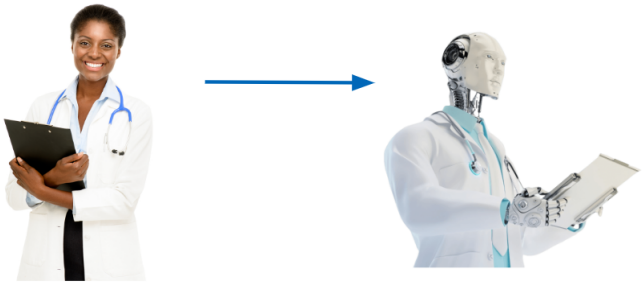
Project Report

Management of septic shock in intensive care using Reinforcement Learning



Student: Oded Mousai

Supervision: Prof. Michael Beil, Prof. Sigal Sviri, Prof. Leo Joskowicz

1. Introduction

Sepsis is a life-threatening medical condition in which the body reacts severely to an infection. Without timely treatment in the intensive care unit (ICU), sepsis can rapidly lead to tissue damage, organ failure, and ultimately, death. The management of septic shock typically involves administering intravenous (IV) fluids and vasopressors. Varied dosing approaches for these treatments can significantly impact patient outcomes [1]. However, there is no universally agreed-upon policy for sepsis management [2] because individual patients respond very differently to these medical interventions, and there is potentially life-threatening harm if the treatment is administered excessively. As no single policy is appropriate for all patients, a greater degree of personalization of sepsis treatment is necessary.

Reinforcement Learning (RL) is a machine learning paradigm that is used to learn optimal sequential decisions in dynamic environments. In the case of septic management, RL can be used to discover treatment strategies for septic patients in ICUs that eventually improve their chances of survival. Unlike fixed protocols, RL continuously incorporates real-time patient information at each decision-making step, which allows it to personalize its treatment recommendations in response to the specific patient's changing condition. In this project, we

reimplemented the methodology from the paper “The Artificial Intelligence Clinician Learns Optimal Treatment Strategies for Sepsis in Intensive Care” by Komorowski et al. [3], which applies an offline RL algorithm to patient data. This replication will serve as a framework for testing new ideas and methods in the future.[1]

2. Materials and Methods

We implemented the same methodology that outlined in [3]. Any changes are explicitly noted.

2.1 Computational Model

We modeled the problem of sepsis treatment with vasopressors and IV fluids as a Markov decision process (MDP). This MDP is characterized by the following components:

- **State Space (S)**: A finite set of states. A state represents a snapshot of a patient's characteristics and clinical measurements at a specific point in time.
- **Action Space (A)**: A finite set of actions. An action corresponds to a combination of IV fluids and vasopressor dosages.
- **Transition Matrix (T)**: Containing the probability that taking action in state s at time t will lead to state s′ at time t+1. This matrix elucidates how the system evolves over time.
- **Reward Function (R)**: Defines the immediate feedback upon transitioning to state s. Positive rewards are granted for transitions leading to favorable states while entering undesirable states results in penalties.

The goal of an RL algorithm is to learn an optimal **policy** - the recommended action for each given state. The RL algorithm processes the patient's current state, selects treatment action from the available options, and assesses the quality of its decisions by receiving feedback in the form of rewards or penalties when transitioning from one state to another (Fig. 1). Over time, it refines its policy through learning from these interactions, ultimately leading to a treatment strategy that maximizes patient outcomes in sepsis management.

[1] Our code is available here: https://github.com/OdedMous/Sepsis-RL/tree/main

Figure 1: Reinforcement Learning (RL) framework. There is an environment (a sick patient) and an agent (a clinician) who acts to maximize some reward (patient health and survival).
Image source: https://www.jmir.org/2020/7/e18477/

Traditional RL algorithms learn optimal policies through online interaction with the environment. Yet, in critical healthcare scenarios like sepsis management, applying suboptimal or unoptimized policies in real-time poses a significant risk, potentially resulting in adverse outcomes. To address this safety concern, we used the **Offline RL** approach, which leverages pre-existing historical patient data and treatment records to train the RL model, without directly experimenting on real-time patients.

