Replication study on reinforcement learning techniques for hepatocellular carcinoma detection with extension to deep learning MDPs

Patrick de Guzman

Section 1. Introduction

Hepatocellular Carcinoma (HCC) is one of the most common forms of liver cancer. Detecting cancers in their early stages is paramount to improving patient survival rates, however, many healthcare settings operate under constrained screening resources which presents a resource allocation problem suitable for reinforcement learning (RL) methods. In the reference paper, Lee et al. (2014) assesses 3 RL approaches in the problem of constrained resource allocation as it relates to the allocation of screenings to a population at risk of HCC.

The aim of this paper will be to perform a replication study of their findings using synthetic data. Data is generated using a simple sampling algorithm based on the population parameters of the original HCC population dataset. In addition, an extension toward Markov Decision Processes (MDPs) will be explored using enhanced state-space definitions and a modified deep Q-learning network (DQN) will be trained against these enhanced state-spaces to assess the relative performance against the original models proposed by the authors.

1.1 Reinforcement Learning Background

Reinforcement learning (RL) is a class of machine learning methods, typically used for problems that require decisions to be made by an agent in a particular environment or state-space to maximize (or minimize) some objective. RL decision-makers (agents) interact with their environment by taking actions which produce rewards or signals to the RL agent, helping to inform better decision-making in the future by learning and storing its experiences in different states. The environment is typically a modelled version of the problem at hand, and individual states represent subsets of this environment that inform the agent on what actions to take based on its past experiences.

RL algorithms vary by their methods for choosing actions, approximating the values of states and actions relative to their primary objective, learning (i.e., storing past experiences), and their ability to observe the state or environment that they occupy. The applications of various RL algorithms depend on the problem definition and objective function.

An example of such variation (relevant to the paper at hand) involves choosing actions that vary by method of ‘exploration’, which is the degree to which an agent will select a non-optimal action. By assessing each action in each state, an RL agent makes estimates on which actions are most optimal to take based on a value function, and standard logic would dictate that an agent should always select the best action in each state. However, this lack of exploration causes problems in the long-term decision-making horizon since there is a chance that other (possibly more favorable) actions will never get chosen because they may not present

immediate rewards to the RL agent but would present higher total rewards (value) in the long run if selected enough times. Lack of exploration effectively places an RL agent in a theoretical 'box', preventing it from learning of other optimal actions by optimizing for local, short-term rewards.

Another variation of RL algorithms involves the extent to which state-spaces are directly observable by agents, along with the degree to which an agent can take an action that influences its transition to a new state. Frameworks like Hidden Markov Models (HMMs) do not have observable states and give the agent no control over transitions to new states through its actions, where the objective is usually to predict the current occupied state using other external observations and probability statistics. On the other hand, Markov Decision Processes (MDPs) will be discussed further in this paper which do have observable states and provide an agent with the ability to influence the next state through its current chosen action.

1.2 Reinforcement Learning in Resource Allocation

Resource allocation presents itself as an RL problem because there is a decision maker that must allocate resources within constraints for cost, time, or other factors like operational efficiency. In this situation, the agent can be thought of as a doctor deciding, from a pool of at-risk subjects, who to allocate screenings to. The action is simply the decision to allocate a screening test to a particular subject, and the environment is the healthcare setting with its variables for the number of total subjects, number of screening resources available during each decision-making period, and information regarding state spaces to inform the agent of its decisions. By using these risk factors to develop a risk score for patients, an RL agent is better informed in the screening test allocation problem with the logic that subjects with higher risk scores should be tested first.

1.3 Structure of the paper

Section 2 of this paper will briefly discuss the primary ways in which RL methods vary in the resource allocation problem in healthcare. Next, Section 3 will describe the data used for experimentation, the RL methods to be compared, along with the simulation logic employed to run all comparative methods. Any deviations from the original simulation logic of the reference paper will also be highlighted here to lay ground for possible deviations in our results against the original. Section 4 will discuss the experimental setup as it pertains to relevant simulation assumptions made and the hyperparameters selected, both on the simulation-level and model-level. Simulation results will be compared against the original paper, and the performance of the deep RL MDP method will also be compared against other RL methods. Lastly, the key learnings, replication study limitations, and possible future directions for the resource allocation problem and deep RL methods will be discussed in Section 6.

Section 2. Literature Review

The following section will briefly highlight the main differences between various approaches applied to the resource allocation RL problem in healthcare screening. Methods have primarily varied in the formation of agent objectives and data gathering (or generation).

2.1 Objective Definition

In resource allocation, there can exist a multitude of objectives for RL algorithms to maximize. Agent objectives have been proposed to either maximize cost-effectiveness of screening strategies or maximize the life years 'gained' from successful early detection of disease. An example of a combination of these objectives can be found in a study for screening of Colorectal Cancer (Frazier et al., 2000) where cost metrics are defined as 'dollars per life-year saved'. The objective that Lee et al. (2014) explore is the maximization of both the proportion of all cancers detected in the early stage (relative to all cancer patients in the dataset), as well as the proportion of screening resources spent on cancer patients (as opposed to non-cancer patients) to effectively minimize wasted resources.

