RESEARCH & INNOVATION DIVISION
IJIST

Model-Based vs Model Free Deep Reinforcement Learning Models for Cancer Treatment: A Critical Analysis with a Solution DRL Model

Madeha Arif [1], Usman Qamar [1]

[1]Department of Computer Software Engineering, College of Electrical and Mechanical Engineering, NUST

***Correspondance:** madeha.arif15@ce.ceme.edu.pk, usmanq@ce.ceme.edu.pk

Citation | Arif. M, Qamar. U, "Model-Based vs Model Free Deep Reinforcement Learning Models for Cancer Treatment: A Critical Analysis with a Solution DRL Model", IJIST, Special Issue pp. 286-295, June 2024

Received | May 15, 2024, **Revised |** May 22, 2024, **Accepted |** May 31, 2024, **Published |** June 05, 2024.

In the field of artificial intelligence, deep reinforcement learning (RL) has grown to be one of the most talked-about issues. It has a wide range of applications, including end-to-end control, robotic control, recommendation systems, and systems for natural language communication. In this paper, we have critically reviewed model-based and model-free deep reinforcement models for the treatment of cancer patients and evaluated each model based on some parameters. Based on the evaluation, a critical discussion is carried out highlighting the limitations and drawbacks of all the existing models. The analysis also gives suggestions and marks the key indicators of future needs in this domain. In the end, a solution model is proposed that tries to cover all the shortcomings and addresses the issues encountered in the existing models. The findings indicate that we can get a 94% accuracy rate with three agents and that increasing the number of agents has no further effect on accuracy.

Keywords: Deep Reinforcement Learning (DRL); Model-based learning; Model-free learning; Deep Learning; Neural Network (NN)

Introduction:

Deep learning and large data have enabled dramatic advancements in the study of artificial intelligence. In the area of post-deep learning, interest in investigating new technologies is expanding. Particularly appealing is deep reinforcement learning (RL), which incorporates neural network modeling into conventional RL algorithms. Deep RL specifically determines which action to take to maximize the benefit in the face of a particular situation by solving decision optimization issues. As a result, deep RL analysis and application are receiving a lot of attention from both the academic community and industry. Between deep RL and conventional machine learning, there are significant variations in processing and analysis.

The current, widely accepted machine learning paradigm mostly gathers or creates dataset tags in advance and executes machine learning using static data that is already available. Contrarily, RL is a classic illustration of the closed-loop learning paradigm, which incorporates feedback signals into the learning process using dynamic data and tags. We attempt to give a summary of the state-of-the-art deep RL algorithms in this research. The first section is all about DRL and its basic information. In the next section, there is a brief description of Model-based vs model-free DRL algorithms and then there are past models that are examined and analysed based on certain parameters followed by a discussion highlighting the shortcomings and benefits of those models. In the end, there is a proposed model that combines all the suggestions and covers all the shortcomings of previous models.

Deep Reinforcement Learning (DRL)

Model Free DRL

Development done in deep reinforcement learning is still in starting phase and a lot is needed to be done. Most academic researchers focus on the static and deterministic environment where states have been fully observed and are static as well [1]. Thus, the majority of the RL techniques are model-free. Model-free RL makes estimates of the system state, value function, and the reward function of an agent using a large number of samples for action policy optimization that is aimed to achieve more rewards. Due to the least complex implementation and open resources, model-free RL has attracted many more scholars for carrying out further research in this field [2].

Model-Based DRL

Things are made easier with known transition dynamics between states and future actions. Such dynamics are called models. Model-based methods include algorithms that learn the transitions to decide which new state (st+1) will be selected after performing an action at in current state st. These methods will figure out how to select the actions. In short, these algorithms learn models of system dynamics and then optimal control strategy for choosing such next actions. Model-based RL algorithms are developed from optimal control methods. In comparison with model-free RL methods, the model-based RL algorithms learn a value function or next policy using a data-efficient manner and they do not need to continuously interact with the environment. This may lead to difficulty in model identification and cause an inaccurate description of the real environment. However, it may suffer from the

issue of model identification and lead to an inaccurate description of the real environment [3]. If there is a simulator available but the cost of performing the simulations is too much then opt for model-free off-policy algorithms e-g NAF, DDPG [4], and SQL [5]. There are multiple choices for all the policy-based model-free methods in RL that can be extra critic based or Q-Learning-based. In case we don't have any simulator, the main question arises that how long the acceptable waiting period should be. In this case, if the waiting period is not much prolonged, then the model-based algorithms are a better option e-g probabilistic ensembles with trajectory sampling (PRTS) [6] and guided policy search (GPS) [7]. In another case, model-free off-policy algorithms can be used that make use of some assumptions and can be made more generic and less domain-specific.