We applied the Policy Iteration method, an iterative model-based RL algorithm. This algorithm works as follows: It begins by initializing a policy $\pi : S \rightarrow A$, that dictates which action is taken while in a particular state. The algorithm then undergoes two main steps: policy evaluation and policy improvement. During policy evaluation, the algorithm estimates the value function V for the current policy. This means computing the expected cumulative rewards for each state s based on the current policy. It does so by solving the Bellman equation iteratively:

$$V(s) \leftarrow \sum_{s'} T(s,\ \pi(s),\ s')\ [R(s,\ \pi(s),\ s')\ +\ \gamma V(s')]$$

Following policy evaluation, policy improvement takes place. The algorithm updates the policy by selecting actions that maximize the expected cumulative rewards for each state using the value function obtained in the previous step:

$$\pi(s) \leftarrow argmax_a \sum_{s'} T(s,\ a,\ s')\ [R(s,\ a,\ s')\ +\ \gamma V(s')]$$

This process continues until the policy stabilizes, meaning $\pi_t(s)\ = \pi_{t-1}(s)$ for every state s at some time-step t.

2.2 Data

We used data from The eICU Collaborative Research Database, which documents patient information and given treatments in a higher time resolution compared to the MIMIC-III database that was used in [3]. The data include over 200,000 patient trajectories across 335 ICUs at 208 hospitals during the period of 2014–2015. From this database, we selected only patients diagnosed with sepsis and excluded individuals under the age of 18 at the time of admission. For each patient, we extracted a set of 45 variables, encompassing demographic characteristics, vital signs, laboratory results, and the usage of IV fluids and vasopressor medications (Table 1). We then structured the patient data as a multidimensional time series, with data points recorded at 1-hour intervals (in contrast to Komorowski's paper that used 4-hour intervals). In instances where certain data variables had multiple measurements within the same 1-hour interval, we applied appropriate aggregation methods such as averaging or summation to ensure the data's consistency. This process resulted in 3188 patient trajectories. Missing values were imputed using a nearest neighbors approach (KNN algorithm [4]).

Category	Variable	Type	State Space
Demographics	Age	Static	+
	Gender	Static	+
	Weight	Static	+
	Readmission to intensive care	Static	+
Vital signs	Modified SOFA	Dynamic	+
	SIRS	Dynamic	*
	Glasgow Coma Scale (GCS)	Dynamic	+
	Heart rate systolic, diastolic blood pressure, mean blood pressure, shock index	Dynamic	+
	Respiratory rate, SpO2	Dynamic	+
	Temperature	Dynamic	+
Lab values	Potassium, Sodium, chloride	Dynamic	+
	Glucose, BUN, creatinine	Dynamic	+
	SGOT, SGPT, total bilirubin, albumin	Dynamic	+
	Hemoglobin	Dynamic	+
	White blood cells count, platelets count, PTT, PT, INR	Dynamic	+
	pH, PaO2, PaCO2, base excess, bicarbonate, lactate	Dynamic	+
Ventilation parameters	Mechanical ventilation	Dynamic	*
	FiO2	Dynamic	+
Medications and fluid balance	Current IV fluid intake over 1h	Dynamic	-
	Maximum dose of vasopressor over 1h	Dynamic	-
	Urine output over 1h	Dynamic	+
	Cumulated fluid balance since admission (includes preadmission data when available)		*
Outcome	ICU discharge status	Static	-
	Hospital mortality	Static	-

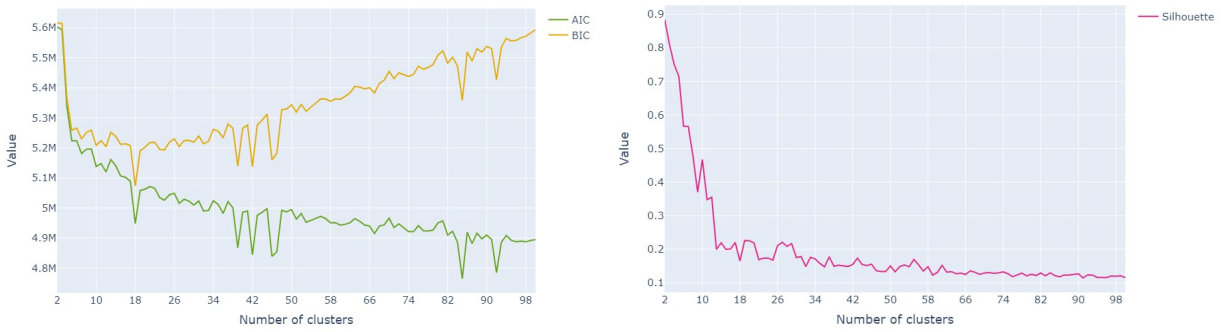
Table 1: List of variables used in this project. The 'State Space' column specifies which variables are used for defining the patient's states. (*) Will be added to the state space in future implementation

