

Stock Market Prediction and Investment using Deep Reinforcement Learning- a Continuous Training Pipeline



Amritha Sharma R, Debjyoti Guha, Hitesh Agarwal, Kothiya Meetkumar Harshadbhai

Abstract: Fluctuating nature of the stock market makes it too hard to predict the future market trends and where to invest. Hence, there is a need for a cross application backed by an ultra-modern architecture. With the latest advancement in Deep Reinforcement Learning, successive practical problems can be modeled and solved with human level accuracy. In this paper, an agent-based Deep Deterministic Policy Gradient system is proposed to imitate professional trading strategies which is a state-of-the-art framework that can predict and make investment of customers money with high return. In addition to this, dealing with interday trading strategy, the proposed architecture is designed as a continuous training pipeline so that the model saved is up-to-date with the recent market trends by giving higher accuracy in prediction. The framework outperforms the base reinforcement learning algorithms and maximizes portfolio return. The experimental result shows how natural language processing and statistical prediction can help us to choose the trending stock based on news headlines and historical data so that model invests money only in the market which gives higher return. To evaluate the performance of the proposed method, comparison of our portfolio results was done with various other reinforcement learning algorithms by keeping the same configuration.

Keywords: Deep Reinforcement Learning, Artificial Neural Network, Artificial Intelligence, Machine Learning, Deep Learning, Stock Market

I. INTRODUCTION

Previously machine-learning techniques were popular that can perform the task by learning from given data without exploring domain knowledge. Machine learning has changed drastically over the period [2],[3],[4]. The introduction of artificial neural networks (ANNs) attracted many of the researchers because of its tremendous capabilities like image recognition, natural language processing and also time series prediction which is most important for today's applications. These ML algorithms are known as Deep Learning (DL) because of its characteristic multilayer nature. Deep learning can perform complicated tasks by exploring domain knowledge [7],[12],[13].

Revised Manuscript Received on December 15, 2020.

* Correspondence Author

Amritha Sharma R*, CSE, MVJ College of Engineering, Bengaluru, India. Email: amritha.sharma27@gmail.com

Debjyoti Guha, CSE, MVJ College of Engineering, Bengaluru, India. Email: debajyotiguha11@gmail.com

Hitesh Agarwal, CSE, MVJ College of Engineering, Bengaluru, India. Email: hiteshagarwal016@gmail.com

Kothiya Meetkumar Harshadbhai, CSE, MVJ College of Engineering, Bengaluru, India. Email: kothiyameet726@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Stochastic Gradient Descent (SGD) and BackPropagation (BP) are the main techniques that help DL to perform effective learning in complex domains.

Use of ML and DL techniques were so popular among researchers and organizations to perform complex tasks like automating stock market prediction and investment. However, because of the dynamic nature of the market these algorithms weren't able to perform predictions with higher accuracy in the real world. Therefore, researchers started applying reinforcement-learning (RL) techniques in stock market prediction [6].

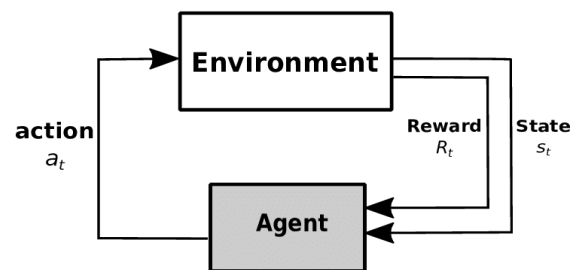


Fig. 1.A Typical RL Schematic

Traditional RL algorithms explore an unknown environment and make an optimal decision by trial and error method. By this self-learning it can achieve human-level accuracy for doing a given task. RL agents always try to maximize the future reward by applying some action on the given environment. Considering this, researchers started applying RL algorithms in stock market prediction problems that show remarkable success in that domain. The learning task for stock market prediction is challenging. Learning only from historical data doesn't help because of its volatile nature, there is a need for the model to learn continuously. In this paper we suggested a state-of-the-art architecture that learns the stock market continuously. This paper adopts the deep deterministic policy gradient (DDPG) reinforcement learning algorithm [21],[24],[23]. Multiple models created, learn from historical data every day. We also suggested how to overcome the disadvantage of use of the DDPG algorithm i.e., exploration problem.