Past contributions

Model-based DRL algorithms

Many RL methods have been proposed in the past for solving many problems related to healthcare and specifically cancer treatment. [8] proposed a method that is a combination of fuzzy sets and reinforcement learning called fuzzy reinforcement learning for controlling the growth of cancer cells. They have used two different drug dosages to reduce the population of cancer cells. The main limitations of the research are that they have used arbitrary parameters selected by the trial-and-error method. Another research conducted by [9] proposed a single-agent deep reinforcement learning model for weight tuning. They used Epsilon greedy approach and solved the optimization problem by minimizing the sum of doses fed to four critical organs with doze balancing. Their proposed method is high-dose-rate brachytherapy (HDRBT) for cervical cancer.

In their research, [10] a hybrid model is proposed that combines a fuzzy system with a neural network for lung cancer nodule detection. The proposed model is only effective for certain types of data and patient samples. They used MATLAB toolbox to simulate their model and x-ray images of lung nodules as a dataset. Results gave 92.56% accuracy. In another research, a Parameter Tuning Policy Network (PPTN) was trained using the procedure of end-to-end reinforcement learning [11]. They tried this method to improve the image quality of CT scan with this generic framework. The drawback was that this method had to wait for an iterative process to further tune its parameters. This method focused on policy optimization.

Another model uses Markov Decision Process for optimization of lung cancer detection by training a dynamic Bayesian network and then discovers an expert's decision-based reward function through the inverse reinforcement learning method. They unfortunately could not handle the stochastic nature of patient responses and the model is not suitable for lung nodule images taken at random frequencies. They used NLST data and simulated on MATLAB toolbox [12]. A Q-Ranking approach was used to detect cell lines' sensitivity to anti-cancer drugs. This method integrates various predictive algorithms and then chooses a suitable algorithm for a certain application. Batch reinforcement learning is used to identify the ranking policy [13]. The model has a limitation of not being scalable and is not generic. NCI-DREAM7 dataset is used for addressing policy optimization problems.

Model Free DRL Algorithms

A value-based single-agent reinforcement learning method with TD and Q Learning was proposed for tumor localization of lung cancer [14]. The authors categorized lung cancer types and described the characteristics of each one. The most challenging part of the application of this RL method was to define a suitable reward function for updating the Q-value for each performed action. Another Q-Learning-based approach was used to control the drug dosing during chemotherapy treatment. A scaled error value of reward function based on the count of cancer and normal cells. The authors in [15] applied their model to patients from different age groups and for each case, a different RL agent was developed to control each case. The main limitation of this study was that the proposed model was not generic to be applied to all the cases, but they needed to be specific according to each patient's characteristics.

The model addressed the application of antiangiogenic therapy for the reduction of tumor volume [16]. The volume of tumor is considered as a reward function if the error value is equal to or less than 1. They tested their model on just a single patient record using Silico Simulations. Due to the higher complexity of this model, model-based controllers are impossible to use. Also, the model could not handle the stochastic nature of the patient's dynamics [17].

[18] also proposed a model for controlling automated radiation adaption for lung cancer. A DRL approach with three component-based neural network framework with a Deep Q-network is developed. A limited number of samples were used, and the reward function was customized for each patient. 114 NSCLC patient data was used. In their research [10], proposed

another model for lung cancer nodule detection with a combined framework of fuzzy systems and neural networks. Again, the model is only suitable for certain patient samples and not generic to be adopted for all patients. They used the MATLAB toolbox for carrying out model simulations and optimized the reward function. [19] proposed another model for reward function optimization for modeling the optimal drug dosing in cancer treatment. 15 patients were simulated for model application with MATLAB Simulations. They also proposed a model by [20] using integral reinforcement learning for optimal drug dosing for a provided performance measure. Only 10 patients were simulated using MATLAB simulations. whereas stochastic parameters like nonlinearities, time delays, and nonnegative constraints are not handled with this model [21].