2.2 Available Data

The usage of data for simulation has also varied in the literature. The use of historical data (i.e., real patient data) has the advantage of more generalizable results since the data is from a real population. However, such data is difficult to obtain and generate at scale, which creates a limitation for RL model learning. On the other hand, simulation-based methods can be used to generate synthetic data at scale by querying population parameters, with the disadvantage that any results reported from such data is less reliable for generalization to unseen patient data. An example of a simulation-based method can be found in a study of ovarian cancer screening (Urban et al., 1997).

Lee et al. (2014) utilizes historical data from a Hepatitis C treatment trial which followed 1050 patients over an average of 5 years. Given that the trial data is not publicly available, synthetic data is generated for the purposes of this study, and the method for generation, along with limitations in assumptions, will be discussed in the following section.

Section 3. Methods

3.1 State-space & Risk-factor definition

The authors utilize several risk factors for developing HCC which include the following: age, black ethnicity, blood platelet count, smoking status, alkaline phosphatase, esophageal varices, and alphafeto-protein (AFP) levels. Importantly, it was noted that AFP levels are not directly correlated with risk of developing HCC, but the observed fluctuations in AFP over time provide for an important signal in the state-space that helps inform prediction of HCC development.

These risk factors are combined to develop a single cumulative risk probability of developing HCC which is then used as the main guide for the agent to take actions upon. All

static risk factors are formed into a vector B_i, while non-static factors like the standard deviation of AFP fluctuation and the least squares estimate of AFP rise are incorporated into separate variables SD_i and RR_i, respectively. As a result, the risk probability (score) is formulated as:

$$P(\text{HCC})_i = [1 + exp(-c_1 B_i - c_2 SD_i - c_3 RR_i)]^{-1}$$

The authors learn c coefficients through logistic regression for each variable. Therefore, the state space at any given time 't' is defined by the following variable:

$$(B_i, \hat{SD}_{i,t}, v_{i,t}, \hat{RR}_{i,t}, w_{i,t})$$
$$\forall i = 1, .., n, \forall t = 0, 1, 2, ...T$$

In the above, n represents the number of possible subjects in the system, while v and w represent the variance of SD_i and RR_i, respectively. Using this risk factor formulation, the agent is then capable of ranking subjects and taking actions to decide which subjects to screen, with some RL methods applying adjustments to this state space variable to employ exploration.

3.2 Action-selection Methods: Varying Exploration

Lee et al. (2014) assesses 3 primary RL policies, each varying in their degree of action exploration: epsilon-greedy, interval estimation, and Boltzmann estimation. These methods are compared against a pure exploitation (i.e., no exploration) policy that simply selects the top 'k' patients to be screened based on risk scores in descending order.

The pure exploitation policy ranks subjects by the risk score $P(HCC)$ and selects the top k patients to screen based on current resources in time t. This method is intuitive since the subjects with the highest risk scores should be screened first, but as discussed, methods with no exploration are limited in their learning ability since they are not exposed to new state spaces and actions to enable comprehensive learning of the environment.

3.2.1 Epsilon-greedy

The epsilon-greedy approach utilizes a parameter ϵ between 0 and 1, where the decision maker selects the top $(1 - \epsilon) * k$ patients, reserving $\epsilon * k$ patients to be screened randomly to fulfill exploration.

3.2.2 Interval estimation

Interval estimation involves adjusting the risk score to account for the variability in the system's confidence in each subject's calculated risk scores. In effect, this method encourages the exploration of screening for subjects whose risk score is less certain using a new parameter z to dictate the degree of exploration based on the variance of subject scores. The authors formulate the following adjusted risk score to be used in ordering the patients for selection:

$$c_1 B_i + c_2(\hat{SD}_{i,t} + z \cdot \sqrt{v_{i,t}}) + c_3(\hat{RR}_{i,t} + z \cdot \sqrt{w_{i,t}})$$

3.2.3 Boltzmann exploration

Lastly, the authors propose a Boltzmann exploration strategy where, instead of simply ordering patients by score and selecting the top k, all subjects are given a probability of being selected, with that probability being weighted by the risk score itself. This probability is formulated as follows, where x represents the original risk score formulation:

$$\frac{e^{x_{i,t}/\tau}}{\sum_{i'=1}^{n} e^{x_{i',t}/\tau}}$$

$$x_{i,t} = (c_1 B_i + c_2 \hat{SD}_{i,t} + c_3 \hat{RR}_{i,t})$$

3.3 Contribution: MDP Formulation, Deep RL

Lee et al. [1] propose to combine risk factor scores into a probability metric which is then used to guide agent decision-making by various policy exploration methods. Aggregating individual risk factors into a single metric prevents the decision-maker from having direct knowledge of the state of each subject. On the other hand, a formal MDP can be formulated which enhances the state-space definition. An MDP through a deep RL implementation will be formulated to assess the validity of enhanced state-spaces for improving predictive performance.