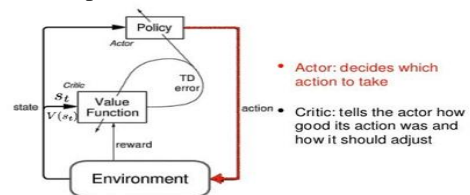


Fig. 2. An Actor-Critic Schematic

Stock Market Prediction and Investment using Deep Reinforcement Learning- a Continuous Training Pipeline

DDPG is an actor-critic framework that learns deterministic policy rather than stochastic one that helps DDPG choose the best action to perform. The critic estimates the value function and the actor updates the policy distribution in the direction suggested by the critic and both the functions are parameterized with neural networks. Selection of the best market to invest is done based on natural language processing (NLP) and statistical prediction. NLP is used to find trending markets by doing sentiment analysis on the daily news headlines [7],[23]. Sentiment analysis on news articles becomes difficult because it has a limited set of words along with expression. One of the major objectives of sentiment analysis is to allocate polarity either positive, negative or neutral to a part of the text. Hence, we have used a lexicon and rule-based sentiment analysis tool VADER lexicon for this purpose. We also combined sentiment analysis with statistical prediction that can be used to find the market which is less volatile and gives higher returns at the same time. Furthermore, we discussed how our architecture can accurately and continuously train the models and can invest money to the stock market that has higher return value with low risk. We also compare other RL algorithms over DDPG to show how it outperforms any other RL algorithm in the stock market domain. At the end we have concluded with the outcomes, advantages, disadvantages and future work that can be performed.

II. RELATED WORK

A. Stock Price Prediction Using Reinforcement Learning

In this paper[6], the author explained how the limitations of Machine Learning techniques are holding industries to trade using ML algorithms. Author suggests that ML algorithms are not adequate for learning complex problems such as stock market trading and investment. As a result, a very basic method of applying Reinforcement Learning in the domain of stock market trading was proposed. The paper proposes TD (0) algorithm which learns from experience and performs well compared to traditional ML algorithms.

B. Deep Robust Reinforcement Learning for Practical Algorithmic Trading

In this paper[1], the authors discussed previous methods to algorithmic trading. In which domain knowledge is used to extract the features that can be used to train models so that it can dynamically adjust the trading strategy, which is not so efficient. They came up with a different approach of using Deep Reinforcement Learning algorithms that can autonomously make trading decisions. They extended value based DQN and A3C along with Stacked Denoising Autoencoders (SDAEs) and the Long Short-Term Memory (LSTM) for better adaptation in the trading market. The experimental results show that they have achieved significant improvement over the baseline.

C. Deep learning for stock market prediction from financial news articles

In this paper[7], the authors suggested a new way of stock trading using a deep learning approach. They also discussed how CNN and RNN can be used for intraday trading. A framework was proposed that combines the convolutional layer with a recurrent layer (RCNN). The input to the model is a combination of historical stock data and news titles. The experimental results show that the RCNN architecture is a

better alternative to CNN and it is a success of Deep Learning methods in Natural Language Processing (NLP).

D. Practical Deep Reinforcement Learning Approach for Stock Trading

In this paper[21], the authors have explored a different DRL algorithm to cope with the complex and dynamic nature of the stock market i.e., Deep Deterministic Policy Gradient (DDPG) algorithm. They have used the historical stock market data to train their agent and also designed a min-variance portfolio allocation strategy to trade using the DDPG algorithm that suggests when to hold, buy or sell the stock. The experimental results show that DDPG archives higher return than the traditional min-variance portfolio allocation method. The only drawback of using DDPG is world exploration, it is observed that DDPG doesn't explore all the possibilities.

E. A Deep Reinforcement Learning Approach to Stock Trading

In this paper[24], the authors investigated the potential of using DRL algorithms for stock trading. Specifically, they have used the DDPG algorithm to train the agent and evaluated the agent based on three trading strategies i.e. trade only once in a week, trade only once in a month, trade only once in a quarter. The experimental results show that the DDPG agent allowed to trade once a month outperforms the buy-hold benchmark of the average across all markets. They also suggested that the use of the DDPG algorithm in stock market trading can be a novel work by further extending the algorithm to mitigate its exploration problem and creating a state-of-the-art framework using this algorithm.

