An Optimized Comparative Analysis of DQN and PPO for Space invaders

Author Information

Ruchith balam[1], Abhishek Busetty[2], Hari Chillakuru[3], Uday Aira[4], Nippun Kumaar A.A.*[,5]

[1-5] Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India

* Corresponding author. Email: aa_nippunkumaar@blr.amrita.edu[*,5]
Contributing authors: BL.EN.U4AIE21017@bl.students.amrita.edu[1],
BL.EN.U4AIE21015@bl.students.amrita.edu[2],
BL.EN.U4AIE21038@bl.students.amrita.edu[3],
BL.EN.U4AIE21005@bl.students.amrita.edu[4]

An Optimized Comparative Analysis of DQN and PPO for Space invaders

Abstract. For training self-controlled agents capable of learning from environmental feedback in the area of AI and machine learning the reinforcement learning has gained prominence. A good example is the historical game Space invaders where an agent has to learn and adapt to the environment with the purpose of moving and avoiding threats. However, in the existing models, the movements of the agent are constrained to one dimension only. This research seeks to address this gap by enhancing the AI agent's movement in Space invaders to incorporate the vertical axis besides the horizontal axis. First, we set hyperparameters to Deep-Q-Networks DQN to enhance the horizontal movement, and then trained the agent with the vertical one. To evaluate effectiveness of the presented method, we also employed Proximal Policy Optimization (PPO) algorithm and compared with the DQN model. In evaluating the models trained with TensorFlow and the simulation environment, both DQN and PPO improved vertical navigation in gameplay. DQN significantly enhanced navigability, but PPO performed better with a mean score 42.67% higher than DQN's, and an overall score difference of 36.

Keywords: Reinforcement Learning, Deep-Q-Networks, Proximal Policy Optimization, Space invaders , Artificial Intelligence, Machine Learning.

1 Introduction

In machine learning, reinforcement learning (RL) is effective in developing self-controlled entities, especially in games like Space invaders . Techniques like Deep-Q-Networks (DQN) and Proximal Policy Optimization (PPO) enhance an agent's control capabilities. DQN, based on value learning, and PPO, which directly optimizes policies, provide stability during learning. Unlike conventional programming with specific control statements, RL allows agents to learn from action outcomes, as noted in the paper on Federated Deep Reinforcement Learning for Mobile Robot Navigation [12].

Space invaders poses problems to real-time because of the non-cluttered environment and the existence of opponents. Earlier models are based on simple agent motion and mostly in the form of repeated lateral or horizontal motion but severely limit exploration. Tactical positioning which operates along the vertical axis is the second type of movement, and it only moves in one direction thus restricting exploration.

This work fills the gap in the literature by adding two new dimensions to an AI agent's movement in Space invaders . Recent advancements, such as Rao et al. [13] in "Deep Adaptive Algorithms for Local Urban Traffic Control", DQN is depicted to work in complex scenarios and has the ability to make decisions in the game just as Space invaders requires. First, the DQN hyperparameters were set for horizontal movement of the agent. DQN, which is a model free reinforcement learning algorithm, applies a neural network to approximate action value function, enhances the experience replay and the target network updates. After accomplishing the horizontal movement, the vertical movement efforts were incorporated and fine-tuned as well. Moreover, PPO was employed to improve the training process by minimizing the policy directly owing to its effectiveness in continuous action space. This combined approach is expected to improve the agent's performance and speed in playing the game of Space invaders.

1.1 Major Contribution

- Inclusion of the feature of vertical movement in the agent's action space in the Space invaders environment.

- Hyperparameter tuning of DQN algorithm to optimize performance in dynamic gameplay scenarios.
- Comparative study of DQN and PPO performance for Space invaders game.

The reminder of the paper is organized as follows, Section 2 of this paper contains, a review of the various research in the field of Atari games using reinforcement learning along with the gaps in them which this research tries to resolve. Section 3 describes the methodology that this research deals with. The results obtained and analysis done with this research is in Section 4 and the Conclusion for the implementation and approach, and future work is in Section 5.

2 Literature Survey

In the case of Atari 2600 game Space invaders , two machine learning techniques have received attention, which consist of imitation learning and reinforcement learning. Another work suggested that the human imitation issue should be decomposed into sub-problems and compared it to GAIL while pointing out the training procedure benefits [1]. Abu-Farha et al. described Deep-Q-Learning and Deep Deterministic Policy Gradient (DDPG) algorithms, assessed with the OpenAI gym, Atari emulator and compared the agent's scores with those of humans and random agents [2].