3.3.1 MDP formulation: Timing

The following timing markers can ensue for an MDP-based simulation of the resource allocation problem per the decision-maker's (DM) perspective:

- DM holds a panel of size n out of the entire population of study N, with k available screening resources for decision epoch t.
- DM observes subjects individually, with their risk factors presented as state s.
- Decision maker determines action a to screen or not screen the subject until the screening resources available k are used up in period t.
- After k resources are allocated, rewards are allocated based on the revealed cancer states of screened subjects (early-stage, late-stage, cancer-free) and the proportion of cancer patients screened against the total available screenings (true positive rate) is calculated as a reward metric.
- Subjects in panel are updated as follows
    - Subjects with cancer who are screened exit the pool and new subjects are sampled as replacement
    - Subjects without cancer who are screened remain in the pool, but DM updates knowledge of these subjects with additional AFP readings (standard deviation, rate of rise) to be used in state-space variables for next decision epoch t
    - Subjects not screened

3.3.2 MDP formulation: Decision Epochs, States, Actions, Rewards

The decision epochs represent a designated planning horizon H until terminal time T as follows. In any given decision epoch, the DM will allocate k screenings to its current panel of subjects:

$$H = \{1, \dots, T\}$$

States are presented one-by-one to the DM during an epoch for each subject in the panel. The state-space vector consists of the following:

- Current subject's static risk factors in vector B (containing all risk factors i including age, black ethnicity, smoker status, alkaline phosphatase levels, and esophageal varices)
- Current subject's dynamic risk factors $SD_{i,t}$ and $RR_{i,t}$ (standard deviation and rate of rise of AFP level readings up to time t)
- Current remaining screenings resources at current stage r of the decision epoch t represented by $k_{r,t}$.

$$s_{i,t} = [B_i, \dots, B_I, SD_{i,t}, RR_{i,t}, k_{r,t}]$$

Available actions to the decision maker are denoted with 0 representing a subject not selected and 1 representing a subject selected for screening:

$$A = \{0,1\}$$

In each decision epoch t, rewards are determined at each review step r (i.e., iterating through the randomly shuffled list of subjects in the DM's current panel and reviewing each one by one). After the DM is shown a subject's state-space vector, an action is selected to screen or not screen the patient and the reward is calculated as follows:

- If the selected action is to screen the subject, set the reward equal to the gain or loss in the running true positive rate.
    - The true positive rate at screening stage r is represented by TP_r and is calculated as the number of cancer patients in the list of patients that have been screened so far in the current decision epoch, divided by the total number of screenings allocated so far.
    - The reward is calculated as the change in the true positive rate between screening steps r and $r + 1$ (see below). This value is also multiplied by a reward multiplier m as the absolute value of each reward becomes smaller as the DM reviews more subjects in the panel (i.e., the TP rate oscillates to a smaller degree).
    - If the DM screens a subject who does have cancer, the running true positive rate in step $r + 1$ will increase and the reward formula will return a positive value. However, if the DM screens a subject who does not have cancer, the running true positive rate will decrease, and the reward formula will return a negative value for the action.

- If the selected action is to not screen the subject, set the reward equal to the gain or loss in the running true negative rate.
    - This follows similar logic to the above, but with the replacement of the true negative rate represented by TN_r and the reversal of the values. The reversal is to maintain to the neural network that positive values should indicate a signal to screen and negative values should indicate a signal to not screen subjects.

$$r(a = 1) = m(TP_{r+1} - TP_r)$$

$$r(a = 0) = m(TN_r - TN_{r+1})$$

3.3.3 MDP formulation: Deep Learning Policy

A modified deep Q-learning network (DQN) implementation of this MDP can be formulated to determine whether any given subject should be screened based on the state-space (similar to the original RL methods discussed). However, a distinction is made where the DQN policy bases its decision to screen (or not screen) on the predicted value of screening each subject (instead of the aggregated risk score metric per the comparative RL methods). A neural network is initiated with inputs corresponding to the enhanced state-space vector containing all risk factors (static and dynamic) and the number of remaining screening resources to allocate per iteration t in review step r. Through a series of densely connected layers, the neural network performs regression to predict the value of screening the current subject based on the state-space input. A positive prediction value indicates that the agent should screen the current subject, while a negative prediction value indicates the opposite.

The weights of the DQN are initialized randomly. The DM then shuffles its panel of subjects and iterates through them by using the DQN policy network to determine the appropriate action for screening. Once an action is taken, a reward is calculated based on the perceived value of the action (discussed in Section 3.3.2). This combination of state, action, and reward is then stored in the agent's 'memory' which represents a cache of predetermined size. Given that the MDP contains several continuous variables in its state-space vector, tracking all discrete states is not feasible, thus, requiring a limited memory cache. A plot of the DQN model is presented in the below figure:

