Research Paper
Impact Factor: 7.352
Peer Reviewed Monthly Journal
www.ijmdrr.com

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

MACHINE LEARNING-DRIVEN STOCK MARKET PREDICTION USING REINFORCEMENT LEARNING TECHNIQUES

Dr. Christo Aditya Bikram Bepari
Assistant Professor, Vignan's Foundation for Science, Technology and Research (Deemed to be University), Hyderabad.

Abstract:
The large amount of data is generated everyday and it becomes very challenging to compile, analyse and making out decisions. Big Data Analytics is a boon to deal with this. Profiteering in Stock market is not an easy task because of its dynamic in nature. When to buy or sell the stocks is to be determined cautiously and thus,the massive amount of data (Big Data) used to process and formulate valuable information for correct decision making. Various algorithmic techniques has been developed and under application to improve stock trading. In this paper, Deep Learning algorithms are applied to predict different stock data values processing large amount of unstructured or unsupervised data. The five important features that define Big Data are its volume, value, variety, velocity and veracity.

Keywords: Deep Learning, Reinforcement Learning, Stock trading, Deep Q-learning, Double DQN, Dueling Double DQN.

Introduction

Optimal decision making in trading is not an easy task. A lot of investigation has to be done with stock data sets and several quantitative techniques to be applied for in-depth study.In this process, a large number of indicators with machine learning and deep learning techniques [1] such as linear regression [4, 5, 6], Moving averages [2, 3], neural networks [7, 8], Recurrent neural network [9, 10, 11, 12] and Reinforcement learning(RL) are implemented to predict the stock prices movements and strategies [13, 14]. When these techniques are applied on several data sets times and again, the results shows that in comparison to the traditional indicators and method the advanced artificial neural network perform with better accuracy [15, 16].

There are several reasons for unpredictability of stock prices with high accuracy. First and foremost is the stock market changes in no time, i.e, very rapidly changes, secondly the appropriate useful data availability is also not sufficient, i.e., incomplete information at hand and thirdly, there is no standard tools and techniques which guarantees for correct prediction in all situation. Reinforcement learning is one of the methods to solve such complex decision problems. The best way therefore that mostly believes is trial and error, i.e, the loop to conduct experiments and store experiences. Reinforcement learning work on above principle and thus can prove to be a better choice for stock price prediction [17].Deep Learning methods are significant to extract features from high dimensional data and provide hidden patterns. However, it somehow lacks the decision-making capabilities. Thus, When the Deep Learning approach is combined with the decision making ability of Reinforcement Learning, it must yield better results. RL techniques contribute to solve the algorithmic trading problem. Recurrent Reinforcement Learning algorithm have been found to give good results without the need to build forecasting models [18]. The foreign exchange market also using Adaptive Reinforcement Learning (ARL)to trade [19].The investigation results of recent times shows researchers found interesting in using Deep Reinforcement Learning(DRL) method to solve the algorithmic trading problem [20, 21, 22, 23, 24, 20, 25].

A key focus of DRL in financial analysis is the development of fully autonomous systems that mimic a financial advisor or a financial analyst, in terms of analysis and decision making for financial tasks such as trading.DRL in finance can be used by agents to make optimal policy decisions in order to receive the maximal cumulative reward or return[28]. Here, the agent collects information about the current state and tries to approximate its value using a neural network, which should lead towards an optimal decision [29].

All the reinforcement learning (RL) algorithms can be classified in several families depending on whether the algorithm explicitly learns and/or uses the environment dynamics or whether they are off-policy or on-policy. If the algorithm uses environment dynamics during the decision making process, then it is known as a model based algorithm, otherwise when it doesn't make use of them, it is known as model free. Similarly, the off-policy algorithms learn a value function independently from the agent's decisions. This means that the behavior policy and the target policy can be different. The DQN, Double DQN and Dueling DQN, all are model free and off-policy.

In the present paper,DRL is applied on ten randomly selected datasets from Indian stock market toautomate the stock trading and to maximize the gain. Model is trained with historical stock data by using Deep Q-Network (DQN), Double Deep Q-Network (DDQN) and Dueling Double DeepQ-Network (Dueling DDQN) to predict the stock trading strategy for holding, buying and selling the stocks. The unseen data from the later period is used for validation of the model and performance is evaluated and compared.