2.3 Environment Generation

The environment of sepsis management consists of a state space, action space, transition matrix, and reward function. We discretized the state and action spaces as follows: The continuous multi-dimensional states were clustered into 18 unique states using a K-means clustering algorithm [5]. We used the 'elbow method' heuristic to determine the optimal number of clusters in regard to Silhouette score and Bayesian and Akaike information criteria (Fig 2). We added two terminal states, representing a patient's discharge (survival) and a patient's expiration (death) at the ICU. We defined a 5×5 action space for the IV fluids and VP treatments, which resulted in 25 possible actions (Table 2, Fig. 3a). Except for zero doses of medicines as bin 0, we discretized the action space into per-drug quartiles and converted each drug at every timestep into an integer representing its quartile bin. We aimed to improve patient outcomes by focusing on optimizing patient survival. Hence, we defined the reward values of the patient expiration and discharge states to be -100 and +100, respectively, and 0 for all other states. Finally, Based on the observations in the dataset, we constructed a 3D 25x10x10 transition matrix where a cell (a,s,s') contains the probability of moving from state s to state s' by performing action a. Fig. 3b displays the observed transition matrix for action 0 ('no drug given').

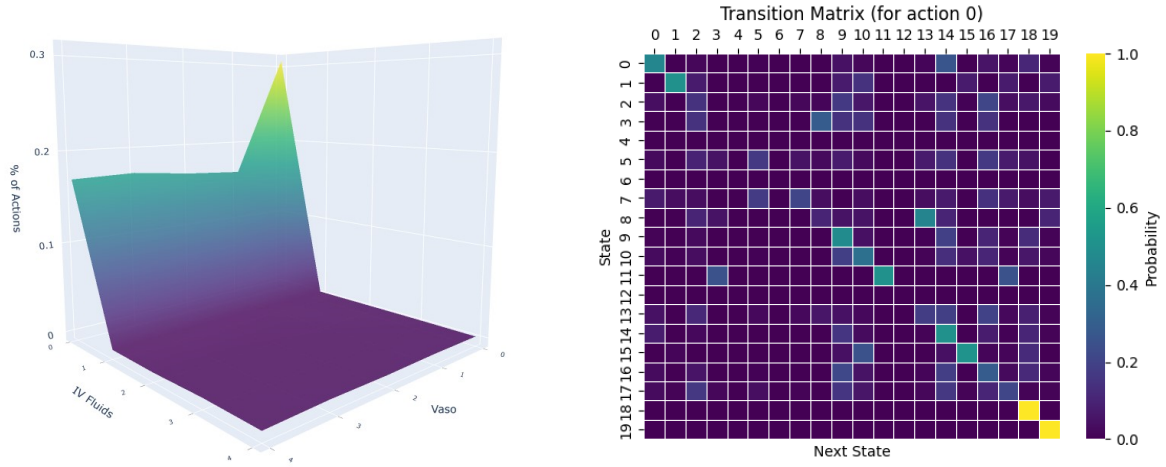
Discretized action	IV fluids (mL/1 hour)	Vasopressors (mcg/kg/min)
0	0	0
1	[0-75]	[0-0.05]
2	[75-150]]	[0.05-0.12]
3	[150-999]	[0.12-0.31]
4	>999	>0.31

Table 2: Range of drugs in all discretized actions. Option '0' corresponds to "no drug given". The remaining non-null dosages are divided into 4 quartiles. The combination of the two drugs made up 5x5=25 possible actions.



(a) **(b)**

Figure 2: Selection of the number of clusters (states) in the model, by **(a)** Bayesian information criterion (BIC) and Akaike information criterion (AIC), and **(b)** Silhouette score.



(a) **(b)**

Figure 3: **(a)** The discretized Action Space. The Z-axis represents the count of each combination of IV fluids and vasopressor treatments. **(b)** The observed transition matrix for action 0 ('no drug given').