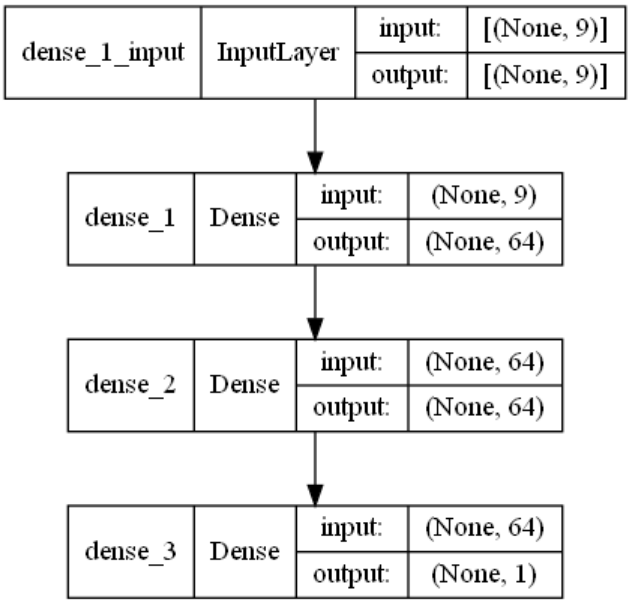

Figure 1: DQN model layout

To enable the agent to learn from its experience and improve prediction performance, experience replaying is employed to train the neural network on a random subset of its past experiences. The memory cache is randomly queried for a predetermined batch size, and this batch of data is fed into the policy neural network as training instances to adjust layer weights in preparation for the next decision epoch.

For each instance, the true prediction value (i.e., 'y') that is passed with the state-space feature input (i.e., 'x') is calculated using a gamma learning rate to estimate a new Q-value based on the model's current estimate and actual reward. Below is a formulation of this learning update, where $R_{s,t}$ represents the current estimate of the reward value before update, and R_a represents the actual reward received from the instance in memory:

$$R_{s,t+1} = R_{s,t} + \gamma(R_a - R_{s,t})$$

3.4 Objective Function

The following are the statistics measured during simulation which relate to the overall objective function proposed by the authors: the number of early-stage cancers detected during simulation E, the number of late-stage cancers detected L, and the number of screenings spent on cancer patients X.

In simulation, a model initializes a set of patients for consideration of screening, and a policy is applied (based on the 3 exploration methods discussed previously). After selection for screening, the cancer state of these subjects is revealed to the RL agent (early-stage, late-stage, or cancer-free), and the statistics $E, L,$ and X are updated accordingly. Patients exit the simulation upon death or from withdrawal of their participation under the surveillance program. Patients also exit the DM's panel upon revealing early- or late-stage cancer and are replaced by

sampling a new patient from the dataset with a probability distribution proportional to their follow-up times in the program.

At terminal time T, the performance metrics are calculated using $E, L,$ and X as follows:

$$Proportion\ of\ early\ stage\ cancers\ detected = \frac{E}{E + L}$$

$$Proportion\ of\ resources\ spent\ on\ cancer\ patients = \frac{X}{K * T}$$

An additional metric is calculated representing the overall detection rate of cancer in the DM's subject panels across all decision epochs. This is introduced to serve as an additional metric to compare given a key limitation of the synthetically generated data and the applied assumption of early- and late-stage cancer distributions (to be discussed in Section 3.6, Simulation setup). The detection rate is calculated as follows, where C represents the subset of panel subjects at any given time that do have cancer:

$$Proportion\ of\ cancers\ detected = \frac{E + L}{|C|}$$

3.5 Data Generation

Lee et al. (2014) use historical data from Hepatitis-C trial study, but due to lack of data availability, this replication paper will aim to generate synthetic data using a graphical modeling sampling approach. However, based on the original simulation, a few key features exist in the original dataset that are not possible to accurately estimate via population parameter sampling. Therefore, simplifications in the simulation will be made, and limitations are noted in Section 3.6.

Population parameters from the original trial dataset are provided by Lee et al. (2014) including average age, blood platelet, smoker statuses, and AFP readings (with mean and standard deviation metrics) grouped by patients with and without HCC. A Simple Sampling algorithm is used to sample from these probability distributions to generate subject pools for HCC and non-HCC patients to be used for simulation.

The data generation algorithm proceeds as follows:

- Generate randomized list of HCC and non-HCC patients based on probability distribution of HCC in the at-risk population of study (P(HCC) = 82/967, P(non-HCC) = 1 – P(HCC)). This is represented as a binary vector of size N for the total number of subjects to include in the pool for study.
- For each subject i in the pool, sample each numerical static risk factor from a normal distribution based on the population parameters per class (HCC vs. non-HCC). Sample categorical risk factors from a binomial distribution with probability equal to the incidence rate for that risk factor (e.g., P(Having ever smoked | HCC)).

Although synthetic data has the limitation of reduced generalizability to unseen authentic data, performance can still be compared on the same dataset between the author's 3 RL

methods and the proposed MDP model to assess the relative effectiveness of MDPs in this resource allocation setting.