F. Stock Trading Bot Using Deep Reinforcement Learning

Although Deep Reinforcement Learning algorithms perform well in stock trading but in real world application it is better to make a decision based on trending stock[23]. This can be done by analyzing the news articles and then applying the DRL algorithm to trade. In this paper the authors propose an automated trading bot using a DRL algorithm that can choose an action to buy, sell or hold the stocks to maximize the accumulated return. They have chosen a DDPG algorithm to train their agent, along with a system that can predict the trend in stock value using news articles. The sentiment analysis on the news article is performed using RCNN. The proposed architecture works only for a single stock. There is a need for a state-of-the-art architecture that can train and trade on multiple stocks by choosing the best out of n models.

III. PROPOSED METHODOLOGY

The main design of the proposed system uses statistical analysis, sentiment analysis and the base Actor-Critic DDPG algorithm which accurately predicts and invests with high returns and at low risks. First, models are created and news datasets are fed to pre-trained models via the news API for each stock. We also perform sentiment analysis to select the stock with highest sentiment, price prediction and statistical analysis to obtain which stock to buy, which stock to sell.



The following paragraph below gives a detailed description of the proposed system.

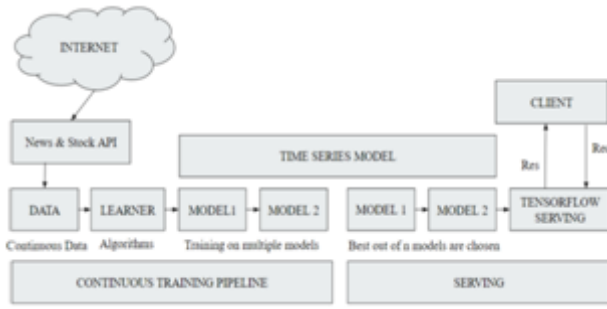


Fig. 3. High-level Architecture

A. Collecting Data

We are vastly using two kinds of datasets in our architecture. Firstly, the statistical data taken from Yahoo Finance and secondly the news data. Both the datasets are fetched real-time using APIs. Using the VADER lexicon sentiment analysis algorithm summarized in Algorithm 1, prediction on the news headlines is done. Later both datasets are fed to the Which Stock to Buy algorithm summarized in Algorithm 2 that in turn gives the stock with low volatility and higher returns.

Algorithm 1: VADER lexicon sentiment analysis algorithm

- 1: For each ticker(stock) do:
- 2: Make API call with auth header by identifying the ticker
- 3: Get the JSON response from the API
- 4: Create a DataFrame using the JSON data
- 5: For each news in the DataFrame do:
- 6: Call the SentimentIntensityAnalyzer by passing the headlines
- 7: Store the polarity score in an array
- 8: Display the polarity score using a matplotlib pie chart
- 9: Return the market(ticker) name that has more positive sentiment over others.

Algorithm 2: Which Stock To Buy Algorithm

- 1: Create an empty DataFrame
- 2: For each ticker/stock do:
- 3: Add to Data Frame the Date and Close values
- 4: Merger all the closing values of the stock on same Date
- 5: Use Pandas pct_change(), that in turn calculates $(B^0 - A^0)/A^0$ i.e. the percentage change from the immediately previous row. Where A and B are columns with index 0 and get the return.

- 6: Calculate the mean daily return i.e.returns.mean()
- 7: Calculate the Volatility i.e.returns.std()
- 8: Plot a graph of Return vs Volatility for visualization and predict which to buy and which to sell.
- 9: Return the market(ticker) name that is less volatile in nature and with high return.

Both the algorithms 1 & 2 can be performed before investing customers money into the stock. These algorithms are implemented as APIs and capable of selecting the best trending market with higher return at that point of time.

A. Learner Phase

Learner phase consists of the trained DRL agent [23] that takes two inputs close and volume. Given the input the agent chooses an action so as to maximize the cumulative reward. The DRL agent makes use of the environment object to interact. On training, it selects the action with the policy and executes daily noticing the environment. The prediction of sentiment analysis is fed as an observation of the environment to the DRL agent. The agent is made to train on multiple stocks.