Research Methodology: Stock prediction in this paper is based on 3 main algorithm; Deep Q-Network, Double Deep Q-Network and Duelling Double Deep Q-Network [26].

2.1 Deep Q-Network: It is a simple, classical but powerful algorithm of DRL and it's model architecture may be shown as:

$$Q_{target} = r_{t+1} + \gamma max_{a'}[Q(s'_t, a'_t; \theta)]$$

where, Q_{target} is the target Q value obtained using the Bellman Equation and θdenotes the parameters of the Q-Network,Q-value function ($Q(s_t; a_t)$) is the expected future reward.

The Deep Q-Network is a model-free reinforcement learning that can deal with sequential decision making. The goal here is to learn an optimal policy that maximizes the reward or gain. The agent depends on the current state of the environment and rewards from the environment to takes appropriate action. The previous state experiences are stored along with the previous states, actions, rewards. To avoid over fitting, the data from stored memory is sampled randomly and fed to the train network in small batch sizes. One major difference between the Deep Q-Network and the Convolution Neural Network (known as Q-Network) is a new Target-Q-Network as given above.

Optimal Q-value or the action-value pair is computed each time to select and measure the actions. The max of all the actions are employed in the Deep Q-Network which leads to overestimation of the Q-value and the errors keeps on accumulating [27]. Thus, Double DQN is used to solve this problem of overestimation of Q-value. Double DQN implement another neural network that optimizes the influence of error.

Double Deep Q-Network

Double DQN uses two neural networks to provide more stability to the target values for update. One is the main network and the target network with same structure as in DQN. Here, the action is selected based on the main Q-Network but the target Q-network work to correspond to select and measure the actions and supply that particular state-action for updation. Thus, all possible actions in the present state is taken from the main Q-Network and which is updated at each time step. Then, all the state-action values of such possible actions are taken in the function argmax (Eq 2), and that specific action is selected where the state-action value maximizes the output.

$$Q_{target} = r_{t+1} + \gamma Q(s_t, argmax_{a'} Q(s_t^{'}, a_t^{'}; \theta); \theta^{'})$$

The problems of overestimation and instability in Q-values can be overcome with two neural networks as above.

Dueling Double Deep Q-Network

Both DQN as well as in Double DQN there is one main network and one target where the network values are copied periodically from the main network's values. The speciality of Duelling Double DQN is its non-sequential network architecture. Here the convolution layers get separated into two streams where both the sub-networks are fully connected layer and output layers. The first sub-network estimate the value of the given state and the second sub-network estimates the merit value of taking a particular action comparing with the base value in the current state.

$$Q(s_t, a_t; \theta, \alpha, \beta) = V(s_t; \theta, \beta) + (A(s_t, a_t; \theta, \alpha) - \max_{a' \in |A|} A(s_t, a_t'; \theta, \alpha))$$

Where A is the advantage value,V is the state-value,θ is common parameter ,α and β are the parameter vectors of the Advantage sub-network and State-Value function respectively. The value of that state which is estimated from the state-value (V) plus the advantage of taking that action in that state gives us the Q value for a given state-action pair.

The above equation can be written as :

$$Q(s_t, a_t; \theta, \alpha, \beta) = V(s_t; \theta, \beta) + (A(s_t, a_t; \theta, \alpha))$$

If we know the value of S and A, we can get the Q-value easily, but we cannot get the values of S andA if Q-value is known. To increases the stability of the algorithm, the last part of the above equation may be written as :

$$Q(s_t, a_t; \theta, \alpha, \beta) = V(s; \theta, \beta) + (A(s_t, a_t; \theta, \alpha) - \frac{1}{|A|} \sum_{a'} A(s_t, a_t'; \theta, \alpha))$$