3.6 Simulation

3.6.1 Simulation Sequence

The following outlines the process logic for a single iteration of the simulation:

1. Initialize
    - Metrics E, L, and X to 0
    - Time period t to 0
    - Agent decision maker populated with policy based on the varying RL methods
    - Training and testing data split from the total pool of synthetically generated subjects:
        - Training data is used to train a logistic regression model to obtain coefficients used in the interval estimation and Boltzmann exploration RL methods for calculating risk factor probabilities
        - Testing data is used to run proceeding simulation.
2. DM initializes panel of size n from the testing subject pool and identifies subsets C and NC of subjects with HCC and without HCC, respectively.
3. Policy Module: each patient's state (risk factors) are input into the current RL policy function and a subset of k patients is selected for screening:
    - For non-selected patients in panel, DM's knowledge of them remains unchanged.
4. Imaging Module: For selected patients:
    - With HCC: increment metric X by +1, then determine cancer stage by random sampling of estimated tumor size s:
        - If $1 \leq s \leq 5$, then early-stage cancer; increment metric E by +1
        - If $s > 5$, then late-stage cancer; increment metric L by +1
        - Otherwise, cancer-free
    - Without HCC: output cancer-free state, then update AFP readings (assign new AFP reading standard deviation and rate of rise based on population parameters).
5. New Patient Module: Screened patients deemed to be early-stage or late-stage cancer are replaced in the panel with new patients from the testing pool.
6. Patient Exit Module: d proportion of panel subjects are chosen randomly to leave study, replicating study participants with withdrawal due to voluntary leave or death in the original trial study.
7. Repeat the above until end of planning horizon T.
8. Report metrics:
    - Proportion of cancers detected
    - Proportion of cancers detected in early stage
    - Proportion of screening resources spent on patients who eventually develop cancer

This sequence is repeated for a set number of iterations, and the metrics from all iterations are averaged to obtain the mean metrics per simulation.

3.6.2 Limitations & Deviations from Original Study

Given the lack of available data from the original HCC trial, key features are not reproducible. The relevant deviations from the original study are discussed in this section, along with the limitations of several key assumptions made in simulation.

First, since the data is generated by sampling population parameters for each risk factor independently (i.e., only with the dependence on the class HCC or no HCC), the data will inherently not account for possible correlations between risk factors that could be relevant for learning and prediction.

In addition, in simulation step 4 outlined above (Imaging Module), the original study had access to time series data that outlined detection dates for which subjects came in for their screenings and cancerous tumors were detected. This information was then used to generate an estimate of the tumor size at the decision epoch time t by backtracking the size of the tumor from its detection date in the dataset. Therefore, an accurate determination of early-stage or late-stage cancer can be developed. In the absence of this data, the alternative employed is to randomly sample tumor sizes for each subject upon screening time based on tumor size parameters. These parameters are obtained from a study comparing the tumor sizes of HCC across varying imaging methods (Chen et al., 2020). However, this sampling method does not account for possible correlations between tumor size and other risk factors over time that could be learned by an agent.

Similarly, individual AFP readings over time were available in the original trial dataset, however, this replication study samples AFP standard deviation and rate of rise directly from the parameters provided by the authors. This sampling method is likely to miss relevant relationships and trends in patient risk factors over time.

During the New Patient Module (simulation step 5), the original study employs probability selection of each remaining subject in the pool as a function of their follow-up times in the original dataset. However, since no distributions are provided for follow-up times, random replacement is used for determining new patients in this replication study.

Lastly, the Patient Exit Module (simulation step 6) was implemented by true departure rates from the subjects in the original study. However, for the purposes of this replication study, a random proportion of panel members d is selected to exit and leave the simulation.

Section 4. Experimental Setup

Based on the simulation logic outlined in section 3.6.1, several hyperparameters are selected on both the simulation and policy level.

On the simulation level, a time horizon of 10 years is selected with decision epochs spaced 90 days apart (similar to the original study), generating approximately 40 decision epochs per simulation run. With the testing subset of the synthetic dataset containing 5,000 unique subjects, each RL policy initializes a panel of 500 subjects and is tested against 4 different levels of resource constraints, denoted as a ratio of the chosen panel size (k = 0.1, 0.2, 0.3, and 0.4). For example, if the chosen panel size n is 500, and the ratio k is 0.2, then the

available screening resources to allocate per decision epoch is 100. Subjects are sampled from the test pool with replacement, and the patient exit rate is set at 5%.

Per model, the parameters tested are as follows:

- ϵ-greedy: e = {0.05, 0.1, 0.25}
- Interval estimation: z = {1, 3, 5}
- Boltzmann exploration: tau = {0.250, 0.325, 0.400}

Section 5. Results

The results of the replication simulation are shown in the below figures, outlining the cancer detection and early-stage cancer detection rates by policy and constraint level ('k proportion').

Figure 6 shows a summary of the detection rate mean and standard deviation results per simulation, along with the optimal hyperparameters determined per model.

Figures 2 and 3 depict rising mean detection rates for all models as the constraint level increases, consistent with the findings in the original study. However, the early-stage cancer detection rate results are significantly higher in the smaller constraint levels compared to those shown by Lee et al. (2014). This discrepancy is likely due to the limitations in the synthetic generation of the data and sampling of population parameters, as well as the key limitation of missing tumor size information for accurately determining early-stage and late-stage cancers in screened individuals (see Section 3.6.2).