Analysis

In this section, many past related contributions have been discussed and analyzed based on certain parameters as shown in Table 1.

Table 1. Model-based drl algorithms

Ref#	Algorithm	Application addressed	Limitations/ Assumptions	Gap Addressed
[8]	free model-based fuzzy reinforcement learning. Combination of integrated fuzzy sets and reinforcement learning.	Control Cancer Cells growth. reducing cancer cell population by using two different drugs dosages	The parameters mentioned in this algorithm are arbitrary and selected by trial and error.	Policy Optimization
[9]	deep reinforcement learning (DRL) based approach to accomplish the weight-tuning. Epsilon greedy process. The optimization problem minimizes a weighted sum of doses to four critical organs with doze balance.	high-dose-rate brachytherapy (HDRBT) for cervical cancer.	The VPN approach is potentially applicable to external beam therapy.	epsilon greedy for reward optimization.
[10]	model based on a composition of fuzzy system combined with a neural network.	Lung cancer nodules detection.	Model works good for a specific type of data and samples.	
[11]	Parameter Tuning Policy Network (PTPN) trained via an end-to-end reinforcement learning procedure	a general framework on the development of a strategy to improve CT image quality	only considered images with a relatively low resolution in a small number of cases. PTPN has to wait for the iterative process to finish, before it can adjust parameters	Policy optimization
[12]	Markov decision process that simultaneously optimizes lung cancer detection. trained a dynamic Bayesian network as an observational model and used inverse reinforcement learning to discover a rewards function based on experts' decisions.	Lung Cancer	a discrete time model may not be well-suited for instances of imaging observations at irregular frequencies. Stochastic nature of patient screening probabilities.	Reward Function
[13]	Q-Rank, to predict the sensitivity of cell lines to anti-cancer drugs Q-Rank integrates different prediction algorithms and identifies a suitable algorithm for a given application. models are automatically ranked based on non-scored meta-features. The ranking policy is identified using batch reinforcement learning. The top-ranked model(s) is (are) used to predict drug responses	Predict drug sensitivity for therapy of cancer.	Model is not scalable or generic.	Policy optimization

Table 2. Model-free DRL algorithms.

ef#	Algorithm	Application addressed	Limitations/ Assumptions	Gap Addressed
[14]	Value-based reinforcement learning approach (TD and Q-Learning)	Tumour localization of Lung cancer	the most challenging issue of applying deep reinforcement learning models to lung cancer treatment is to define an appropriate reward function that is used to update the Q-value for each action.	
[15]	Q-learning-based approach for the closed-loop control of drug dosing related to chemotherapy. Based on cancer and normal cells count, scaled value of the error is used in the reward function.	Cancer in different age groups. different RL agents are developed to address the drug-dosing control in each of these cases	different RL agents need to be trained to account for the patient characteristics of different patient groups.	
[16]	Q-Learning method for drug dosing closed-loop control. 30,000 training episodes are considered. Tumor volume is used in the reward function as an error less than or equal to 1.	Antiangiogenic therapy for Tumor volume reduction.	since the model has high complexity, it is impossible to use model-based controllers.	
[17]	RL based Q Learning model free method for closed loop control of cancer chemotherapy drug dosing.	control of cancer chemotherapy drug dosing.	Does not handle patient dynamics.	Policy Optimization
[18]	a three component neural networks framework with a deep Q-network (DQN) was developed for deep reinforcement learning (DRL) of dose fractionation adaptation.	Automated radiation adaptation in lung cancer.	customization of the reward function if individual cases were to be considered. Limited sample set.	Reward function customization to observe policy changes
[10]	model based on a composition of fuzzy system combined with a neural network.	Lung cancer nodules detection.	Model works good for a specific type of data and samples.	
[19]	Q-Learning algorithm using different reward functions to model different constraints in cancer treatment	Optimal chemotherapy drug dosage for cancer treatment	Specific reward function for certain scenarios. Scale value of error in reward function.	Reward function optimization
[20]	online integral reinforcement learning (IRL) algorithm is designed to provide optimal drug dosing for a given performance measure.	sedative drug dosing to maintain a required level of sedation.	Numerical results are presented using only 10 simulated patients. Stochastic nature like time delays, nonlinearities, and nonnegative constraints are not handled.	Reward optimization

