Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A self-supervised method for treatment recommendation in sepsis*

Sihan ZHU[1], Jian PU[†‡1,2]

[1]School of Computer Science and Technology, East China Normal University, Shanghai 200062, China

[2]Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai 200433, China

[†]E-mail: jianpu@fudan.edu.cn

Received Mar. 27, 2020; Revision accepted June 7, 2020; Crosschecked July 5, 2021

Abstract: Sepsis treatment is a highly challenging effort to reduce mortality in hospital intensive care units since the treatment response may vary for each patient. Tailored treatment recommendations are desired to assist doctors in making decisions efficiently and accurately. In this work, we apply a self-supervised method based on reinforcement learning (RL) for treatment recommendation on individuals. An uncertainty evaluation method is proposed to separate patient samples into two domains according to their responses to treatments and the state value of the chosen policy. Examples of two domains are then reconstructed with an auxiliary transfer learning task. A distillation method of privilege learning is tied to a variational auto-encoder framework for the transfer learning task between the low- and high-quality domains. Combined with the self-supervised way for better state and action representations, we propose a deep RL method called high-risk uncertainty (HRU) control to provide flexibility on the trade-off between the effectiveness and accuracy of ambiguous samples and to reduce the expected mortality. Experiments on the large-scale publicly available real-world dataset MIMIC-III demonstrate that our model reduces the estimated mortality rate by up to 2.3% in total, and that the estimated mortality rate in the majority of cases is reduced to 9.5%.

Key words: Treatment recommendation; Sepsis; Self-supervised learning; Reinforcement learning; Electronic health records

https://doi.org/10.1631/FITEE.2000127 **CLC number:** TP391.4

1 Introduction

Sepsis is a severe life-threatening medical emergency. It causes infections with organ failure and becomes a leading cause of patient mortality. Sepsis is often managed by two main interventions, intravenous (IV) fluid (adjusted for fluid tonicity) and vasopressor (VP). They focus on correcting the hypovolemia and counteracting vasodilation induced by sepsis. However, different dosage strategies for these two interventions greatly affect patient outcomes, and individual patients respond differently to treatments; this could have implications for patient mortality.

Treatment recommendations have been studied for a long time to assist inexperienced doctors (Chen JG et al., 2018) and to develop personalized-risk estimation (Katzman et al., 2018). The methods based on expert systems (Almirall et al., 2012; Chen Z et al., 2016; Gunlicks-Stoessel et al., 2016) heavily rely on prior knowledge of human experience. However, human knowledge summarized from experience is not necessarily the optimal choice for treatment. Data-driven approaches offer a new avenue to extract knowledge from large-scale data. Prior experience knowledge is then combined with personalized

[‡] Corresponding author

[*] Project supported by the National Natural Science Foundation of China (No. 61702186)

ORCID: Sihan ZHU, https://orcid.org/0000-0003-0960-8867; Jian PU, https://orcid.org/0000-0002-0892-1213

healthcare information to provide a treatment plan for each individual.

One straightforward method of data-driven approach is to mimic a doctor's prescriptions and learn the relationship between disease and drug categories (Bajor and Lasko, 2017; Zhang et al., 2017). However, it still relies on supervision by humans. This supervision is not the optimal and not correct in some situations. Such a learning paradigm is usually error-prone and problematic.

Instead of simply mimicing a doctor's prescriptions, another way is to learn from patients' responses and then give a personalized recommendation. This falls into the reinforcement learning (RL) setting (Kaelbling et al., 1995; Yu et al., 2019). Combined with deep learning for feature extraction, RL-based methods (Mnih et al., 2015; Wang ZY et al., 2016) regard patient treatments as independent states, and the objective is to learn the relationship between these states from the whole treatment process. For sepsis treatment, Raghu et al. (2017) proposed deep approaches to learn the optimal policy from delayed reward and the specific policy of physicians in the continuous state space. In Weng et al. (2017), policies were tested under state expected value based metrics of estimated mortality.

Nevertheless, there is no perfect solution for individual medical cases due to the uncertainty of treatment responses. A good treatment recommendation system could not only learn from a doctor's experience but also combine it with patients' responses. Therefore, it is important to decide which part of a doctor's policy to learn. Several studies considered the quality of predictions and used uncertainty evaluation which trades off between the mean and variance of RL (Shortreed et al., 2011; Asiain et al., 2018). These studies shed light on our evaluation of the variation of actions and confidence of policies. In clinical scenarios, one aims to estimate the value of the optimal policy based on the data collected by a doctor's policy. Off-policy value evaluation obtains unbiased estimation with variance under control (Jiang and Li, 2016). However, it provides less flexibility for cases with diverse requirements. Such flexibility would help policy exploration with doctors. For rare diseases, treatment predictions have a low confidence level. Otherwise, the method usually results in overfitting. Therefore, high-value recommendations should be both accurate and confident on reliable samples, and also should be salient on other samples.