As it has been established in this paper attempts towards optimization of RL algorithms deals with the performance and the amount of time taken to learn. Kumar et al. proposed ALE for the representation learning difficulties of RL agents and designed the min atari framework regarding dense computational assessment [4]. Memory and computing limited networks such as Light-Q-Network and Binary-Q-Network enhance decision making in game sequences in deep RL [6]. This paper also aims at analysing how pixel inputs are processed to percepts for better interpretability and precision in Atari games[10].

There is focus towards the creation of emotional NPCs and adding interaction to NPCs in the tank-battle games in a way that has made the AI smarter than the player by using pre-supervised learning [5]. To further increase variation and imply value addition to the game, plays similar to Space invaders , Venkatesh et al. suggested the combination of HMM and machine learning [8]. Looking at RL in relation to the NavMesh approach in game development, it was observed that RL was faster and more conducive to the creation of NPCs in well managed modelling conditions [7].

Other works also emphasize on the characteristics of RL in gaming setting. In other experiments, Vinoth et al. aimed at using Theory-Based Reinforcement Learning (TBRL) along with the bayesian tool for fast learning throughout numerous issues and games, whereas EMPA effectively trained with human sampling functions. Other authors such as Prasanna Kumar et a, al also underlined how various techniques of RL improve result in different gaming scenarios particularly during the stages of action removal and discretization of games such as VAT [9].

The reviewed papers collectively advance reinforcement learning (RL) in gaming by introducing novel algorithms, enhancing performance in Atari games like Space invaders , and addressing challenges such as interpretability and computational efficiency. They push the boundaries of AI in dynamic environments, but research gaps remain in scaling RL across complex gaming environments and expanding action spaces.

3 Methodology

We implemented and tested the AI model for Space invaders by modifying the simulation and applying DQN and PPO reinforcement learning. First, the parameters such as exploration rate and learning rate were set and adjusted using the DQN model without vertical motion in the Space invaders game. After optimizing these parameters, they were incorporated to expand the action space of the agent to include the vertical dimension.

Next, DQN and PPO algorithms were trained and tested with the set hyperparameters in order to improve the agent's control and survival rate in a dynamic environment of the game.

3.1 Game Environment Set up and RL Formulation

The agent is the player-controlled character whose goal is to maximize its score by shooting down invading aliens while avoiding their attacks. The environment represents the game of Space invaders itself, including the game screen, the aliens, the player's spaceship, and any other elements present in the game

In this environment, the state space (S) consists of variables such as the current position, speed of the alien, speed of bullets, position of aliens, current frame, and speed of the agent. The action space (A) initially includes actions like moving right, moving left, firing a bullet, or doing nothing, and is later expanded to include moving up and moving down. The reward function (R) is defined by specific actions and outcomes a reward of -0.001 is given for every frame change and for shooting, a reward of 1 is given for killing an enemy, and a reward of -2 is given for losing the game.

3.2 Deep-Q-Networks

DQN is a primary reinforcement learning algorithm. In DQN, a deep neural network is used to estimate the Q-function, that defines the sum of the potential future rewards given a particular action in a certain state. This approach is best implemented where the state space is large and when the environment is dynamic and adversarial, such as in a game like Space invaders . The target network and the experience replay are two techniques that make learning in DQN much more stable and efficient. In this project, DQN was used to enhance the horizontal motion with the help of hyperparameters such as exploration rate and learning rate and both these parameters are useful for the exploration of new strategies and exploitation of learned strategies.