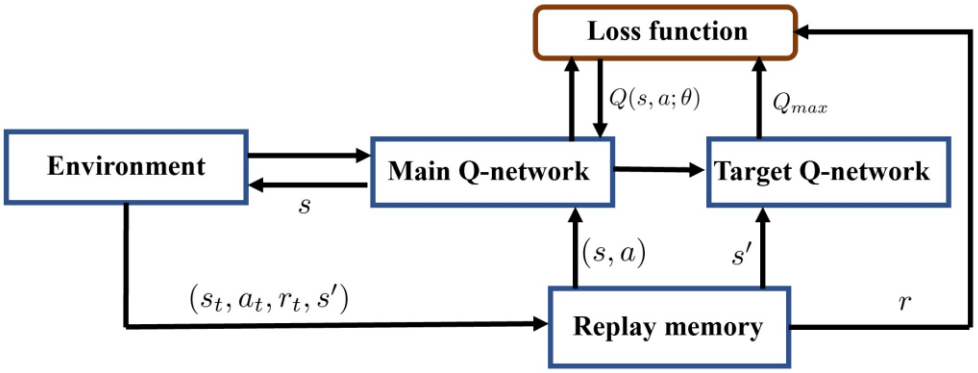


Figure 1: Deep Q-Network model architecture

Data Collection, Analysis and Results:

We collected 10 Indian stock dataset from National Stock Exchange (NSE) India, that consists of the price history and trading volumes of stocks in the index NIFTY 50 from the period 01.01.2003 to 31.12.202. Then, Deep Q-Network(DQN), Double Deep Q-Network (DDQN), and Dueling Double Deep Q-Network (Dueling DDQN) is employed to automate the stock trading and to maximize the gain. In this process, the dataset has been divided for training and testing purpose in equal proportions and fed to the models. The train and test rewards and profits are estimated and compared.

Agent Training

As we know the Q-Network has input, hidden and output layers and to obtain the optimal weights, the hyper parameters are tuned as per equation (1). In time-series problems it is very crucial for the long-term reward. The Q-Network is trained by minimizing the loss function as given below:

$$L(\theta) = E[(Q_{target} - Q(s_t, a_t; \theta))^2]$$

The learning rate is 0:00025 and the optimizer is Adam optimizer. The training is done for 50 episodes with batch size of 64 and the agent performs three actions: hold, buy and sell.

(Figure 2 Plot showing train and test dataset of ULTRACEMCO stock price)

IJMDRR
E- ISSN –2395-1885
ISSN -2395-1877

Table 1: Model hyperparameters

Hyperparameters	Values
Window size	90
Batch size	64
Episodes	50
Gamma	0.95
Epsilon	1
Learning rate	0.00025
Epsilon minimum	0.1
Epsilon decay	0.995
Optimizer	Adam
Loss function	Mean square error

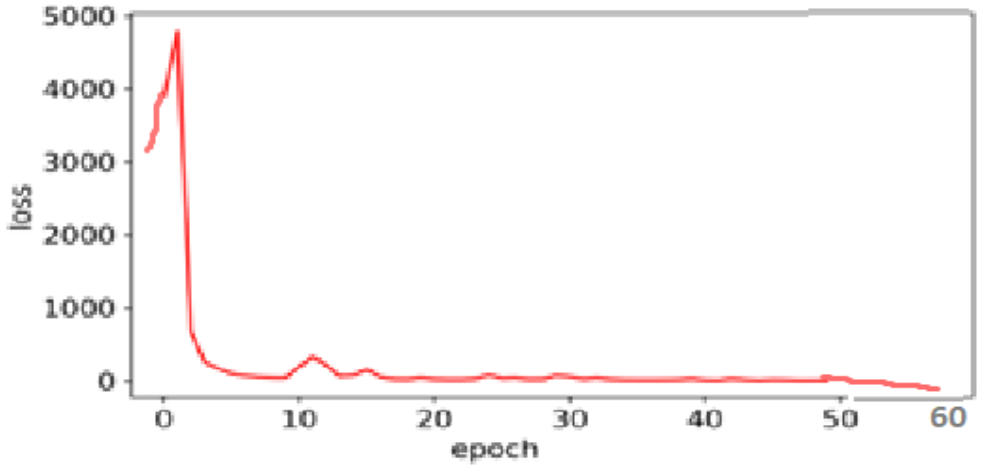
Agent Testing

The unseen test dataset of later periods are taken for the testing of the agent andthe performance of the agent is measured in terms of total profit. The profit is calculated by sale price - purchase price.