The limitations of the synthetic data are apparent in the results by model since the original study showed the Boltzmann method to have outperformed all methods, and the myopic RL method to be the worst performer. However, in Figure 2, the Boltzmann method exhibits similar, though slightly lower, performance against the myopic method, greatly contradicting the original study.

The proposed MDP DQN method also appears to be underperforming against all models across constraint levels. However, from examining the standard deviation of cancer detection rates in Figure 4, the DQN method exhibits significantly lower deviation of performance results across 50 iterations. This suggests that the performance of the DQN model still lies within the bounds of the other RL methods given the noise introduced in various stages of the simulation and the sampling of population parameters.

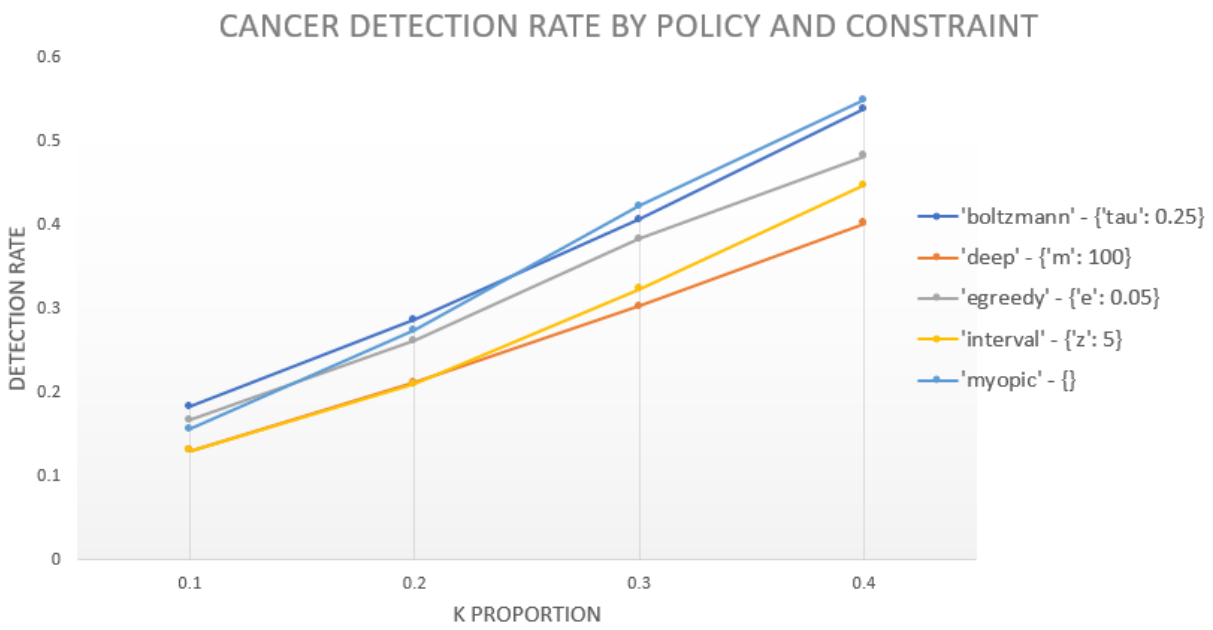

Figure 2: Cancer detection rates by policy and resource constraint level.

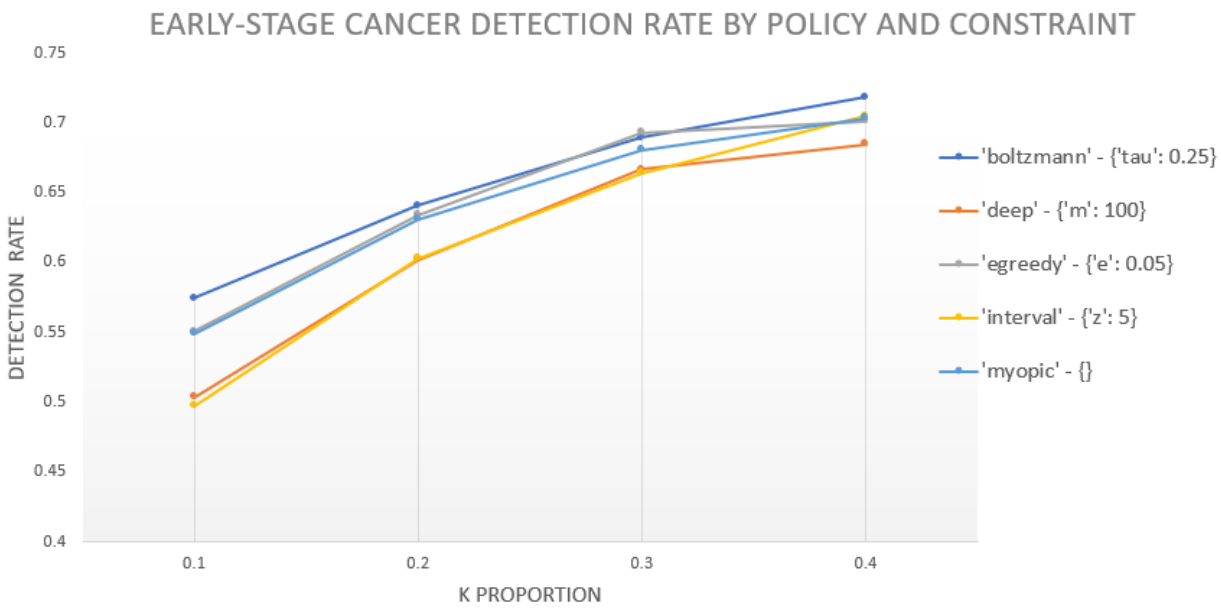

Figure 3: Early-stage cancer detection rates by policy and resource constraint level.