3.3 Proximal-Policy-Optimization

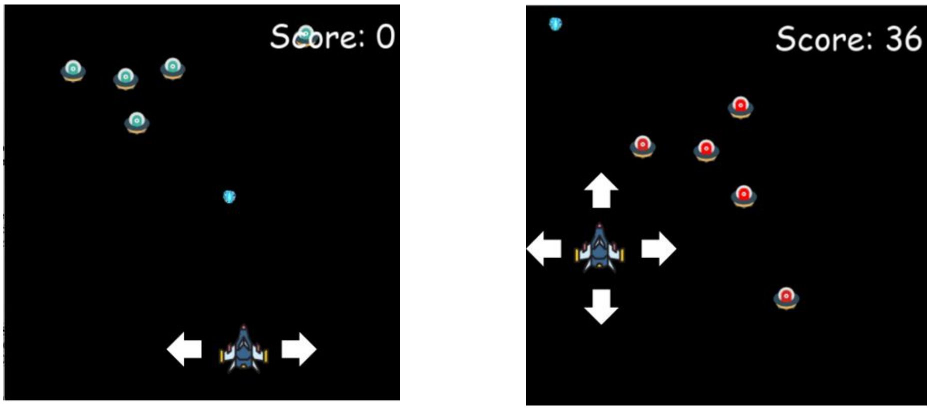
The proposed system also trained using PPO. Unlike DQN, which targets at estimating the Q-function, PPO directly acts on the policy function that defines the agent's action probabilities from the observed state. PPO is more effective when used in environments with both discrete and continuous action spaces and therefore the ability to improve the performance of an agent in games such as Space invaders is highly desirable. In this study, PPO was used together with DQN to analyse their performances after the additional action dimension of vertical movement was incorporated into the agent's action space. This made it possible to compare their learning rate and the overall game score within the same experimental setting, establishing the merits and demerits of each algorithm in improving the agent's navigation and survival rates.

3.4 Hyperparameters

The learning rate is used to define how large a step is taken to update policies in relation to new information and prior knowledge. The exploration fraction lies between exploration and exploitation, a higher value of fraction makes the state-action exploration better but if the value is high may not be efficient. Training time is in episodes the more episodes lead to better learning but can have overfitting and inefficiency with improper hyperparameters.

Three sets of hyperparameters taken for tuning the model. The learning rate and exploration fraction are varied across the sets. Set 1 has a learning rate of 0.00025 and an exploration fraction of 0.025. Set 2 has a learning rate of 0.00050 and an exploration fraction of 0.050. Set 3 has a learning rate of 0.00075 and an exploration fraction of 0.075.

3.5 Adding Vertical Movement



(a) Horizontal movement only.

(b) Horizontal and Vertical movement.

Fig. 1 Space invaders action space of the agent.

Initially, hyperparameters like exploration rate and learning rate were tuned using a DQN model where the agent couldn't move vertically in Space invaders Figure (1a). These parameters were then applied to expand the agent's action space vertically Figure (1b). Both DQN and PPO were subsequently trained with these hyperparameters.

4 Results and Analysis

Firstly, the hyperparameters were adjusted to the values. Based on the experiments which were conducted to compare different values of the learning rate and the exploration fraction, the right parameters were chosen. After that, the parameters of both DQN and PPO algorithms were trained with the optimized parameters for both agents with and without the vertical movement. The direct comparison between DQN and PPO considered the goal of assessing how these two methods affect the improvement of the agent's movement and survival skills in the Space invaders game.

4.1 Hyperparameter Tuning

For hyperparameter tuning, over 10,000 episodes for each set discussed above was used and mean reward was used as a measure to decide on the best parameters. These best performing parameters were then used to train DQN and PPO algorithms with and without vertical movement to compare and analyze the result in Space invaders game environment.

After training of 10,000 episodes the mean reward for each set has obtained. Set 1 achieved a mean reward of -0.56868, while Set 2 and Set 3 recorded mean rewards of -0.57844 and -0.58127, respectively. These values reflect the average performance of the agent across multiple episodes under the conditions specified, providing insights into the effectiveness of different strategies or configurations in the game environment.

As it is evident from the results presented above, an increase in the exploration fraction and learning rate leads to a decrease in the mean reward for all the parameter sets. In particular, concerning the first set of outcomes, the average award is relatively higher in the range of -0. 56868. However, it is evident that the mean reward in Set 3 is slightly lower and equals -0. 58127. This seems to be a general trend of negative correlation between exploration fraction, learning rate and the mean reward which implies that when these hyperparameters are set higher, the performance in terms of reward outcome is likely to be poor in the given context.