We propose a deep architecture for the treatment recommendation problem with self-supervised learning. Samples are divided into reliable and unreliable sets according to the variance on actions and confidence on policies. We use transfer learning (TL) (Long et al., 2015; Lopez-Paz et al., 2016; Zhao et al., 2017) to shift unreliable samples into reliable ones and distill target sample information into original samples. We perform a variety of experiments on the large-scale publicly available real-world dataset MIMIC-III (multi-parameter intelligent monitoring in intensive care), with state-of-the-art RL-based comparison methods, and discuss the influence of specific parameters on risk control and trade-off on conservative decisions. The experimental results verify that our method provides flexibility on high-risk decisions with parameterized representations. The estimated mortality rate in the hospital is reduced to 2.3%, and the estimated mortality rate in the majority of cases is reduced to 9.5%.

2 Related works

The clinical treatment recommendation process can be considered as a sequential decision-making problem which suits RL settings well. Various RL approaches have been proposed to model the problem into RL settings (Nemati et al., 2016; Yu et al., 2019) and solve the problem by maximizing the accumulated reward. Many works on modeling RL setting on sepsis treatment (Komorowski et al., 2018; Saria, 2018) use static and dynamic clinic indices to represent the patient state, and the medical interventions, especially IV and VP with different dosages, are modeled as actions.

Several works on RL for sepsis treatment are based on variants of deep RL (DRL) models, such as deep Q-network (DQN) (Mnih et al., 2015) and dueling DQN (DDQN) (Wang ZY et al., 2016). They used the deep learning approach to extract latent representations of patient features and states, and found the best policy evaluation according to the accumulated reward. This assesses the disease severity or the survival rate of the patient. In variants of the RL framework on sepsis, Raghu et al. (2018) used a recurrent neural network (RNN) for intensive care unit (ICU) stay record sequence encoding tied to a continuous state-space RL framework. A Gaussian process tied with recurrent long-short term memory (LSTM) layers (Futoma et al., 2017) showed the effectiveness in sequence encoding and was clinically interpretable. However, a major concern of an RL-based framework is the representation stability. Especially for sepsis, patients respond differently to treatment. This affects the uncertainty of clinical indices and makes the RL framework hard to train. To make RL policy clinically credible and stable, Peng et al. (2018) proposed a mixture of the DRL framework and a conservative kernel RL framework. Wang L et al. (2018) combined the benefits of supervised learning and RL.

The main idea of our work benefits from self-supervised learning (SSL) and TL. SSL is for improving learning performance when labeled data is scarce. It exhibits promising results in semi-supervised setting only when partial samples have labels (Gidaris et al., 2018; Zhai et al., 2019) and video tasks where annotation is costly (Vondrick et al., 2016; Li et al., 2019). Recently, SSL helps improve the robustness on adversarial examples and label corruption, and also benefits out-of-distribution detection on difficult, near-distribution outliers (Hendrycks et al., 2019). Our proposed method is considered as a self-supervised method since we refine the unreliable samples using the reliable samples in the training stage. In RL settings, states may have different distributions and behave differently. With this concern, information learned from good situations can help a model perform well in bad situations. TL methods try to use models trained in a source domain to have a desirable performance in the target domain, and instance-based TL seeks to find the image of samples in the source domain where the original trained model makes reasonable decisions. Recent research on the variational auto-encoder (VAE) (Kingma and Welling, 2014; Kingma et al., 2016) made effective use of latent representations of states; deep adaptation models (Long et al., 2015) used maximum mean discrepancy (MMD) loss to measure the distance between domains in deep adaptation networks. However, VAEs tend to ignore the latent variables when combined with a decoding distribution that is too flexible. Info-VAE (Zhao et al., 2017) mitigates these problems. Our proposed method combines VAE for feature representation and MMD for measurement between domains, and constructs real sample approximations.

3 Formulations and preliminaries

In this section, we formulate the treatment plan with sequence decision and then provide the architecture using deep learning to extract latent features into separate value and advantage. From uncertainty risk evaluation, we transfer sample distribution to an ideal domain to make the decision reliable and stable.