2.4 Model Evaluation

We performed an offline evaluation approach named off-policy evaluation (OPE), in which a new policy (the "target policy") is evaluated by only using historical data D collected by a different policy (the "behavior policy"). In our case, the target and behavior policies are the learned AI policy and the clinicians' policy (as recorded in the eICU database), respectively. The evaluation is done without actually running the target policy in a real-time environment, this is because of safety concerns - deploying potentially suboptimal policy in a clinical environment poses serious risks to patient well-being. Specifically, we used the weighted importance sampling (WIS) method [7], in which the target policy is assessed by comparing the trajectories it generates using the historical data, to the actual trajectories contained in that historical data. The

comparison is done by giving more importance to observations that overlap in both trajectories. Formally, we define the per-step importance ratio by:

$$\rho_t = \pi(a_t \mid s_t) / \pi'(a_t \mid s_t)$$

and the cumulative importance ratio up to step t by:

$$\rho_{1:t} = \prod_{k=1}^{t} \rho_k$$

The average cumulative importance ratio at horizon t in the historical data D is defined as:

$$w_t = \frac{1}{|D|} \sum_{i=1}^{|D|} \rho_{1:t}^{(i)}$$

where $|D|$ is the number of trajectories in D, which is in our case the number of unique ICU hospitalizations. We define the discounted cumulative reward of the i'th trajectory with horizon H as:

$$Return^{(i)} = \sum_{t=1}^{H} \gamma^{t-1} r_t$$

Then the trajectory-wise WIS estimator is given by:

$$V_{WIS}^{(i)} = \frac{\rho_{1:H}}{w_H} \cdot Return^{(i)} = \frac{\rho_{1:H}}{w_H} \cdot \sum_{t=1}^{H} \gamma^{t-1} r_t$$

and the WIS estimator is the average estimate over all trajectories is given by:

$$WIS = \frac{1}{|D|} \sum_{i=1}^{|D|} V_{WIS}^{(i)}$$

The weighting is essential because if the target policy diverges from the recorded trajectory at some point - meaning it selects a different action than what's documented in the historical data - all future states and rewards will be changed. Since these alterations may not be observed in the data, this could lead to a decreased evaluation, even if the generated trajectory is actually a good one [6]. Therefore, the method put emphasizes on shared observation.

We generated 100 unique optimal AI policies by utilizing varying subsets of a randomly selected 80% of the eICU data and subsequently assessed the AI policies using the WIS technique on the remaining 20% of the data. The assignment of state membership for test set data points was based on their proximity to the cluster centroid of the training set.

3. Results

Fig. 4 shows the estimated policy value of the clinicians' actual treatments, the AI policy, a random policy, and a zero-drug policy, across 100 realizations of the environment based on the eICU database. It can be seen that in terms of expected value, the best AI policy performs much better than the clinicians' policy. In addition, for most of the realizations, the zero-drug policy outperforms the AI policy. Fig. 5 displays the learned value function for each state, along with the number of surviving ICU patients in that state from both the training and test sets. Fig. 6 presents the action frequency of the evaluation stage for the optimal AI policy and the clinician policy, highlighting distinct distribution patterns.