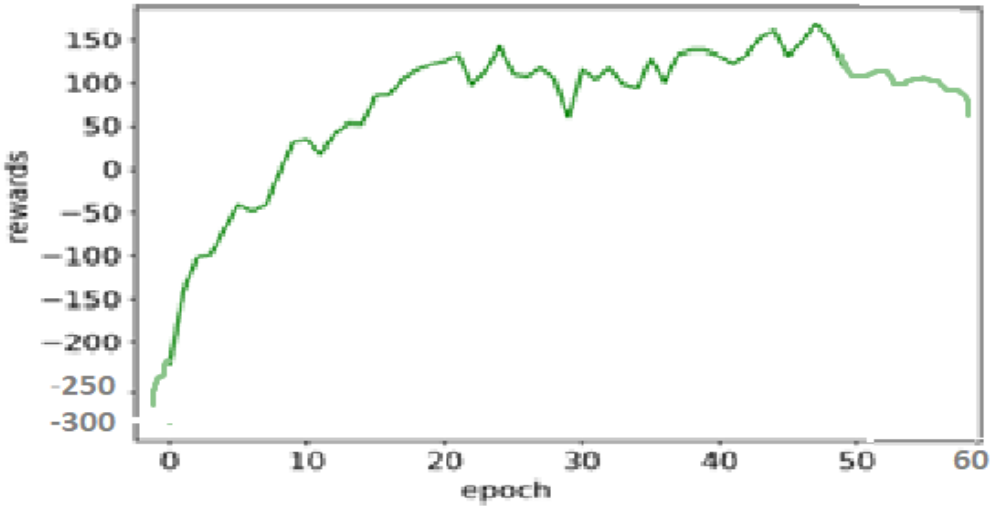
Empirical Results

In this research, ten Indian stock datasets and three deep Q-networks are employed to perform the experiments. Training to each dataset is done on train data and then tested on the unseen test data. Three deep reinforcement learning models (DQN, Double DQN and Dueling DDQN) are employed to find total rewards and profit from training data and test data usingten Indian stocks as shown in Table 2,3,4 respectively.

Figure 2 shows the train and test data used for each dataset. One stock dataset (ULTRACEMCO dataset) is selected randomly and the train and test data are plotted along with the training loss and training rewards with respect to number of epochs for DQN (Figure 2-a,b). MSE (Mean square error) is used to calculate the loss that estimates the difference between the actual and predicted values. Figure 2c shows the time-market value of the DQNmodel corresponding to the ULTRACEMCO dataset. Red, green and blue points corresponds to hold, buy and sell the stock respectively. Similarly, Figure 3-a,b,c shows the training loss, training rewards and time-market value for the ULTRACEMCO dataset using Double DQN. Figure 4-a,b,c shows the training loss, training rewards and time-market value for the ULTRACEMCO dataset using Dueling Double DQN.