Discussion

This survey has been divided into two types of DRL Algorithms i-e Model Based DRL and Model Free DRL. Both DRL methods have achieved some levels of success but there are still a lot of optimizations that need to be done and there are many limitations of each finding that need to be addressed as well. There are many shortcomings of the past research in the form of limitations. These are summed up here to identify the research gaps for future use:

- Models proposed do not handle the stochastic nature of patient responses as they are tested on a few patients or small datasets with specific known body characteristics.
- Proposed models are not generic in nature i-e they do not handle a variety of cancer patients but are designed too specific for a few scenarios thus limiting the use of these models in vast levels of treatments.
- Finally, the models are all designed as single agents. They can also be implemented in a multiagent scenario to get a better performance.

Many future research directions can be concluded from these limitations of past DRL models for cancer treatment. Future DRL model should comprise of:

- Handle stochastic nature of cancer disease and handle noisy or incomplete states.
- Deal with credit assignment problem.
- Should be ideal in cost of exploration and exploitation.

- Reward function should be adaptive and not the misleading one.
- Algorithm should be model free and based on multi modal approach.

Proposed Solution Model

The proposed solution model addresses the following limitations as evaluated in our critical analysis:

- Multi agent creates a knowledge base that helps in reducing information overlaps and voting system creates more reliable and contributing agent is highlighted as well.
- Deep RL will provide more coverage to multidimensional environments with its NN learning the policies and optimizing the reward without huge Q-Table.
- Hyperparameter tuning using NN will help in balancing the cost of exploration and exploitation.

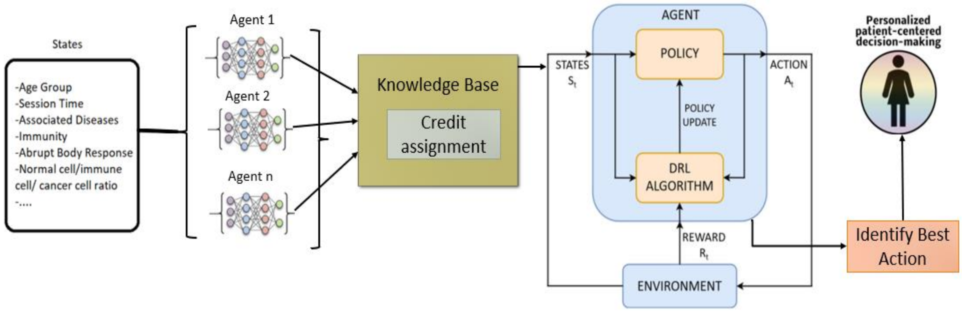

Figure 1. Proposed Multi-Agent DRL Model for Cancer Patients Treatment

Experiment

In this research, we have proposed a multi-agent DRL model that can support decision-making for the treatment of cancer patients in stochastic clinical scenarios. Stochastic scenarios are the ones that are partially observable and thus handled through stochastic policies where the agent assigns a probability to each action and selects the action based on the probability value. The approach used is a model-free Deep Q-Network algorithm. When the environment is straightforward the Q-Learning algorithm performs exceptionally well. However, the database becomes too large, and tabular methods are no longer useful when there are billions of possible unique states and thousands of different actions for each of them. [22] created the Deep Q-Networks (DQN) algorithm to address this. Deep neural networks (DNNs) and the Q-learning method are combined in this algorithm. DNN is a non-linear function approximator. Thus, it is used in DQN to approximate Q values by replacing the need of a table to store Q values which is very complex and nearly impractical in the case of a large number of states [23].

Dataset

Dataset is taken from a famous repository Kaggle Dataset includes sociodemographic information about cancer patients, signs and symptoms of the disease, histological and imaging characteristics, and TNM stage information. The records also include the patient's diagnostic procedures and available treatments Our model is trained and tested with Dataset of Colorectal cancer patients.