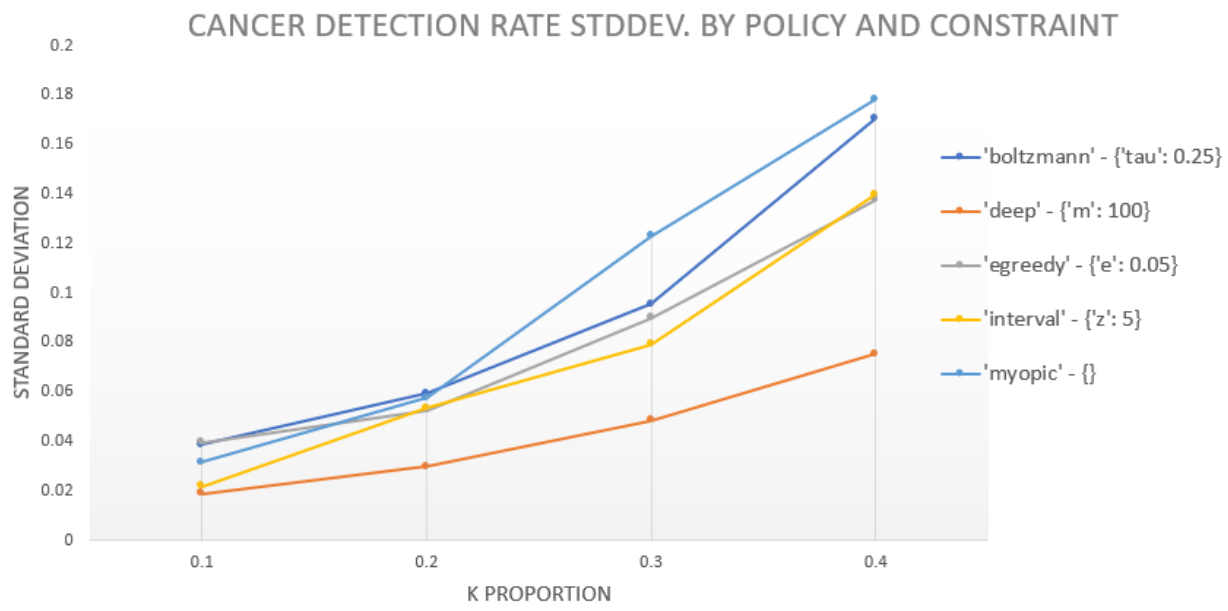

Figure 4: Cancer detection rate standard deviation by policy and resource constraint level.

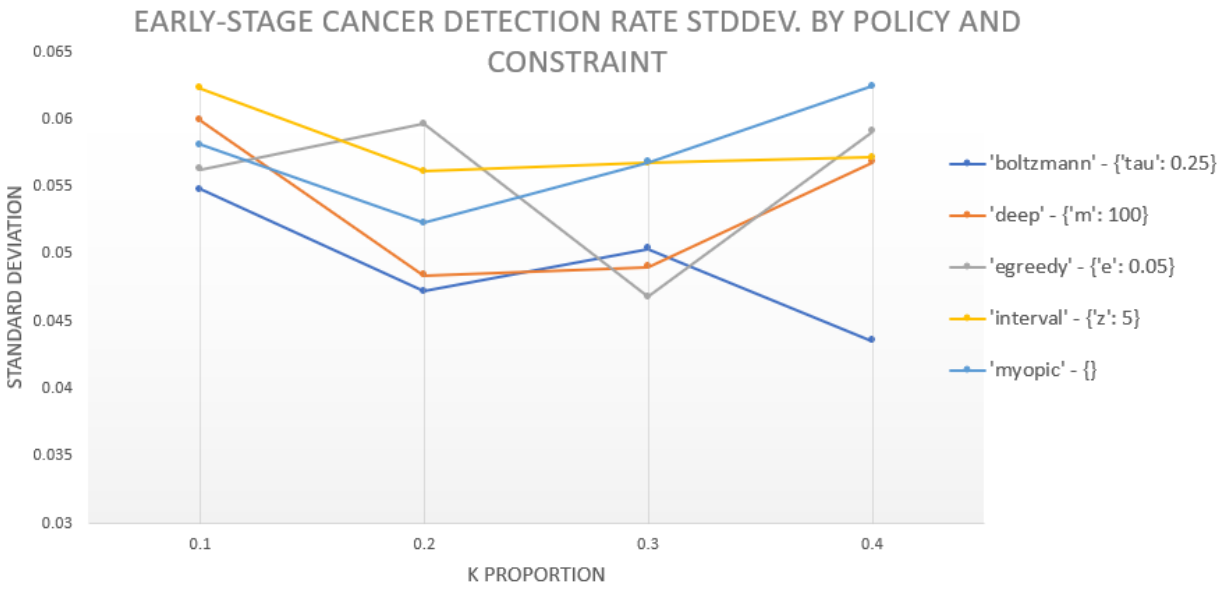

Figure 5: Early-stage detection rate standard deviation by policy and resource constraint level.