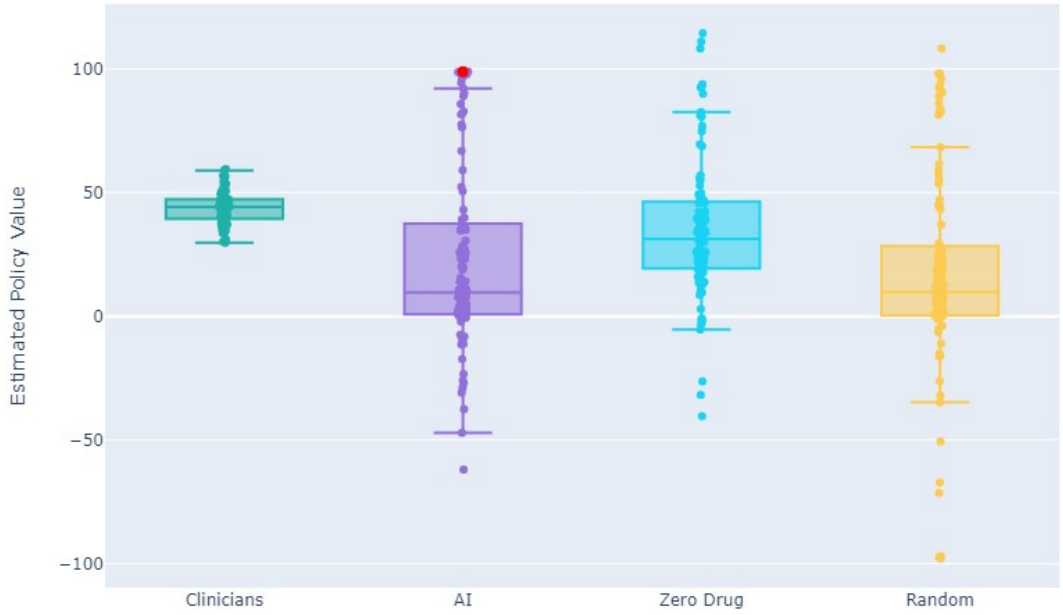


Figure 4: Comparison of the estimated value for the clinicians' actual treatments, the AI policy, a random policy, and a zero-drug policy, across 100 realizations of the environment. The best-found AI policy is marked with a red dot. The resulted AI distribution is statistically different (p-value < 0.05) from the Clinicians and Zero Drug distributions.

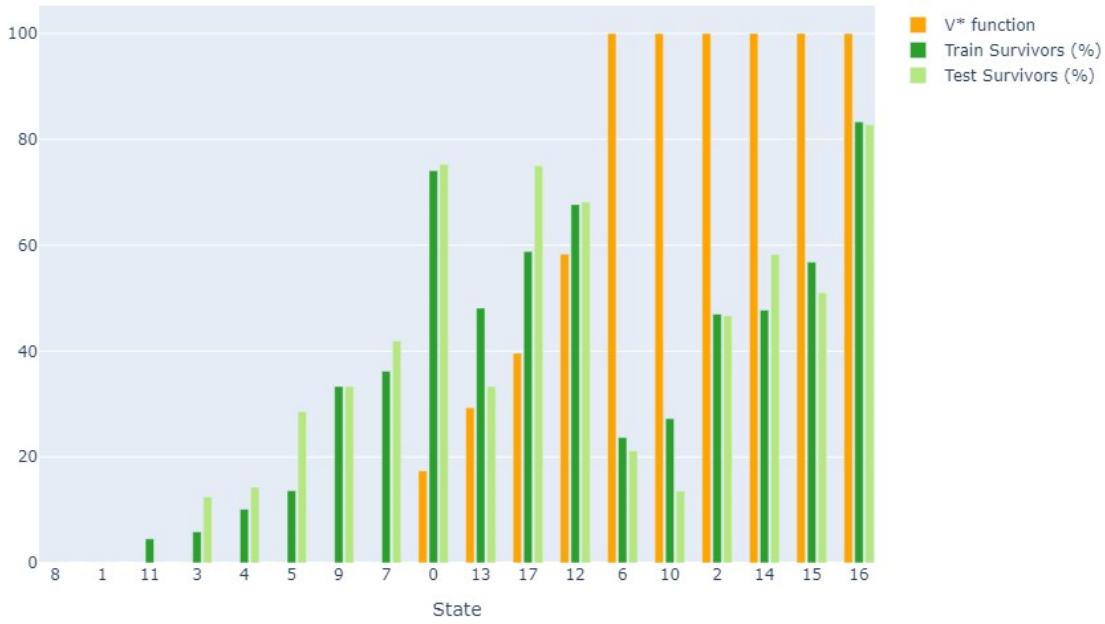


Figure 5: For each state, display the learned value function (orange bars), and how many patients who survived ICU were in this state in the training set (green bars) and the test set (light green bar).