State Space

State space consists of around 25 discrete values. These states for Colorectal Cancer patients are listed below:

Table 1. Input states dataset1

Sno.	**States**	**Data type**	
1	Age	Integer	
2	Diarrhoea	Boolean	**Value**
3	Constipation	Boolean	0 - 100
4	blood_stool	Boolean	0,1
5	abdominal_pain	Boolean	0,1
6	weight_loss	Boolean	0,1
7	Fatigue	Boolean	0,1
8	Biopsy	Boolean	0,1
9	Colonoscopy	Boolean	0,1
10	Imaging	Boolean	0,1
11	US	Boolean	0,1
12	CXR	Boolean	0,1
13	Location	Ordinal Categorical	1,2,3,4
14	Hist_type	Ordinal Categorical	1,2
15	Hist_grade	Ordinal Categorical	1,2,3
16	TNM_stage	Ordinal Categorical	1,2,3,4
17	Clinical_stage	Ordinal Categorical	1,2,3
18	Lymph_node	Boolean	0,1
19	Vascular_Invasion	Boolean	0,1
20	Residual_Tumor	Boolean	0,1
21	CEA_baseline	Ordinal (0 to 10)	0 to 10
22	Dist_metastasis	Boolean	0,1
23	Liver	Boolean	0,1
24	Lung	Boolean	0,1
25	Peritonuem	Boolean	0,1
25	Peritonuem	Boolean	0,1

Action Space

Action space in this scenario consists of 3 discrete values and any of their combination will be an action taken to interact with the environment. Three types of actions can be taken to interact with the environment. The actions for the Colorectal cancer dataset include:

TABLE 2. ACTIONS SET DATASET1

Sno	**Action**	**Data type**	**Value**
1	Chemotherapy	Boolean	0,1
2	Surgery	Boolean	0,1
3	Radiotherapy	Boolean	0,1

Reward

The reward is considered a binary value of 0 or 1 in case if patient stays alive or does not survive.

Custom Environment Creation

To combine states and actions, we have created a custom environment with the help of the Python library OpenAI Gym. A consistent interface for engaging with environments is offered by OpenAI Gym, making it simpler to compare and replicate results across various algorithms and academic articles. Therefore, if everything has the same structure, you can easily train and test multiple settings with different algorithms. OpenAI Gym is an extremely user-friendly and flexible library for creating real-time custom environments in addition to gaming environments offered by the gym [24]. In this environment a patient comes with some symptoms that are considered input states, those states are passed to the agent to train against the three actions decided above. Based on the reward function, policies are built and NN optimizes the training agent's abilities by helping out in learning faster and applying those built policies for test data.

Results: In this research, DRL agents have been passed through multiple trials and the returned accuracy values are recorded with varying numbers of agents as well. The results recorded have been stored in Table 5.

Table 3. Trial results

Parameters	**Agents=2**	**Agents=3**	**Agents=4**
Trials	500	500	250
Timestep	1000	2000	3000
Learning_rate	0.000237875	0.009897	0.004275607782
buffer_size	56288	39537	10495
Gamma	9102253735	0.902997	0.8796811109
exploration_fraction	0. 4484000	0.1663777	0.14679955981
batch_size	1510	1749	735
Accuracy	0.91	0.94	0.94

Results show that using three agents we can achieve an accuracy value of 94% and there is no additional change in accuracy if we increase the no of agents.