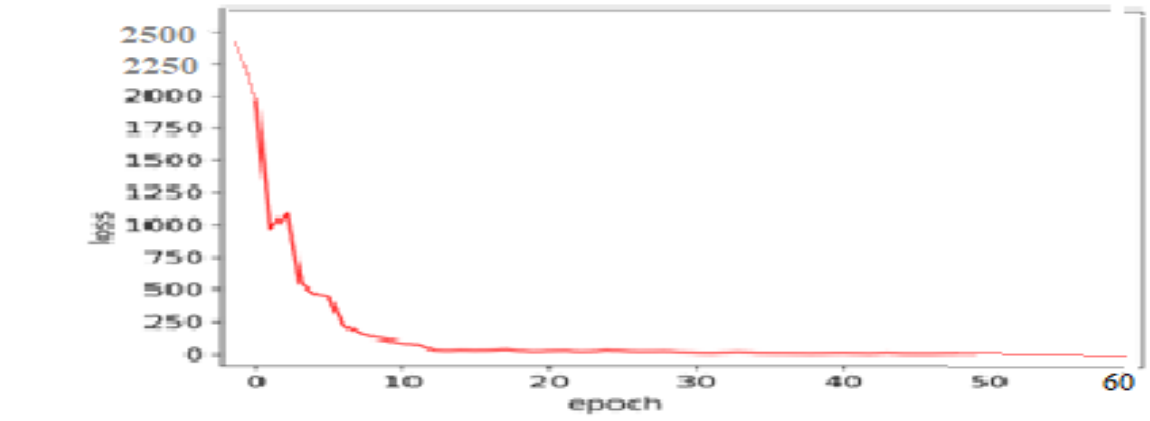


(Figure 2-a Train Loss w.r.t. Number of epochs for DQN**)**

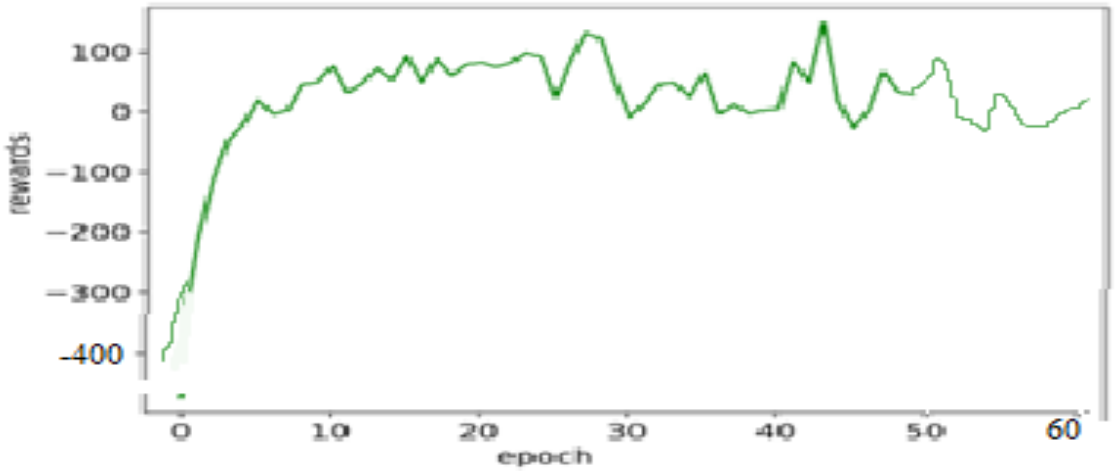


(Figure 2-**b**Train Rewardw.r.t. Number of epochs for DQN**)**

(Figure 2-**C**time-Market Value for DQN**)**

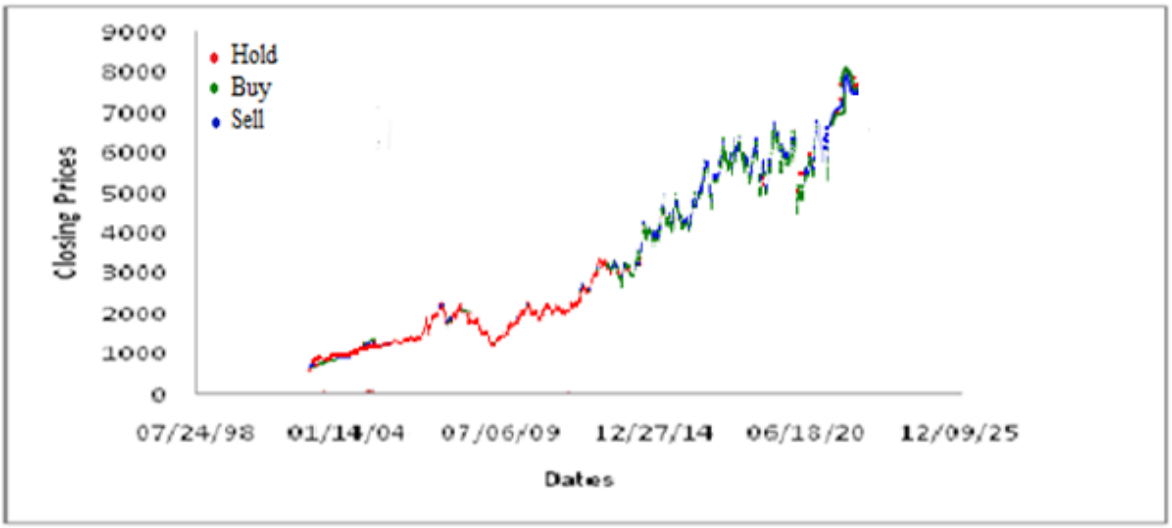


(a)