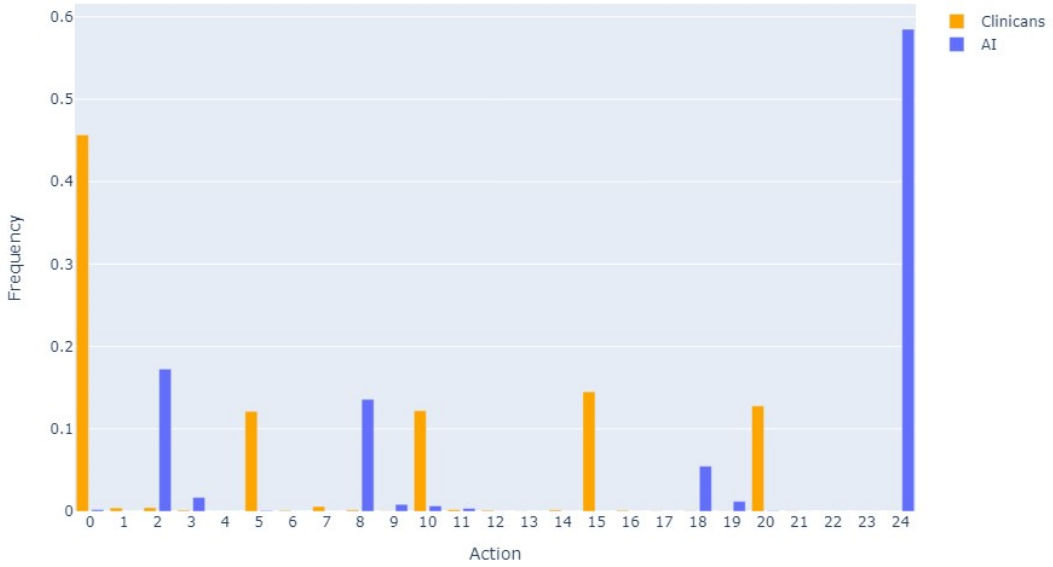
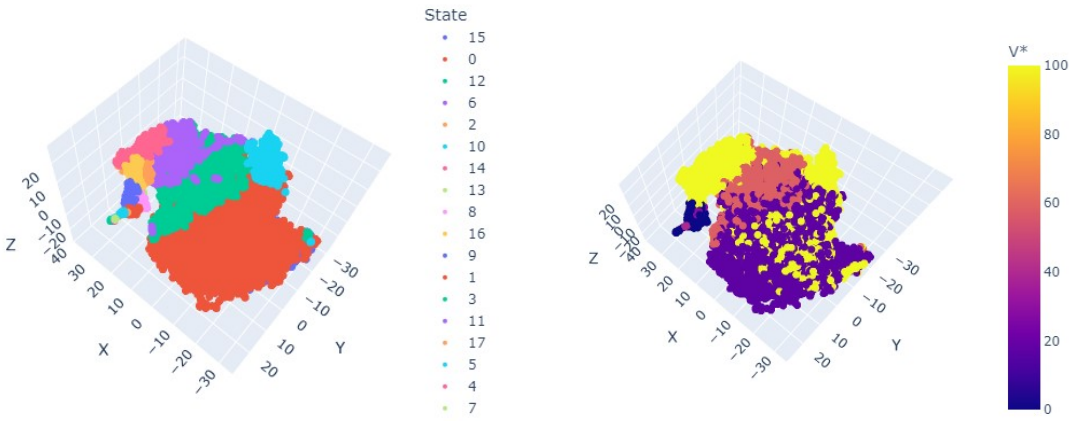


Figure 6: Action frequency of the evaluation stage for the optimal AI policy and the clinician policy. An action is a combination of IV fluids and VP treatments among 25 possible combinations)

Fig. 7 shows the state space that is reduced to a 3D space by the T-SNE method. The data points are taken from the training set and each represents a patient at some time point. It can be seen that states with similar optimal V-function tend to be close to each other.



(a) **(b)**

Figure 7: State space reduced to a 3D space by the T-SNE method. The data points are taken from the training set and each represents a patient at some time point. The data points coloring is by **(a)** the state association and **(b)** the optimal value function.

4. Discussion

The current methodology has several challenges to address.

Zero-drug policy outperforms AI policy: The zero-drug policy demonstrates superior performance compared to the AI policy (Fig. 4), which appears counterintuitive considering that septic patients typically require treatment. A potential explanation for this unexpected result is that the action of "Zero drug dosages" may be commonly applied to healthier patients who eventually survive ICU, and consequently the model erroneously associates "Zero drug dosages" with a positive outcome, forming a spurious correlation. Another possible reason, which will be elaborated further below, is that the evaluation method (WIS) puts emphasis on events recorded in the historical dataset, and since the frequency of "Zero drug dosages" in the dataset is high (Fig. 3a), WIS gives it more importance during the evaluation phase.