Detection Rate Means and Standard Deviations Comparison by Model and Constrain Level

k	Policy	Parameter	Detection Rate	Early-stage Det. Rate	Detection Rate STDDEV.	Early-stage Det. STDDEV.
	Boltzmann	Tau = 0.250	18%	57%	4%	5%
	DQN	m = 100	13%	50%	2%	6%
0.1	e-greedy	e = 0.05	17%	55%	4%	6%
	Interval	z = 5	13%	50%	2%	6%
	Myopic		16%	55%	3%	6%
	Boltzmann	Tau = 0.250	29%	64%	6%	5%
	DQN	m = 100	21%	60%	3%	5%
0.2	e-greedy	e = 0.05	26%	63%	5%	6%
	Interval	z = 5	21%	60%	5%	6%
	Myopic		28%	63%	6%	5%
	Boltzmann	Tau = 0.250	41%	69%	10%	5%
	DQN	m = 100	30%	67%	5%	5%
0.3	e-greedy	e = 0.05	38%	69%	9%	5%
	Interval	z = 5	32%	66%	8%	6%
	Myopic		42%	68%	12%	6%
	Boltzmann	Tau = 0.250	54%	72%	17%	4%
	DQN	m = 100	40%	68%	8%	6%
0.4	e-greedy	e = 0.05	48%	70%	14%	6%
	Interval	z = 5	45%	70%	14%	6%
	Myopic		55%	70%	18%	6%

Figure 6: Summary of detection rate means and standard deviations across models and constraint levels.

Section 6. Conclusion

This implementation paper aimed to replicate the simulation model and RL methods proposed by Lee et al. (2014) on a synthetic dataset generated by sampling population parameters. In addition, an extension to the original study was explored by formulating an MDP model and implementation through a modified DQN algorithm using experience replay.

The results of this replication study revealed the key limitations in the usage of the synthetic dataset as the lack of key features including tumor size and detection date, along with the independent sampling of static risk factors, were shown to have affected model performance through the imperfect representation of relevant relationships between features.

However, although the proposed DQN method did not yield an improvement in detection performance in this study, the smaller standard deviation of results (allowing the model to float within the performance bounds of the alternative methods) shows potential for further hyperparameter tuning and reward function adjustments that could enable the model to improve detection performance.

Key limitations of this replication study were discussed in detail within section 3.6.2, the most significant of which being the inherent lack of representation of the correlations between risk factors over time due to the independent population sampling method used.

Future work can be done to explore hyperparameter tuning of the proposed DQN model (e.g., number of hidden layers, units), along with varying definitions of the reward and cost functions fed to the DQN learning agent. The current reward function tracking changes in the

true positive and true negative rates inherently distributes greater rewards for subjects examined earlier in the queue (given the true positive and true negative rates will exhibit lower oscillation as both the numerator and denominator values increase), so regularization methods might be beneficial to redistribute rewards more evenly through time. In addition, penalty functions can be explored to properly penalize the learning agent for false positives and false negatives representing wasted resources and missed cancer detections, respectively (with a possible emphasis to weight false negatives more greatly relative to false positives).

References

[1] Lee E, Lavieri MS, Volk ML, and Xu Y. Applying reinforcement learning techniques to detect hepatocellular carcinoma under limited screening capacity. Health Care Manag Sci. 2015 Sep, 18(3):363-75. doi: 10.1007/s10729-014-9304-0. Epub 2014 Oct 12. PMID: 25308168.

[2] Elliot Lee, Mariel S. Lavieri, Michael Volk (2018) Optimal Screening for Hepatocellular Carcinoma: A Restless Bandit Model. Manufacturing & Service Operations Management, 21(1):198-212

[3] Frazier AL, Colditz GA, Fuchs CS, Kuntz KM (2000) Cost-effectiveness of screening for colorectal cancer in the general population. JAMA: J Am Med Assoc 284(15):1954–1961

[4] Urban, N., Drescher, C., Etzioni, R., & Colby, C. (1997). Use of a stochastic simulation model to identify an efficient protocol for ovarian cancer screening. Controlled clinical trials, 18(3), 251–270. https://doi.org/10.1016/s0197-2456(96)00233-4

[5] Chen, F., Wang, F., Sun, S. et al. Size measurements of hepatocellular carcinoma: comparisons between contrast and two-dimensional ultrasound. BMC Gastroenterol 20, 390 (2020). https://doi.org/10.1186/s12876-020-01535-1