RL for Sepsis Control

Abstract

This project uses reinforcement learning (RL) to simulate how doctors make treatment decisions for sepsis patients over time. We built a custom simulator that models patient health and treatment effects as a Markov Decision Process (MDP). Three RL algorithms, Q-learning, Deep Q-Network (DQN), and Policy Gradient were trained and compared for learning performance and safety. Both Q-Learning and DQN learned effective and clinically reasonable strategies, with DQN achieving higher rewards and smoother learning. These results show that RL can model realistic decision-making in critical care simulations.

Introduction

Sepsis is a life-threatening condition characterized by organ dysfunction caused by an uncontrolled response to infection. It remains a leading cause of mortality in intensive care units because doctors must make rapid, sequential decisions under uncertainty regarding fluid resuscitation, antibiotic timing, and vasopressor dosing. These decisions exhibit delayed effects, complex physiological interactions, and a narrow margin for error. These characteristics make sepsis management a natural candidate for reinforcement learning.

A major driver of RL research in critical care has been the availability of large-scale observational datasets such as MIMIC-III, a publicly accessible, de-identified critical care database. MIMIC-III contains detailed ICU records, including vital signs, laboratory values, medication usage, interventions, and outcomes, from over 40,000 patients, enabling researchers to train machine learning models on real physiological trajectories. Because of its richness and open-access nature, MIMIC-III has become foundational in data-driven sepsis research and has supported studies ranging from mortality prediction to optimal treatment analysis.

Early work by Raghu et al. (2017) applied deep RL to observational ICU data and demonstrated that learned treatment policies sometimes aligned with expert clinical judgment and were associated with lower estimated mortality. Similarly, Gottesman et al. (2019) outlined principles for evaluating RL in healthcare, emphasizing the importance of safety, interpretability, and robustness. While these studies highlight the importance of RL for clinical decision support, they also reveal fundamental limitations: Observational ICU datasets reflect clinician treatment choices, do not cover the full range of possible interventions, and include systematic biases in how care is delivered and documented. In addition, privacy, computational cost, and reproducibility constraints make it challenging for students or researchers to systematically compare RL algorithms or explore safety-critical modifications. These challenges motivated us to create a controlled simulation that captures key clinical dynamics without relying on restricted ICU data.

In this project, we address these gaps by constructing a fully controllable sepsis treatment simulator modeled as a Markov Decision Process. The environment includes continuous patient states, blood pressure, heart rate, lactate levels, and infection severity, as well as discrete treatment actions such as fluid boluses, antibiotics, vasopressors, or no intervention. The simulator incorporates clinically motivated physiological relationships, such as the stabilizing but potentially harmful effects of aggressive fluids or vasopressors, enabling RL agents to learn policies that balance efficacy and safety.

We evaluate three RL algorithms with increasing representational capacity: Q-learning with linear approximation, Deep Q-Networks, and Policy Gradient. Beyond standard approaches, we implement a safety-constrained learning mechanism that penalizes action learning that leads to physiologically unstable states, effectively integrating safety considerations directly into the reward structure. This mechanism mimics real-world clinical constraints, where unsafe treatments are unacceptable regardless of exploratory value.

Our main goals were to build a simple, reproducible simulator for sepsis treatment and to test how different RL algorithms perform under safety constraints. First, we develop an interpretable and fully reproducible simulation framework for studying reinforcement learning in a healthcare setting. This simulator overcomes several limitations inherent to observational datasets like MIMIC-III, including confounding treatment biases, incomplete action coverage, and restricted opportunities for controlled experimentation. By modeling sepsis treatment as a continuous-state, discrete-action MDP with clinically motivated physiological dynamics, the environment enables comparison of RL models, exploration of safety mechanisms, and clear attribution of agent behavior to underlying model design choices. Second, we experimentally compare classical and modern RL methods, Q-learning with linear approximation, Deep Q-Networks, and Policy Gradient algorithms, alongside a safety-constrained variant that penalizes actions leading to physiologically unstable states. This addition incorporates safety considerations directly into the learning process, yielding more cautious treatment strategies and offering insight into the tradeoffs between survival, medication usage, and model complexity. Together, these contributions provide a controlled platform for understanding RL behavior in safety-critical domains and highlight promising directions for future work on safe decision-making in clinical environments.