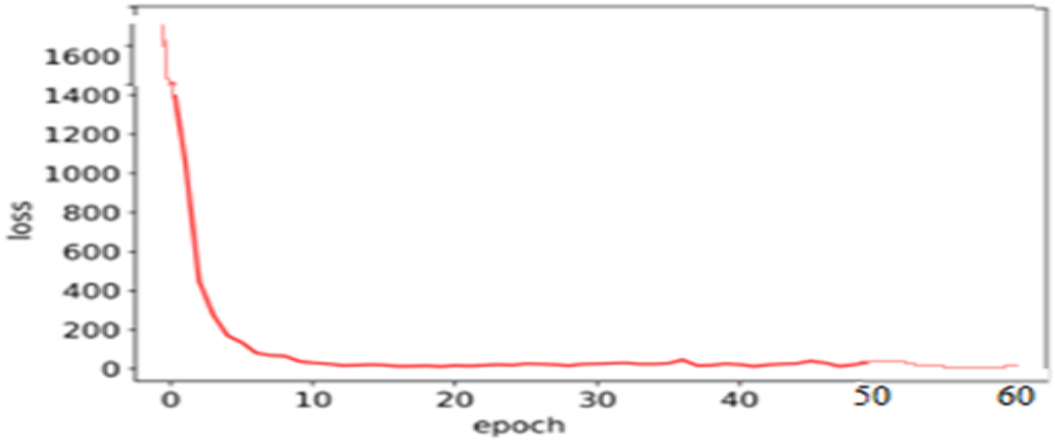


(b)

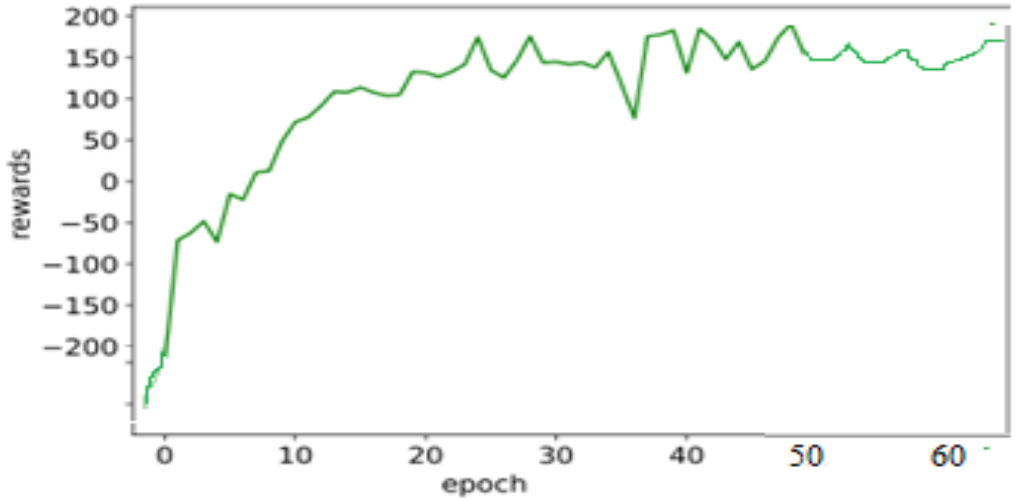
(Figure 3-a,bTrain Loss and Train Rewardw.r.t. Number of epochs forDDQN)



(Figure 3-ctime-market value forDDQN)