B. Continuous training pipeline

As dealing with live data, we are making sure that the models are up-to-date with the current stock market trends. The main advantage of availing continuous training to the DRL agent is to learn the up-and-down of the current market price. This feature enables the architecture to be stable in any circumstances. The real-time APIs are responsible for continuously fetching and training the DRL model on a daily basis. Training the agent this way achieves higher return while learning the most recent market glitches.

C. Actor-Critic DDPG

DDPG algorithm is applied to maximize the returns on the stocks. DDPG is an algorithm that simultaneously learns Q-function by using off-policy data, the Bellman equation and a policy which is learned based on this Q-function. [21] The stochastic and interactive behavior of stock market can be formulated as Markov Decision Process(MDP) that has State $s=[p,h,b]$, Action a , Reward $r(s,a,s')$, policy $\pi(s)$, Action-value function $Q\pi(s,a)$, where p =information of the prices of the stocks, h =amount of holdings of stocks, b =the remaining stocks. Stock market actions available are Buying, Selling and Holding[21]. The DDPG algorithm of[21] is taken and in addition to a trade API is fed. The process of trade is summarized in Algorithm 3.

Algorithm 3: Trading algorithm for DDPG

- 1: Input: close, volume
- 2: Set scaled_data, real_close, close
- 3: Observe the size of queue, if less than window_size

Stock Market Prediction and Investment using Deep Reinforcement Learning- a Continuous Training Pipeline

- 4: Return status as data not enough to trade, action as fail, current balance and time stamp
- 5: Set State by calling get_state method
- 6: if action == 1 and scaled capital >= close:
Append close to inventory update scaled capital = scaled capital - close and capital = capital - real_close
Return status as buy 1 unit, cost (real_close), action as buy, current balance and time stamp
- 7: Else if action == 2 and size of inventory != 0:
Calculate invest, the default is 0 otherwise $((real_close - scaled_bought_price) / scaled_bought_price) * 100$
Return status as sell 1 unit, price (real_close), invest, gain, action as sell, current balance and time stamp
- 8: Otherwise
Return status as do nothing, action as nothing, current balance and timestamp

IV. EXPERIMENTS

Various experiments have been performed to understand how our proposed system outperforms the baseline. We have chosen other Reinforcement Learning algorithms over DDPG to check the outcomes. The experiments are performed on historical data (dated: 14/05/20) by considering the same configurations (Stock: NASDAQ: GOOG, Initial money: 10000, Window size: 20, iterations = 200, max_buy & sell = 1).

A. Actor-Critic Agent

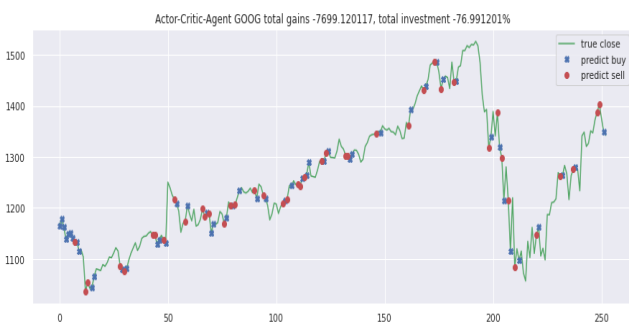


Fig. 4. Actor-Critic Agent Total Gains -7699/10000

B. Actor-Critic-Duel Agent

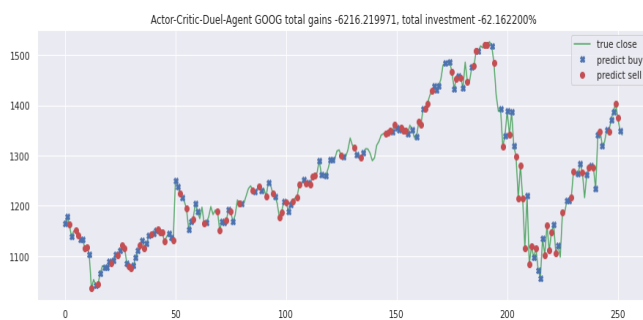


Fig. 5. Actor-Critic-Duel Agent Total Gains -6216/10000

C. DQN Agent



Fig. 6. DQN Agent Total Gains 504/10000

D. DDPG Agent



Fig. 7. DDPG Agent Total Gains 4323/10000

Table- I: Trading Performance

	DDPG	Actor-Critic	Actor-Critic-Duel	DQN
Initial Portfolio Value	10000	10000	10000	10000
Final Portfolio Value	14523	- 7699.12	- 6216.21	10504.24
Annualized Return	45.23%	- 76.99%	- 62.16%	5.04%