Methodology

The sepsis treatment problem is modeled as a Markov Decision Process (MDP) to simulate how a medical agent learns to stabilize a patient's vital signs through sequential decision-making. Each simulation step represents a short time interval during which treatment is applied, and the patient's physiological state evolves. In this setting, the agent observes the patient's condition, selects a treatment, and receives feedback based on the patient's response.

- **State (s_t):** Each state represents the patient's current physiological condition, including continuous variables such as mean arterial pressure (MAP), heart rate (HR), lactate concentration, and infection severity. These variables are normalized between 0 and 1 to ensure stable learning.

$$\boldsymbol{s_t}=[\boldsymbol{MAP_t},\boldsymbol{HR_t},\boldsymbol{Lactate_t},\boldsymbol{Infection_t}]$$

- **Actions (a_t):** The agent chooses one of six discrete available treatment options at each step:
    1. No action

2. Small fluid bolus
3. Large fluid bolus
4. Antibiotics on/off
5. Low-dose vasopressor
6. High-dose vasopressor

Each treatment option influences the next state of the patient.

- **Transition Dynamics:** Patient states evolve according to deterministic equations with added random noise to represent physiological uncertainty. For example, a fluid bolus raises blood pressure but can also increase lactate slightly if overused.
- **Reward Function:** The reward function balances treatment success against medication burden and risk:

$$\boldsymbol{r_t}=+1\text{ (stable vitals)}-0.2\text{ (each treatment)}-2\text{ (unsafe state)}$$

A large positive reward (+10) is given for survival, and a large penalty (−10) for death. This structure encourages the agent to achieve stability with minimal intervention. The objective is to learn a policy π(a|s) that maximizes the expected cumulative reward.

$$J(\pi)=E\pi[\Sigma\gamma^t r_t\ ]\text{ where }\gamma=0.99\text{ discounts future outcomes}$$

RL Algorithm

Three reinforcement learning algorithms were implemented and compared to evaluate performance and learning behavior.

1. **Q-learning (Baseline):**

Q-Learning serves as the baseline algorithm. The Q-function Q(s,a) is updated after each interaction using the Bellman equation:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_t+\gamma\max Q(s_t+1,a)\text{-}Q(s_t,a_t)]$$

where α is the learning rate and γ is the discount factor. This method uses a linear function approximation over state features to allow limited generalization. Its simplicity enables transparent evaluation of reward signals and policy convergence.

2. **Deep Q-Network (DQN):**

The Deep Q-Network (DQN) replaces the Q-table used in basic Q-Learning with a neural network that learns to estimate how good each action is for a given patient state. The network has two hidden layers with 64 and 64 neurons, using ReLU activation functions, and outputs one value per action. To make learning stable, DQN uses two main techniques:

- Experience replay: stores past experiences and randomly samples them during training so the model does not overfit to recent events.
- Target network: a second network that updates more slowly to prevent large swings in learning.

DQN was chosen as the main algorithm because it can handle continuous state variables like blood pressure and lactate levels while still providing clear, discrete treatment decisions. It also captures nonlinear patterns in the data, which are common in patient physiology.

3. **Policy Gradient (REINFORCE):**

The Policy Gradient (REINFORCE) algorithm learns the treatment strategy directly instead of estimating Q-values. It adjusts the probabilities of taking each action so that actions leading to better outcomes become more likely over time.

This approach can produce smoother and more realistic treatment behavior, since it learns a full probability distribution over actions rather than a single best action. However, Policy Gradient methods usually require more episodes to reach stable performance compared to DQN.

A safety constraint is added to the reward function to discourage actions that lead to physiologically unsafe states. A penalty of -2 is applied whenever vital signs fall below safe clinical thresholds, such as mean arterial pressure < 60 mmHg or heart rate > 120 bpm. Learning favors stable, cautious treatment sequences rather than aggressive interventions that yield short-term gains but unsafe states, mimicking real clinical standards, where risky interventions are avoided regardless of potential reward.