Environment Representation: The present approach involves discretizing both the state and action spaces, resulting in information loss and added noise. In addition, the trajectories are compressed into 1-hour intervals, potentially failing to encompass the entire sequence of events as certain variables can exhibit significant fluctuations on a minute-by-minute basis.

Reward Designing: At present, the reward function is focused only on ICU discharge mortality, which might not be the most optimal feedback because long-term rewards may neglect the

accomplishment of essential short-term objectives within the ICU, including stabilizing vital signs and managing acute symptoms. This oversight can limit the model's ability to provide timely and effective care. Furthermore, relying solely on ICU discharge mortality may result in delayed learning since the model receives feedback only at the end of the patient's stay, missing opportunities for real-time adaptation. Additionally, a focus on a single, manually defined reward function, such as ICU discharge mortality, raises the risk of introducing bias into the model's decision-making process. It may also fail to consider the full spectrum of factors influencing patient outcomes, potentially leading to suboptimal treatment decisions.

Evaluation Method: Evaluating a learned policy in an offline setting is a challenging task because it requires the assessment of counterfactual events – essentially, what might have occurred if the learned policy had chosen a different course of action than the behavior policy. In [3], a solution was implemented using the WIS method, which aims to narrow the gap between the learned and behavioral policies by assigning greater weight to shared decisions. Nevertheless, it's important to note that this approach has drawbacks (see also section 5.1 in [8]). Notably, it tends to favor simpler decisions, like taking no action in the case of a stable patient. This inclination towards simplicity resulted in the zero-drug policy receiving a higher estimated value compared to the AI policy in the majority of experiments (Fig. 4). Furthermore, this method has limitations in assessing out-of-domain strategies, which are treatments that are infrequently observed by the behavior policy. This limitation is noteworthy because, as evident from Fig. 6, there is a significant disparity in action frequency between the learned policy and the behavior policy, and consequently, the WIS method tends to focus on a limited portion of the treatment space, which can affect its ability to accurately assess the full spectrum of possible treatment strategies.

5. Conclusion

In this project, we reimplemented the methodology described in [3] which presents a proof of concept for utilizing offline reinforcement learning to acquire optimal treatment strategies for the management of sepsis shock. The methodology includes discretizing the state and action spaces into finite sets and then applying the Policy Iteration algorithm for learning an optimal value function. Model evaluation is conducted using the WIS method.

For future work, we aim to address the limitations of this methodology. Firstly, we plan to explore different reinforcement learning algorithms capable of handling continuous state and action spaces and incorporating a higher time resolution for patient trajectories. Secondly, regarding reward function engineering, we propose integrating the SOFA score at a specific time point into the reward mechanism, for more nuanced feedback throughout the trajectory. We also intend to employ Inverse Reinforcement Learning (IRL) to autonomously derive the reward function from the data, which may reduce potential bias from manually crafted reward functions. Lastly, we plan to investigate alternatives to the current model evaluation method that avoids the tendency to favor simpler decisions and may be better handle the assessment of counterfactual events.

References

[1] De Backer, D., Cecconi, M., Chew, M.S. et al. A plea for personalization of the hemodynamic management of septic shock. Crit Care 26, 372 (2022).

[2] Stephanie Hunter, Julie Considine, Elizabeth Manias, Nurse decision-making when managing noradrenaline in the intensive care unit: A naturalistic observational study, Intensive and Critical Care Nursing, Volume 77, 2023, 103429, ISSN 0964-3397,

[3] Komorowski, M., Celi, L.A., Badawi, O. et al. The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. Nat Med 24, 1716–1720 (2018).

[4] Batista, Gustavo & Monard, Maria-Carolina. (2002). A Study of K-Nearest Neighbour as an Imputation Method. Hybrid Intelligent Systems, ser Front Artificial Intelligence Applications. 30. 251-260.

[5] Jin, X., Han, J. (2011). K-Means Clustering. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.

[6] Li, L. A perspective on off-policy evaluation in reinforcement learning. Front. Comput. Sci. 13, 911–912 (2019).

[7] Thomas, P., Theocharous, G., & Ghavamzadeh, M. (2015). High-Confidence Off-Policy Evaluation. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1).

[8] Jeter, Russell, et al. "Does the" Artificial Intelligence Clinician" learn optimal treatment strategies for sepsis in intensive care?." arXiv preprint arXiv:1902.03271 (2019).