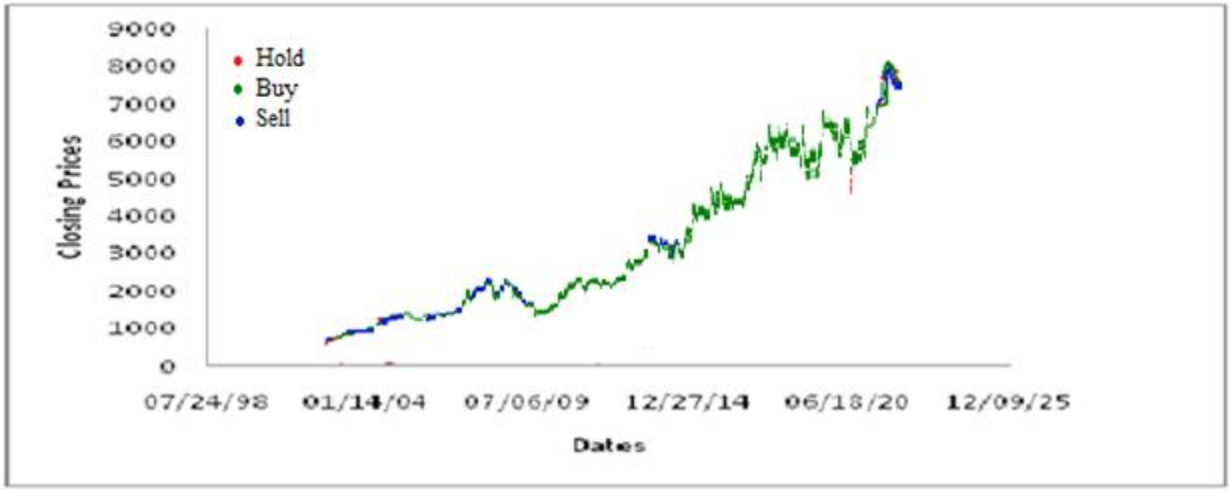


(a)



(b)

(Figure 4-a,b Train Loss and Train Rewardw.r.t. Number of epochs forDueling DDQN)



(Figure 4-cTime-Market value forDueling DDQN)

Table -2(Rewards and profit obtained using DQN)

Dataset	Train Rewards	Train Profit	Test Rewards	Test Profit
ULTRACEMCO	212	8838	19	25206
TECHM	-404	78	-387	-659
ZEEL	312	6652	132	2946
RELIANCE	126	17119	-62	1262
TCS	258	12402	28	4794
UPL	193	3692	98	4842
TATAMOTORS	224	10519	29	25202
TATASTEEL	238	3502	48	61
POWERGRID	208	1162	68	826
NESTLEIND	-138	11795	-162	16405

Table -3 (Rewards and profit obtained using Double DQN)

Dataset	Train Rewards	Train Profit	Test Rewards	Test Profit
ULTRACEMCO	38	672	335	57647
TECHM	-12	71	5	225
ZEEL	-3	21	6	18
RELIANCE	-162	0	-201	55
TCS	238	14966	289	39021
UPL	13	423	13	665
TATAMOTORS	61	702	72	992
TATASTEEL	32	1162	-3	4
POWERGRID	146	-156	149	802
NESTLEIND	5	8568	6	22028

Table -4 (Rewards and profit obtained using Dueling Double DQN)

Dataset	Train Rewards	Train Profit	Test Rewards	Test Profit
ULTRACEMCO	116	7098	29	6242
TECHM	53	25989	93	14849
ZEEL	39	1688	128	2798
RELIANCE	348	29361	329	29712
TCS	56	3481	99	17159
UPL	112	7994	168	10262
TATAMOTORS	238	16568	171	8286
TATASTEEL	2	19	6	23
POWERGRID	48	544	96	101684
NESTLEIND	121	43878	96	1224

Conclusion

In view of above, the deep reinforcement learning may be used to automate trade execution and generate profit. The performance of the DRL in solving stock market strategy problems is analysed and three DRL networks: DQN, DDQL and Dueling DDQN are compared for 10 Indian stock datasets. The result shows that the three deep learning algorithms perform well in solving the decision-making problems of stock market. Further, It may be stated that the algorithms perform better than traditional methods. A thorough observation reveals that on an average the Dueling DDQN network performed better than DDQN and DQN, and Double DQN performed better than DQN.