The simulator and reinforcement learning models are developed in Python using PyTorch and an OpenAI Gym style interface. Each training episode consists of up to 50 time steps and ends early if the patient stabilizes or dies. The learning process uses an epsilon-greedy exploration strategy, with exploration gradually decreasing as the agent gains experience.

Model performance is evaluated using the following metrics:

- Efficacy: Percentage of episodes ending with patient survival and stabilized vital signs.
- Safety: Average amount of medication administered per episode, where lower values indicate less overtreatment.
- Sample efficiency: Number of environment interactions required for convergence.

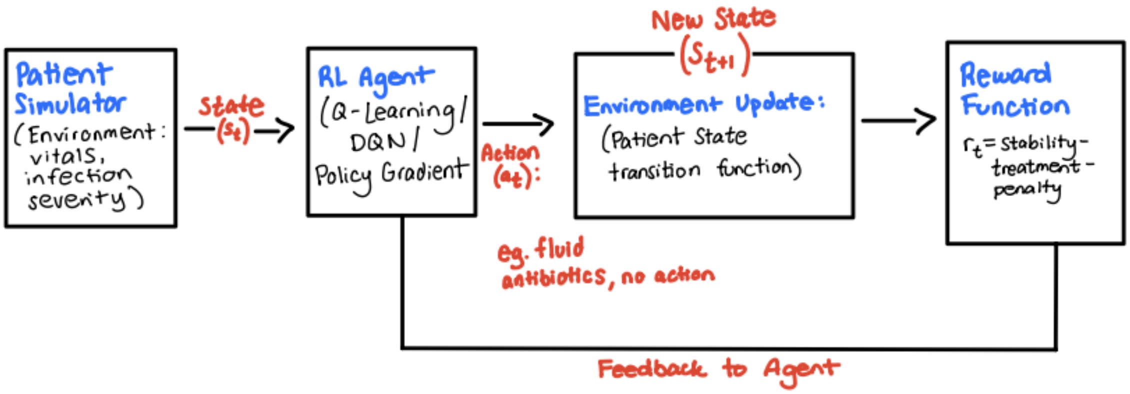

Patient Simulator
(Environment: vitals, infection severity)
State
(s_t)
RL Agent
(Q-Learning/ DQN/ Policy Gradient
Action
(a_t):
eg. fluid antibiotics, no action
New State
(S_{t+1})
Environment Update:
(Patient State transition function)
Reward Function
r_t= Stability - treatment - penalty
Feedback to Agent

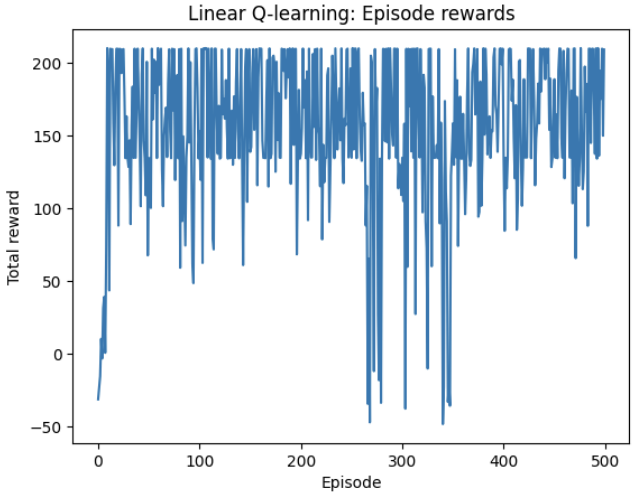
Figure 1: Overview of the Reinforcement Learning System for Sepsis Treatment

