learning for extracting stock trading rules," *Appl. Soft Comput.*, vol. 68, pp. 478–493, Jul. 2018.

[2] Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," *IEEE Access*, vol. 7, pp. 108014–108022, 2019.

[3] G. Briganti and M. O. Le, "Artificial intelligence in medicine: Today and tomorrow," *Frontiers Med.*, vol. 7, p. 27, Feb. 2020.

[4] E. Casiraghi, V. Huber, M. Frasca, M. Cossa, M. Tozzi, L. Rivoltini, B. E. Leone, A. Villa, and B. Vergani, "A novel computational method for automatic segmentation, quantification and comparative analysis of immunohistochemically labeled tissue sections," *BMC Bioinf.*, vol. 19, no. S10, pp. 75–91, Oct. 2018.

[5] W. Cao, J. Yuan, Z. He, Z. Zhang, and Z. He, "Fast deep neural networks with knowledge guided training and predicted regions of interests for real-time video object detection," *IEEE Access*, vol. 6, pp. 8990–8999, 2018.

[6] C. Luo, C. Tan, and Y. Zheng, "Long-term prediction of time series based on stepwise linear division algorithm and time-variant zonary fuzzy information granules," *Int. J. Approx. Reasoning*, vol. 108, pp. 38–61, May 2019.

[7] C. Ji, C. Zhao, S. Liu, C. Yang, L. Pan, L. Wu, and X. Meng, "A fast shapelet selection algorithm for time series classification," *Comput. Netw.*, vol. 148, pp. 231–240, Jan. 2019.

[8] H. Liu, B. Xu, D. Lu, and G. Zhang, "A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm," *Appl. Soft Comput.*, vol. 68, pp. 360–376, Jul. 2018.

[9] C. Ji, X. Zou, S. Liu, and L. Pan, "ADARC: An anomaly detection algorithm based on relative outlier distance and biseries correlation," *Softw., Pract. Exper.*, pp. 1–17, 2019, doi: 10.1002/spe.2756.

[10] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167653–167671, 2019.

[11] C. Luo, X. Song, and Y. Zheng, "A novel forecasting model for the long-term fluctuation of time series based on polar fuzzy information granules," *Inf. Sci.*, vol. 512, pp. 760–779, Feb. 2020.

[12] S. Pathak, L. Pulina, and A. Tacchella, "Verification and repair of control policies for safe reinforcement learning," *Appl. Intell.*, vol. 48, no. 1, pp. 1–23, 2017.

[13] V. Mnih, K. Kavukcuoglu, and D. Silver, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[16] H. Liu, S. Liu, and K. Zheng, "A reinforcement learning-based resource allocation scheme for cloud robotics," *IEEE Access*, vol. 6, pp. 17215–17222, 2018.

[17] C. J. C. H. Watkins and P. Dayan, "Technical note: Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[18] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, Aug. 1988.

[19] Z. Tan, C. Quek, and P. Y. K. Cheng, "Stock trading with cycles: A financial application of ANFIS and reinforcement learning," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 4741–4755, May 2011.

[20] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.

[21] J. Moody and M. Saffell, "Learning to trade via direct reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 875–889, Jul. 2001.

[22] Y. Deng, Y. Kong, F. Bao, and Q. Dai, "Sparse coding-inspired optimal trading system for HFT industry," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 467–475, Apr. 2015.

[23] A. M. Rather, A. Agarwal, and V. N. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3234–3241, 2015.

[24] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Comput. Appl.*, vol. 2019, pp. 1–17, Sep. 2019.

[25] G. Zaiyi, C. Quek, and D. L. Maskell, "FCMAC-AARS: A novel FNN architecture for stock market prediction and trading," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 2375–2381.

[26] S. J. Lee, J. J. Ahn, and K. J. Oh, "Using rough set to support investment strategies of real-time trading in futures market," *Appl. Intell.*, vol. 32, no. 3, pp. 364–377, 2010.

[27] J. Moody, L. Wu, Y. Liao, and M. Saffell, "Performance functions and reinforcement learning for trading systems and portfolios," *J. Forecasting*, vol. 17, nos. 5–6, pp. 441–470, Sep. 1998.

[28] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 653–664, Mar. 2017.

[29] P. Gabrielsson and U. Johansson, "High-frequency equity index futures trading using recurrent reinforcement learning with candlesticks," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2015, pp. 734–741.

[30] V. Popov, "Correlation estimation using components of japanese candlesticks," *Quant. Finance*, vol. 16, no. 10, pp. 1615–1630, Oct. 2016.

[31] Z. Nannan and L. Chao, "Adaptive online time series prediction based on a novel dynamic fuzzy cognitive map," *J. Intell. Fuzzy Syst.*, vol. 36, no. 6, pp. 5291–5303, Jun. 2019.

[32] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A *K*-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, Jan. 1979.

[33] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy *c*-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2, pp. 191–203, 1984.

[34] X. Gu, P. P. Angelov, A. M. Ali, W. A. Gruver, and G. Gaydadjiev, "Online evolving fuzzy rule-based prediction model for high frequency trading financial data stream," in *Proc. IEEE Conf. Evolving Adapt. Intell. Syst. (EAIS)*, May 2016, pp. 169–175.

[35] C. Luo, C. Tan, X. Wang, and Y. Zheng, "An evolving recurrent interval type-2 intuitionistic fuzzy neural network for online learning and time series prediction," *Appl. Soft Comput.*, vol. 78, pp. 150–163, May 2019.

[36] F. D. A. T. de Carvalho, Y. Lechevallier, and F. M. de Melo, "Partitioning hard clustering algorithms based on multiple dissimilarity matrices," *Pattern Recognit.*, vol. 45, no. 1, pp. 447–464, Jan. 2012.

[37] M. Tokic, "Adaptive ε-greedy exploration in reinforcement learning based on value differences," in *Proc. Annu. Conf. Artif. Intell.* Berlin, Germany: Springer, 2010, pp. 203–210.

[38] J. E. Boritz and W. G. No, "The quality of interactive data: XBRL versus compustat, Yahoo Finance, and Google Finance," Yahoo Finance, New York, NY, USA, Tech. Rep. 2013, Apr. 2013.
[39] S.-C. Chien, T.-Y. Wang, and S.-L. Lin, "Application of neuro-fuzzy networks to forecast innovation performance—The example of taiwanese manufacturing industry," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1086–1095, Mar. 2010.

DING FENGQIAN received the B.E. degree from Shandong Normal University, Shandong, China, in 2015, where he is currently pursuing the master's degree in computer science. His research interests include application of time series analysis and machine learning.

LUO CHAO received the Ph.D. degree in computer science from the Dalian University of Technology, China, in 2013. He is currently a Vice Professor with the Faculty of Electronic Information and Electrical Engineering, Shandong Normal University, China. He has published more than 30 scientific articles in refereed journals and proceedings. His research interests include complex systems, machine learning, and time series analysis.