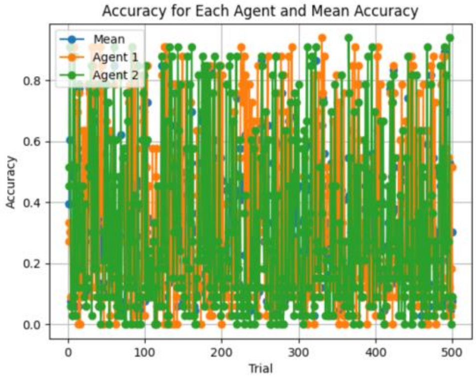

Figure 1. Accuracy Plot with 2 Agents

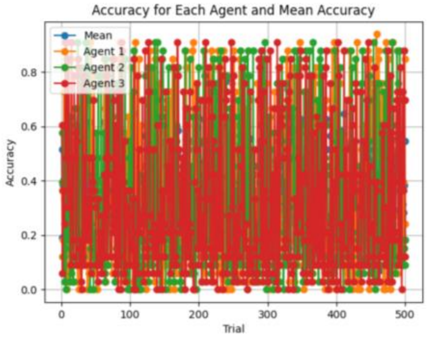

Figure 2. Accuracy Plot with 3 Agents

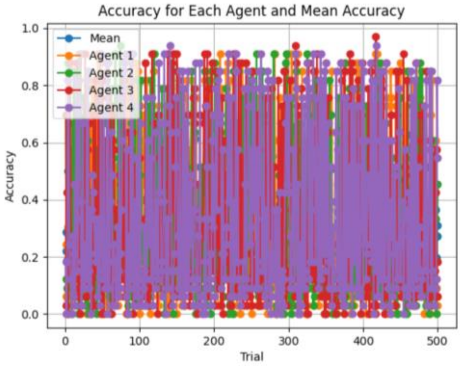

Figure 3. Accuracy Plot with 4 Agents

Conclusion

There are many methods proposed to solve automated disease detection and drug dosing through deep reinforcement learning. Most of the methods proposed so far are so immature or they are not generic with lots of limitations that we need to contribute much in this field. This analysis evaluates each past approach to deal with the stochastic nature of the DRL environment and treatment automation with the help of certain parameters. Finally, it proposes an experimented model that fills up the required gaps and this model proves to provide results with an accuracy of 94%.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: 10.1145/3065386.

[2] "Sutton & Barto Book: Reinforcement Learning: An Introduction." Accessed: May 19, 2024. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[3] M. Hessel et al., "Rainbow: Combining Improvements in Deep Reinforcement Learning," 32nd AAAI Conf. Artif. Intell. AAAI 2018, pp. 3215–3222, Oct. 2017, doi: 10.1609/aaai.v32i1.11796.

[4] S. Gu, T. Lillicrap, U. Sutskever, and S. Levine, "Continuous Deep Q-Learning with Model-based Acceleration," 33rd Int. Conf. Mach. Learn. ICML 2016, vol. 6, pp. 4135–4148, Mar. 2016, Accessed: Jun. 02, 2024. [Online]. Available: https://arxiv.org/abs/1603.00748v1

[5] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement Learning with Deep Energy-Based Policies," 34th Int. Conf. Mach. Learn. ICML 2017, vol. 3, pp. 2171–2186, Feb. 2017, Accessed: Jun. 02, 2024. [Online]. Available: https://arxiv.org/abs/1702.08165v2

[6] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models," Adv. Neural Inf. Process. Syst., vol. 2018-December, pp. 4754–4765, May 2018, Accessed: Jun. 02, 2024. [Online]. Available: https://arxiv.org/abs/1805.12114v2

[7] S. Levine and V. Koltun, "Guided Policy Search." PMLR, pp. 1–9, May 26, 2013. Accessed: Jun. 02, 2024. [Online]. Available: https://proceedings.mlr.press/v28/levine13.html

[8] M. Goharimanesh, A. Akbari, and B. Lotfi, "On using fuzzy reinforcement learning to control the cancer cells," J. Cell. Immunother., vol. 1, pp. 34–35, 2015, doi: 10.1016/j.jocit.2015.10.036.

[9] C. Shen, Y. Gonzalez, H. Jung, L. Chen, N. Qin, and X. Jia, "Automatic Inverse Treatment Planning for Cervical Cancer High Dose-Rate Brachytherapy via Deep Reinforcement Learning," Int. J. Radiat. Oncol., vol. 102, no. 3, p. e540, Nov. 2018, doi: 10.1016/j.ijrobp.2018.07.1510.

[10] G. Capizzi, G. Lo Sciuto, C. Napoli, D. Polap, and M. Wozniak, "Small Lung Nodules Detection Based on Fuzzy-Logic and Probabilistic Neural Network with Bioinspired Reinforcement Learning," IEEE Trans. Fuzzy Syst., vol. 28, no. 6, pp. 1178–1189, Jun. 2020, doi: 10.1109/TFUZZ.2019.2952831.