Results

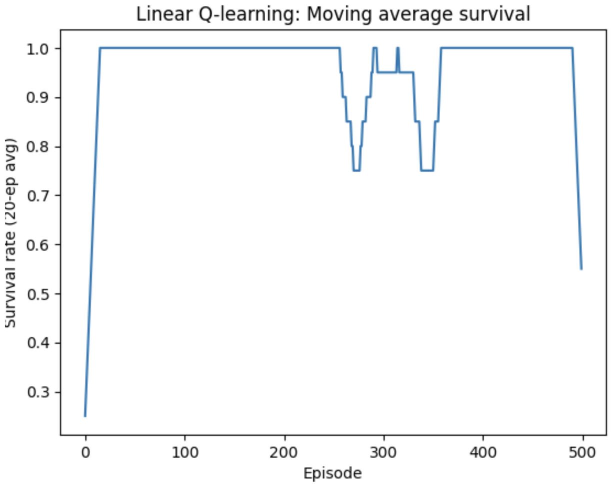
All experiments were conducted using the custom sepsis simulator, with variables such as blood pressure, heart rate, lactate, and infection, and six discrete treatment actions. Each episode ran for 50 time steps and terminated early if the patient stabilized or died. The reward function incentivized stable vitals while penalizing medication burden and unsafe physiological states. Survival received a terminal reward of +10 and death -10.

We trained three reinforcement learning agents: Linear Q-Learning, Deep Q-Network, and a Policy Gradient agent, each for 500 episodes. The agents used $\gamma = 0.99$ and epsilon-greedy exploration policy that decayed from 1.0 to 0.05. The linear Q-learning agent used a feature vector, while the DQN used a two-layer neural network with ReLU activations, experience replay, and a target network. Performance was evaluated using efficacy (survival and stabilization rate), safety (average medication use per episode), and learning stability (reward convergence and training stability). The Policy Gradient agent optimized a stochastic policy directly using episode returns.

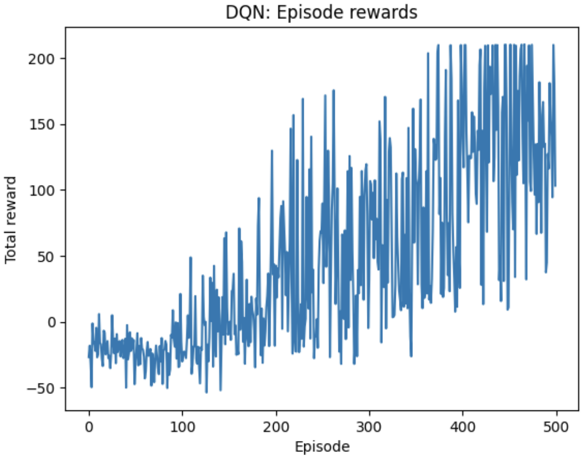
Linear Q-learning showed rapid and stable improvement. In the first 100 episodes, total reward was highly volatile and frequently negative, reflecting random, unsafe treatment decisions and early patient death. As learning progressed, reward began trending upward, and by episodes 400-500 the agent consistently achieved positive returns.



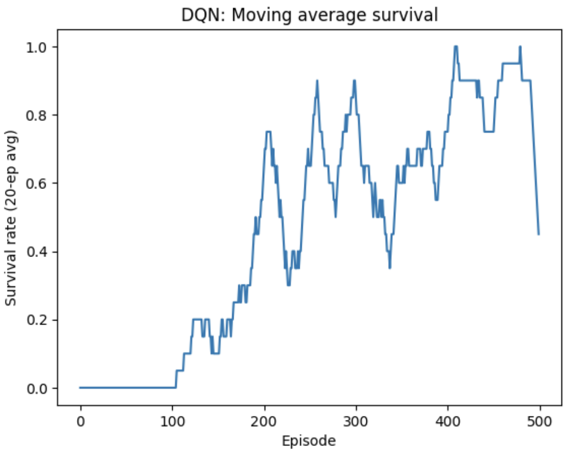
Survival followed a similar trajectory. The agent initially had near-zero survival but quickly improved as it learned to prioritize antibiotics and moderate fluid administration. After 150 episodes, the moving average survival rate climbed above 0.8, and during the final stage of training, the agent survived in nearly all episodes. These findings indicate that a simple linear approximation is expressive enough to learn effective sepsis treatment behavior in this simulator.



The DQN agent learned more slowly early in training. During the first 100–150 episodes, DQN produced unstable reward values and near-zero survival because the replay buffer was still populating and the network weights were largely uninitialized. After this warm-up period, DQN rewards began steadily increasing, eventually reaching higher peak values than the linear agent.