V. RESULTS & DISCUSSION

Based on the above experiments we can conclude that use of the DDPG algorithm in stock market trading has the potential to perform well compared to other RL algorithms. Hence, we selected the DDPG model suggested [21]. To achieve human-level control over the trading system we have also introduced sentiment analysis on news headlines [7],[23] along with statistical analysis on historical data to suggest which stock to invest, that can potentially give higher returns. These functionalities drastically change the portfolio return value when compared with the baseline. The proposed architecture selects the stock that can give higher return, then our DRL algorithm invests money in that stock automatically.



Choosing a stock that is potentially out of risk and can give higher returns crucial, because if a highly volatile market is chosen then the DRL algorithm will not be efficient enough as human to understand the up-and-down of the market. Hence, we implemented this state-of-the-art architecture that can select markets with low volatility and high return value and can train the DDPG agent on multiple markets so that it can gain knowledge about the world and invest efficiently in the selected market. The experimental results tell us that Actor-Critic agent and Actor-Critic-Duel agent are not efficient enough and give negative returns, while DQN agent performs well when compared to the other two with a total yearly return about 5.04%. While the DDPG agent performs best with a total yearly return of 45.23%. The proposed architecture makes extensive use of resources like computation power, memory, etc. The time for training multiple markets also takes longer than usual which can be neglected observing the outcomes. The architecture is efficient, stable and meant for positive growth. Architecture is flexible enough to adapt new changes and components use APIs to communicate with each other. One can easily change the DRL algorithm to test with a different one and can be deployed in any cloud environment as a SaaS model.

VI. CONCLUSION

In this paper we proposed a novel framework that mimics the professional trading strategies. We have explored the potential of training the Deep Deterministic Policy Gradient (DDPG) agent and introduced a real-time API to learn stock trading strategy. Results show that our trained agent is more robust and outperforms in accumulated return, balancing risks than the other models. This work proves that by using the most recent approach of applying deep reinforcement learning we can get human-level control over the stock market prediction and investment. This architecture is suitable for practical, real-time data and scaled to take advantage of the stocks of multiple industries. Our future work will be to focus on reducing the computation time and to make a production grade application.