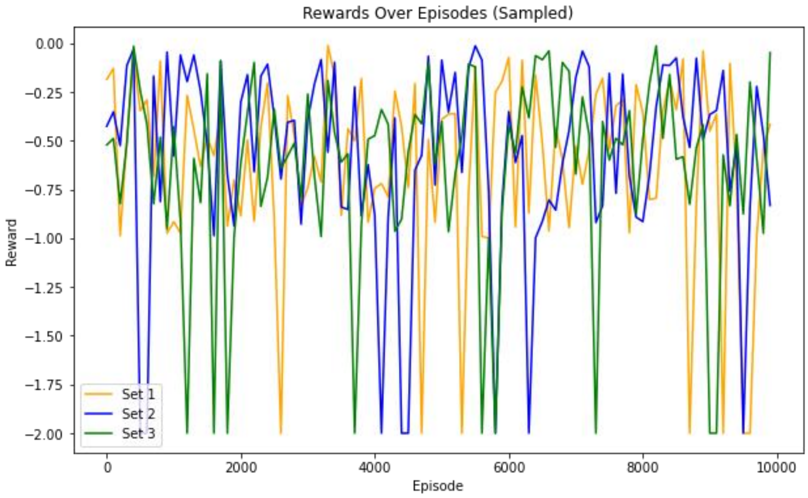


Fig. 2. The rewards are sampled over every 100th episode reward in 10,000 trained episodes.

The Figure 2 shows the hyperparameter tuning plots of rewards for every 100th episode over 10,000 episodes for 3 sets in which set 1 got higher mean reward and also a smaller number of negative peaks when compared with other 2 sets. Therefore, Set 1 parameters are chosen for further experiment study.

4.2 Training

The agent was trained for a significantly longer duration to assess its performance over extended training with both DQN and PPO algorithm.

4.2.1 Deep-Q-Network and Proximal-Policy-Algorithm

The Agent was trained over 10 million episodes with and without vertical movement to compare the performance of the agent with different action space in same environment.

Table 1. DQN and PPO agent performance metric.

Parameters	**Horizontal Movement Only**	**Vertical and Horizontal Movement**
Learning Rate	0.00025	0.00025
Exploration Fraction	0.025	0.025
Episodes	10 million	10 million
Mean Reward (DQN)	12.57	14.60
Mean Reward (PPO)	13.69	20.831

From Table 1, it can be seen that the DQN and PPO was trained over 10 million episodes with and without the vertical movement. When evaluated based on only the horizontal movement the mean reward that was recorded by DQN was 12.57 and PPO was 13.69. However, adding the vertical movement of the agent mean reward increased to 14.60 for DQN and 20.831 for PPO with the same learning and exploration factors. This implies that integrating the vertical movement can greatly improve the operation of the DQN and PPO since it supports broader exploration of the training environment as well as improved decision making.

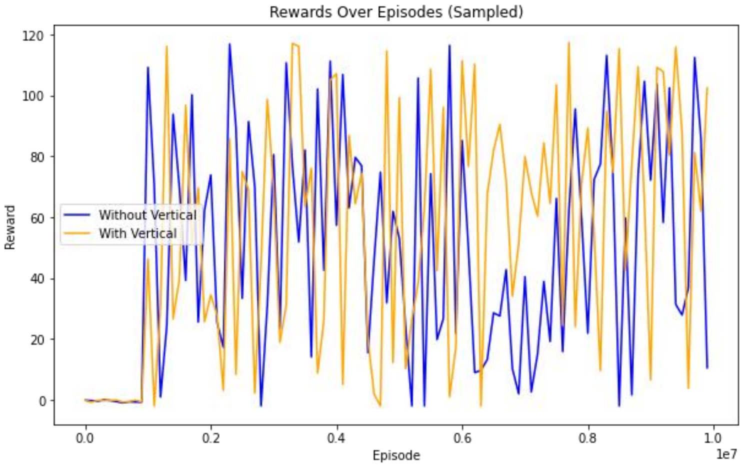


Fig. 3. DQN agent reward plot sampled over every 1 lakh episodes.

The Figure 3 shows the plot of DQN rewards of 10 million episodes sampled for every 1 lakh episode. When we observe the graph, after 6 million episodes, the rewards with vertical movement have got more when compared with rewards with no vertical movement. This shows the importance of inclusion of vertical movement.