References

1. M Hiransha, E AbGopalakrishnan, Vijay Krishna Menon, and KP Soman. Nse stock market prediction using deep-learning models. Procedia computer science, 132:1351–1362, 2018.
2. SGM Fifield, DM Power, and DGS Knipe. The performance of moving average rules in emerging stock markets. Applied Financial Economics, 18(19):1515–1532, 2008.
3. Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. Stock price prediction using the arima model. In2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, pages 106–112. IEEE,2014.
4. Dinesh Bhuriya, GirishKaushal, Ashish Sharma, and Upendra Singh. Stock market predication using a linearregression. In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA),volume 2, pages 510–513. IEEE, 2017.
5. YahyaEruCakra and Bayu Distiawan Trisedya. Stock price prediction using linear regression based on sentiment analysis. In 2015 international conference on advanced computer science and information systems (ICACSIS),pages 147–154. IEEE, 2015.
6. J Gregory Trafton, Erik M Altmann, and Raj M Ratwani. A memory for goals model of sequence errors. Cognitive Systems Research, 12(2):134–143, 2011.
7. GoutamDutta, PankajJha, Arnab Kumar Laha, and Neeraj Mohan. Artificial neural network models for forecastingstock price index in the bombay stock exchange. Journal of Emerging Market Finance, 5(3):283–295, 2006.
8. Reza GharoieAhangar, Mahmood Yahyazadehfar, and Hassan Pournaghshband. The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in tehran stock exchange. arXiv preprint arXiv:1003.1457, 2010.
9. Zahra Berradi and Mohamed Lazaar. Integration of principal component analysis and recurrent neural network toforecast the stock price of casablanca stock exchange. Procedia computer science, 148:55–61, 2019.
10. Taewook Kim and Ha Young Kim. Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data. PloS one, 14(2):e0212320, 2019.
11. David MQ Nelson, Adriano CM Pereira, and Renato A de Oliveira. Stock market's price movement prediction withlstm neural networks. In 2017 International joint conference on neural networks (IJCNN), pages 1419–1426.
12. IEEE, 2017.
13. AdilMoghar and Mhamed Hamiche. Stock market prediction using lstm recurrent neural network. Procedia Computer Science, 170:1168–1173, 2020.
14. Parag C Pendharkar and Patrick Cusatis. Trading financial indices with reinforcement learning agents. Expert Systems with Applications, 103:1–13, 2018.
15. Terry LingzeMeng and Matloob Khushi. Reinforcement learning in financial markets. Data, 4(3):110, 2019.

16. RenJieKuo. A decision support system for the stock market through integration of fuzzy neural networks and fuzzydelphi. Applied Artificial Intelligence, 12(6):501–520, 1998.
17. Norio Baba and MotokazuKozaki.An intelligent forecasting system of stock price using neural networks. In[Proceedings 1992] IJCNN International Joint Conference on Neural Networks, volume 1, pages 371–377. IEEE,1992.
18. Jae Won Lee. Stock price prediction using reinforcement learning. In ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570), volume 1, pages 690–695. IEEE, 2001.
19. John Moody and Matthew Saffell. Learning to trade via direct reinforcement. IEEE transactions on neural Networks, 12(4):875–889, 2001.
20. Michael AH Dempster and Vasco Leemans. An automated fx trading system using adaptive reinforcement learning .Expert Systems with Applications, 30(3):543–552, 2006.
21. Yuming Li, Pin Ni, and Victor Chang. Application of deep reinforcement learning in stock trading strategies and stock forecasting. Computing, pages 1–18, 2019.
22. ZhuoranXiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. Practical deep reinforcementlearning approach for stock trading. arXiv preprint arXiv:1811.07522, 2018.
23. Yue Deng, FengBao, Youyong Kong, ZhiquanRen, and Qionghai Dai. Deep direct reinforcement learningfor financial signal representation and trading. IEEE transactions on neural networks and learning systems,28(3):653–664, 2016.
24. JoãoCarapuço, RuiNeves, and NunoHorta. Reinforcement learning applied to forex trading. Applied Soft Computing, 73:783–794, 2018.
25. IoannisBoukas, Damien Ernst, ThibautThéate, AdrienBolland, AlexandreHuynen, Martin Buchwald, ChristelleWynants, and Bertrand Cornélusse. A deep reinforcement learning framework for continuous intraday market bidding. arXiv preprint arXiv:2004.05940, 2020.
26. Jinho Lee, Raehyun Kim, YookyungKoh, and Jaewoo Kang. Global stock market prediction based on stock chart images using deep q-network. IEEE Access, 7:167260–167277, 2019.
27. MohitSewak. Deep q network (dqn), double dqn, and dueling dqn. In Deep Reinforcement Learning, pages95–108.Springer, 2019.
28. Hado Van Hasselt, Arthur Guez, and David Silver.Deep reinforcement learning with double q-learning. InProceedings of the AAAI Conference on Artificial Intelligence, volume 30, 2016.
29. Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance. arXiv preprint arXiv:2011.09607, 2020.
30. Uta Pigorsch and Sebastian Schäfer. High-dimensional stock portfolio trading with deep reinforcement learning. arXiv preprint arXiv:2112.04755, 2021.