REFERENCES

1. Y.Li, W.Zheng and Z.Zheng, "Deep Robust Reinforcement Learning for Practical Algorithmic Trading," in IEEE Access, vol. 7, pp. 108014-108022, 2019, doi: 10.1109/ACCESS.2019.2932789.
2. K. Hiba Sadia, Aditya Sharma, Adarsh Paul, Sarmistha Padhi, Saurav Sanyal, "Stock Market Prediction Using Machine Learning Algorithms", April 2019. <https://www.ijeat.org/wp-content/uploads/papers/v8i4/D6321048419.pdf>
3. I. Kumar, K. Dogra, C. Utreja and P. Yadav, "A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, pp. 1003-1007, doi: 10.1109/ICICCT.2018.8473214.
4. I. Parmar et al., "Stock Market Prediction Using Machine Learning," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 2018, pp. 574-576, doi: 10.1109/ICSCCC.2018.8703332.
5. Emerson, Sophie and Kennedy, Ruairi and O'Shea, Luke and O'Brien, John, Trends and Applications of Machine Learning in Quantitative Finance (May 30, 2019). 8th International Conference on Economics and Finance Research (ICEFR 2019). Available at SSRN: <https://ssrn.com/abstract=3397005>
7. Jae Won Lee, "Stock price prediction using reinforcement learning," ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570), Pusan, South Korea, 2001, pp. 690-695 vol.1, doi: 10.1109/ISIE.2001.931880.
8. M. R. Vargas, B. S. L. P. de Lima and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles," 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Annecy, 2017, pp. 60-65, doi: 10.1109/CIVEMSA.2017.7995302.
9. D. Shah, H. Isah and F. Zulkernine, "Predicting the Effects of News Sentiments on the Stock Market," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4705-4708, doi: 10.1109/BigData.2018.8621884.
10. Pang, X., Zhou, Y., Wang, P. et al. An innovative neural network approach for stock market prediction. J Supercomput 76, 2098–2118 (2020). <https://doi.org/10.1007/s11227-017-2228-y>
11. Meng, T.L.; Khushi, M. Reinforcement Learning in Financial Markets. Data 2019, 4, 110.
12. Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15). AAAI Press, 2327–2333.
13. A. Sachdeva, G. Jethwani, C. Manjunath, M. Balamurugan and A. V. N. Krishna, "An Effective Time Series Analysis for Equity Market Prediction Using Deep Learning Model," 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 2019, pp. 1-5, doi: 10.1109/IconDSC.2019.8817035.
14. Arora, Naman and M. Parimala, Financial Analysis: Stock Market Prediction Using Deep Learning Algorithms (February 24, 2019). Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur - India, February 26-28, 2019. SSRN: <https://ssrn.com/abstract=3358252> or <http://dx.doi.org/10.2139/ssrn.3358252>
15. Watkins, C.J.C.H., Dayan, P. Q-learning. Mach Learn 8, 279–292 (1992). <https://doi.org/10.1007/BF00992698>
16. Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double Q-Learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16). AAAI Press, 2094–2100.
17. Liang, Z., Jiang, K., Chen, H., Zhu, J., & Li, Y. (2018). Adversarial Deep Reinforcement Learning in Portfolio Management. arXiv: Portfolio Management.
18. K. S. Zarkias, N. Passalis, A. Tsantekidis and A. Tefas, "Deep Reinforcement Learning for Financial Trading Using Price Trailing," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, 2019, pp. 3067-3071, doi: 0.1109/ICASSP.2019.8683161.
19. Xinyi Li & Yinchuan Li & Yuancheng Zhan & Xiao-Yang Liu, 2019. "Optimistic Bull or Pessimistic Bear: Adaptive Deep Reinforcement Learning for Stock Portfolio Allocation," Papers 1907.01503, arXiv.org.
20. C. T. Chen, A. Chen and S. Huang, "Cloning Strategies from Trading Records using Agent-based Reinforcement Learning Algorithm," 2018 IEEE International Conference on Agents (ICA), Singapore, 2018, pp. 34-37, doi: 10.1109/AGENTS.2018.8460078.
21. Dang QV. (2020) Reinforcement Learning in Stock Trading. In: Le Thi H., Le H., Pham Dinh T., Nguyen N. (eds) Advanced Computational Methods for Knowledge Engineering. ICCSAMA 2019. Advances in Intelligent Systems and Computing, vol 1121. Springer, Cham doi: https://doi.org/10.1007/978-3-030-38364-0_28.
22. Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang (Bruce) Yang, and Anwar Walid, "Practical Deep Reinforcement Learning Approach for Stock Trading", 2018, arXiv:1811.07522v2.
23. Wonsup Shin, Seok-Jun Bu, Sung-Bae Cho, "Automatic Financial Trading Agent for Low-risk Portfolio Management using Deep Reinforcement Learning", 2013, arXiv:1909.03278v1
24. Azhikodan A.R., Bhat A.G.K., Jadhav M.V. (2019) Stock Trading Bot Using Deep Reinforcement Learning. In: Saini H., Sayal R., Govardhan A., Buyya R. (eds) Innovations in Computer Science and Engineering. Lecture Notes in Networks and Systems, vol 32. Springer, Singapore, doi: https://doi.org/10.1007/978-981-10-8201-6_5
25. Gran, Petter Kowalik; Holm, August Jacob Kjellevoid; Søgård, Stian Gropen, "A Deep Reinforcement Learning Approach to Stock Trading", 2019, <http://hdl.handle.net/11250/2622891>

Stock Market Prediction and Investment using Deep Reinforcement Learning- a Continuous Training Pipeline

AUTHORS PROFILE



Amritha Sharma R, BE(CSE) graduate from MVJ College of Engineering, Whitefield, Bengaluru, Karnataka



Debjyoti Guha, BE(CSE) graduate from MVJ College of Engineering, Whitefield, Bengaluru, Karnataka



Hitesh Agarwal, BE(CSE) graduate from MVJ College of Engineering, Whitefield, Bengaluru, Karnataka



Kothiya Meetkumar Harshadbhai, BE(CSE) graduate from MVJ College of Engineering, Whitefield, Bengaluru, Karnataka