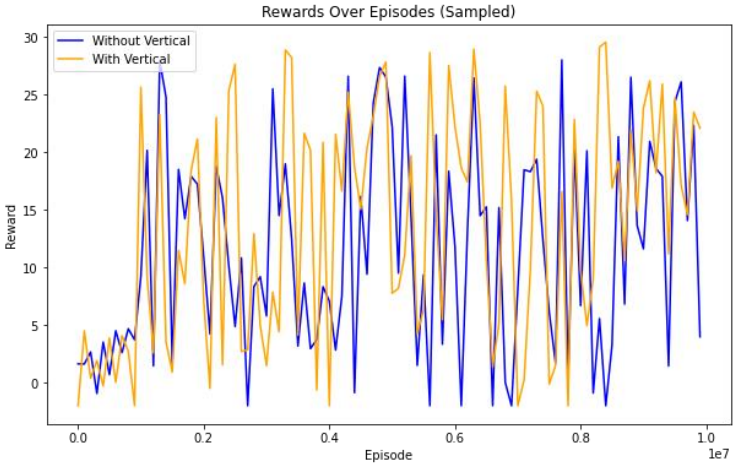


Fig. 4. PPO agent reward plot sampled over every 1 lakh episodes

The Figure 4 shows the plot of PPO rewards of 10 million episodes sampled for every 1 lakh episode. When we observe the graph, after 2 million episodes, the rewards with vertical movement have got more when compared with rewards with no vertical movement. This shows the importance of inclusion of vertical movement.

4.3 Discussion

4.3.1 Training

Based on the comparison of mean rewards between PPO and DQN with vertical movement, it's clear that vertical movement significantly enhances performance for both algorithms. In the case of DQN, incorporating vertical movement improves the mean reward from 12.57 to 14.60 in the first scenario and from 13.69 to 20.831 in the second scenario. This indicates that vertical movement helps both PPO and DQN achieve higher mean rewards, suggesting its beneficial impact on learning robust policies across different training scenarios. But when PPO and DQN are compared based on the reward we can easily say that PPO has higher mean reward so it is better than DQN.

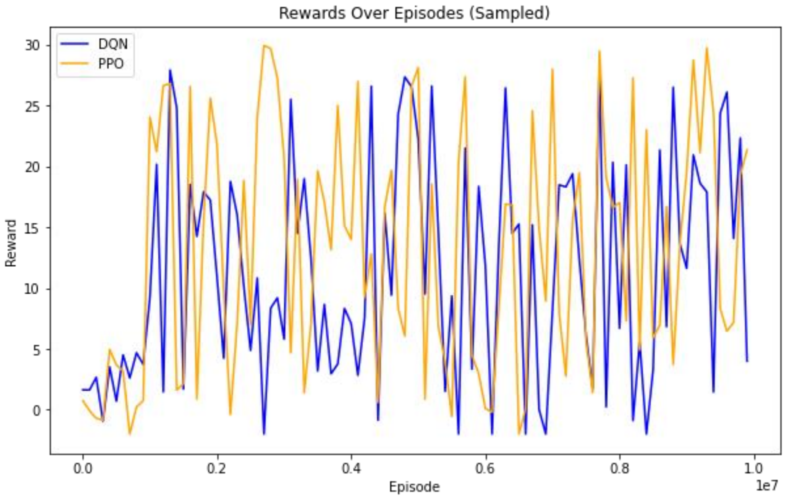


Fig. 5. The Rewards of PPO and DQN both are sampled over every 1 lakh episodes reward in 10 million trained episodes.

The Figure 5 shows the reward plots of PPO and DQN. Most of the episodes the rewards of PPO are higher than the rewards of DQN and also when mean reward is calculated PPO has higher mean reward so it is better than DQN.

4.3.2 Testing

After the training part for 200 episodes both algorithms with vertical movements were tested. Below there is a table with results for testing part.

Table 2. Mean rewards and highest Score for PPO and DQN tested over 200 episodes with vertical movement.

Algorithm	Episodes	Average Score	Highest Score
DQN	200	17.55	139
PPO	200	24.478	175

The Table 2 shows the mean reward and highest score of DQN and PPO tested for 200 episodes with vertical movement added.

Incorporating vertical movement, PPO consistently outperforms DQN in both training and testing. Over 10 million training episodes, PPO achieves a mean reward of 14.60, compared to DQN's 20.831. During testing over 200 episodes, PPO maintains a higher average score of 24.478 and reaches a top score of 175, whereas DQN averages 17.55 with a highest score of 139.