Survival also improved steadily. Around episode 250 the moving average survival approached 0.7–0.8, and by the final 100 episodes DQN achieved near-perfect survival. This reflects DQN’s ability to capture nonlinear relationships, such as infection-MAP-lactate interactions, that the linear agent cannot fully model.



The Policy Gradient agent exhibited substantially different behavior compared to the value-based methods. While total reward showed modest improvement over training, learning remained highly noisy and plateaued at a lower level than Linear Q-Learning and DQN. In contrast to value-based agents, the Policy Gradient model did not achieve patient stabilization under the current simulator configuration, resulting in a zero survival rate during evaluation. This outcome reflects both the high variance of Policy Gradient methods and the limitations of the simulator's termination dynamics, which make survival difficult to achieve without stronger reward shaping.

This project demonstrates that reinforcement learning can effectively model sequential treatment decisions in a simulated sepsis environment, while highlighting important differences across algorithmic approaches. Linear Q-Learning and Deep Q-Networks both learned clinically plausible treatment strategies, achieving stable improvements in reward and high survival performance, with DQN capturing more complex nonlinear dynamics at the cost of slower convergence. In contrast, the Policy Gradient method showed limited effectiveness, improving reward modestly but failing to achieve patient stabilization under the current simulator configuration. This result shows how difficult it can be to apply basic policy-gradient methods in complex, safety-critical environments like healthcare. Overall, our simulator provides a solid base for future work, including making the environment more realistic, improving the reward design, and testing actor–critic methods.

Discussion and Conclusion

The results of this project demonstrate that reinforcement learning can learn coherent and clinically plausible sepsis treatment strategies, even within a simplified simulation environment. Linear Q-Learning adapted quickly and achieved strong survival outcomes, indicating that key aspects of treatment logic can be captured using relatively simple value-based methods. DQN required a longer training period but ultimately achieved more stable learning and higher overall returns, reflecting its ability to model nonlinear physiological interactions such as the relationship between fluids, infection, and blood pressure. In contrast, the Policy Gradient approach showed limited effectiveness under the current simulator configuration, exhibiting noisy learning dynamics and modest reward improvement without achieving patient stabilization.

Working on this project helped us understand how sensitive RL models are to reward signals and how small design choices can lead to very different outcomes. The value-based agents benefited from the simulator's structured reward function, which balanced treatment effectiveness with medication burden, while the Policy Gradient agent struggled. This highlights how reward design in clinical reinforcement learning can lead to unstable or unsafe behaviors, particularly for policy-based methods.

Future work should extend the simulator to include partial observability, delayed antibiotic effects, and more granular action spaces. Incorporating the safety penalty into all experiments, comparing constrained versus unconstrained learning, and validating policies against real-world ICU trajectories would further bridge the gap between simulation and clinical applicability. Together, these improvements would deepen understanding of how RL can support decision-making in high-risk medical settings.

Roles

Paulina led the design of the sepsis simulation environment, establishing the MDP framework, defining state representations and reward functions, and implementing the baseline Linear Q-Learning model. She also analyzed training behavior and learning stability to validate the environment and baseline performance. Uma focused on developing and training more advanced reinforcement learning approaches, including Deep Q-Network (DQN) and Policy Gradient models. She conducted comparative evaluations across methods and analyzed learning efficiency and safety-related performance to assess trade-offs between model complexity and clinical reliability.

Citations

Gottesman O, Johansson F, Komorowski M, Faisal A, Sontag D, Doshi-Velez F, Celi LA. Guidelines for reinforcement learning in healthcare. Nat Med. 2019 Jan;25(1):16-18. doi: 10.1038/s41591-018-0310-5. PMID: 30617332.

Johnson AEW, Pollard TJ, Shen L, Lehman LWH, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, Mark RG. MIMIC-III, a freely accessible critical care database. Sci Data. 2016 May 24;3:160035. doi: 10.1038/sdata.2016.35. PMID: 27219127.

Raghu A, Komorowski M, Celi LA, Szolovits P, Ghassemi M. Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach. *arXiv [cs.LG]*. 2017. Available from: https://arxiv.org/abs/1705.08422.