[11] C. Shen, Y. Gonzalez, L. Chen, S. B. Jiang, and X. Jia, "Intelligent Parameter Tuning in Optimization-based Iterative CT Reconstruction via Deep Reinforcement Learning," IEEE Trans. Med. Imaging, vol. 37, no. 6, pp. 1430–1439, Nov. 2017, doi: 10.1109/TMI.2018.2823679.

[12] P. Petousis, A. Winter, W. Speier, D. R. Aberle, W. Hsu, and A. A. T. Bui, "Using Sequential Decision Making to Improve Lung Cancer Screening Performance," IEEE access Pract. Innov. open Solut., vol. 7, pp. 119403–119419, 2019, doi: 10.1109/ACCESS.2019.2935763.

[13] S. Daoud, A. Mdhaffar, M. Jmaiel, and B. Freisleben, "Q-Rank: Reinforcement Learning

for Recommending Algorithms to Predict Drug Sensitivity to Cancer Therapy," IEEE J. Biomed. Heal. Informatics, vol. 24, no. 11, pp. 3154–3161, Nov. 2020, doi: 10.1109/JBHI.2020.3004663.

[14] Z. Liu, C. Yao, H. Yu, and T. Wu, "Deep reinforcement learning with its application for lung cancer detection in medical Internet of Things," Futur. Gener. Comput. Syst., vol. 97, pp. 1–9, Aug. 2019, doi: 10.1016/J.FUTURE.2019.02.068.

[15] R. Padmanabhan, N. Meskin, and W. M. Haddad, "Reinforcement learning-based control of drug dosing with applications to anesthesia and cancer therapy," Control Appl. Biomed. Eng. Syst., pp. 251–297, Jan. 2020, doi: 10.1016/B978-0-12-817461-6.00009-3.

[16] P. Yazdjerdi, N. Meskin, M. Al-Naemi, A. E. Al Moustafa, and L. Kovács, "Reinforcement learning-based control of tumor growth under anti-angiogenic therapy," Comput. Methods Programs Biomed., vol. 173, pp. 15–26, May 2019, doi: 10.1016/J.CMPB.2019.03.004.

[17] R. Padmanabhan, N. Meskin, and W. M. Haddad, "LEARNING-BASED CONTROL OF CANCER CHEMOTHERAPY TREATMENT," IFAC-PapersOnLine, vol. 50, no. 1, pp. 15127–15132, Jul. 2017, doi: 10.1016/J.IFACOL.2017.08.2247.

[18] H. H. Tseng, Y. Luo, S. Cui, J. T. Chien, R. K. Ten Haken, and I. El Naqa, "Deep reinforcement learning for automated radiation adaptation in lung cancer," Med. Phys., vol. 44, no. 12, pp. 6690–6705, Dec. 2017, doi: 10.1002/MP.12625.

[19] R. Padmanabhan, N. Meskin, and W. M. Haddad, "Reinforcement learning-based control of drug dosing for cancer chemotherapy treatment," Math. Biosci., vol. 293, pp. 11–20, Nov. 2017, doi: 10.1016/J.MBS.2017.08.004.

[20] R. Padmanabhan, N. Meskin, and W. M. Haddad, "Optimal adaptive control of drug dosing using integral reinforcement learning," Math. Biosci., vol. 309, pp. 131–142, Mar. 2019, doi: 10.1016/J.MBS.2019.01.012.

[21] V. Mnih et al., "Playing Atari with Deep Reinforcement Learning," Dec. 2013, Accessed: Jun. 02, 2024. [Online]. Available: https://arxiv.org/abs/1312.5602v1

[22] V. Mnih et al., "Human-level control through deep reinforcement learning," Nat. 2015 5187540, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

[23] V. Vidyadhar, R. Nagaraj, and D. V. Ashoka, "NetAI-Gym: Customized Environment for Network to Evaluate Agent Algorithm using Reinforcement Learning in Open-AI Gym Platform," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 4, pp. 169–176, 2021, doi: 10.14569/IJACSA.2021.0120423.

[24] G. Brockman et al., "OpenAI Gym," Jun. 2016, Accessed: Jun. 02, 2024. [Online]. Available: https://arxiv.org/abs/1606.01540v1