5 Conclusion and Future Scope

This research started with a hyperparameter tuning process to determine the appropriate learning rate and exploration fraction, where a negative relationship between them and mean reward was established. After determining the best hyperparameters, DQN and PPO algorithms were trained with and without vertical movements. These findings revealed that the incorporation of vertical movement enhanced the performance of both algorithms in terms of mean rewards. For instance, the mean reward of DQN rose by 16.14% , while PPO’s mean reward increased drastically by 52.16%. This superiority is also seen during the testing phase where PPO obtains an average score of 39.47% higher than DQN’s and scored a 25.89% higher than DQN’s high score. Here, these results demonstrate that PPO is a more efficient and general algorithm for training an agent in the Space invaders environment with an extended action space.

Further optimize hyperparameters such as learning rates and exploration fractions to potentially enhance performance of both algorithms. Explore transfer learning techniques to apply knowledge gained from one game level to improve performance on related levels, reducing training time.

Compliance with Ethical Standards

Conflict of Interest: Ruchith Balam declares that he has no conflict of interest, Abhishek Busetty declares that he has no conflict of interest, Hari Chillakuru declares that he has no conflict of interest, Uday Aira declares that he has no conflict of interest, Nippun Kumaar AA declares that he has no conflict of interest.

Ethical Approval: This article does not contain any studies involving humans or animals performed by any of the authors.

References

1. McKenzie, M., Loxley, P., Billingsley, W. and Wong, S., 2017. Competitive reinforcement learning in atari games. In AI 2017: Advances in Artificial Intelligence: 30th Australasian Joint Conference, Melbourne, VIC, Australia, August 19–20, 2017, Proceedings 30 (pp. 14-26). Springer International Publishing.
2. Martinez-Nieves, C.L., Defeating the Invaders with Deep Reinforcement Learning.
3. Tsividis, P.A., Loula, J., Burga, J., Foss, N., Campero, A., Pouncy, T., Gershman, S.J. and Tenenbaum, J.B., 2021. Human-level reinforcement learning through theory-based modeling, exploration, and planning. arXiv preprint arXiv:2107.12544.
4. Young, K. and Tian, T., 2019. Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments. arXiv preprint arXiv:1903.03176.
5. "Applying Reinforcement Learning for the AI in a Tank-Battle Game." (2010).
6. Li, Y., Fang, Y. and Akhtar, Z., 2020. Accelerating deep reinforcement learning model for game strategy. Neurocomputing, 408, pp.157-168.
7. Gomes, G., Vidal, C.A., Cavalcante-Neto, J.B. and Nogueira, Y.L., 2022. A modeling environment for reinforcement learning in games. Entertainment Computing, 43, p.100516.
8. Zamith, M., da Silva Junior, J.R., Clua, E.W. and Joselli, M., 2020, November. Applying hidden Markov model for dynamic game balancing. In 2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames) (pp. 38-46). IEEE.
9. Kanervisto, A., Scheller, C. and Hautamäki, V., 2020, August. Action space shaping in deep reinforcement learning. In 2020 IEEE conference on games (CoG) (pp. 479-486). IEEE.
10. Sieusahai, A. and Guzdial, M., 2021, October. Explaining deep reinforcement learning agents in the atari domain through a surrogate model. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 17, No. 1, pp. 82-90)
11. A. A. Nippun Kumaar and S. Kochuvila, "Reinforcement learning based path planning using a topological map for mobile service robot," 2023 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2023, pp. 1-6, doi: 10.1109/CONECCT57959.2023.10234766
12. Shivkumar, S., Amudha, J., and Nippun Kumaar, A.A. ‘Federated Deep Reinforcement Learning for Mobile Robot Navigation’. 1 Jan. 2024 : 1 – 16, doi: 10.3233/JIFS-219428.
13. Rao, P.S., Polisetty, V.R.M., Jayanth, K.K., Manoj, N.S., Mohith, V. and Kumar, R.P., 2024, January. Deep Adaptive Algorithms for Local Urban Traffic Control: Deep Reinforcement Learning with DQN. In 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) (pp. 592-598). IEEE.
14. Akkshay, K.U. and Sreevidya, B., 2023, December. Development and Performance Analysis of an AI based Agent to Play Computer Games using Reinforcement Learning Techniques. In 2023 IEEE 3rd Mysore Sub Section International Conference (MysuruCon) (pp. 1-8). IEEE.