3.1 Problem formulations

In an RL setting, the clinical treatment recommendation process can be represented by a Markov decision process (MDP). The process is represented by the observed state s, which takes an action a according to policy π and obtains a reward r from the environment. π assigns a probability to actions in each state.

3.1.1 Reinforcement learning in treatment

Various treatment plans for patients can be seen as solutions to various sequential decision problems (Nemati et al., 2016), and RL methods try to find the best policy which provides decision sequences that maximize the expected reward (Kaelbling et al., 1995).

The goal of an RL agent is to maximize the expected long-term discounted return $\mathbb{E}[\sum_t \gamma^t r_t]$, where γ is the discount factor that represents the trade-off between current and future rewards. The optimal value function and state-action value function are defined as $V_\pi^*(s) = \max_\pi \mathbb{E}[\sum_t \gamma^t r_t | s, \pi]$ and $Q_\pi^*(s,a) = \max_\pi \mathbb{E}[\sum_t \gamma^t r_t | a, \pi]$, respectively. The state-action value function $Q_\pi^*(s,a)$ satisfies the Bellman equation $Q_\pi^*(s,a) = r(s,a) + \gamma \max_{a'} \mathbb{E}[Q_\pi^*(s',a')]$ and is optimized by minimizing the temporal difference (TD) error of $r(s,a) + \gamma Q_\pi^*(s',a') - Q_\pi^*(s,a)$. The primary RL method seeks a solution to the problem

$$a_{\text{opt}} = \underset{a \in \mathcal{A}}{\arg\max}\ Q_\pi^*(s,a), \tag{1}$$

where s represents the patient state, a is an action provided by the optimal policy π, and $\mathcal{A}$ is the set of all possible actions.

3.1.2 State approximation

We consider that in the medical environment, state s is not exactly the real state of the patient but an observation which can be seen as an approximation of the real state s^*. For this reason, clinical features are measurements related to not only the patient's actual body state but also the environment. Thereby, problem (1) can be reformed as

$$a_{\text{opt}} = \underset{a\in\mathcal{A}}{\arg\max}\ Q_\pi^*(s^*, a) \quad \text{s.t.} \quad s^* \approx s. \tag{2}$$

Specifically, we attempt to find the optimal policy π and the mapping from the observed state s to the real state s^* simultaneously:

$$\begin{cases} a_{\text{opt}} = \underset{a\in\mathcal{A}}{\arg\max}\ Q_\pi^*(f_t(s), a), \\ f_t : s \to s^*, \end{cases} \tag{3}$$

where f_t denotes the mapping from the observed state s to the real state s^*. In this study, the mapping f_t is implemented by an auto-encoder based transfer module. Final decisions are made by policy π on observation $f_t(s)$.

3.1.3 Value-advantage separation

In the medical environment, treatment effects rely on both the patient's physical state and the doctor's different prescriptions, and the dueling net (Wang ZY et al., 2016) architecture maintains separate value and advantage functions corresponding to the above characteristics. The aggregating module is updated as follows:

$$Q_\pi(s,a)=V_\pi(s,a)+\left[A_\pi(s,a)-\frac{1}{|A_\pi|}\sum_{a'}A_\pi(s,a')\right], \tag{4}$$

where V is the value of the patient state and A is the advantage of prescription according to the specific policy π. Both V and A are outputs of the dueling net. The final Q-value of action a on patient state s is calculated using Eq. (4), which is related to the survival rate of the patient. π is a policy provided by the model. With the basic dueling structure, we consider the value and advantage separately to divide samples and reconstruct biased samples.

3.2 Framework preliminaries

We use variance and confidence of separate state value-advantage pairs to evaluate sample reliability, which divides sample distribution into the high-uncertainty (unreliable) domain and low-uncertainty (reliable) domain. With the transfer model, we reconstruct samples into an ideal domain that keeps the final Q-function away from high-risk uncertainty. Although it is a strong assumption that samples behave differently in bimodal distributions, the shift between two distributions should be controlled. By adjusting the parameter value according to the performance on cross validation, we can obtain models with different specialities, which we will discuss in Section 5 together with experiments.

3.2.1 Model architecture

As discussed in Section 3.1, the dueling architecture generates latent representations for patient states and separates the value and advantage. Eq. (4) combines the final Q-value with both value and advantage to update the Q-function. The following loss is considered in the RL framework:

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{U}(\mathcal{D})}[(y_i^{\text{DDQN}} - Q(s,a;\theta_i))^2], \tag{5}$$

where quadruples (s,a,r,s') are sampled from the replay buffer $\mathcal{U}(\mathcal{D})$. The true value y_i^{DDQN} is obtained by the target network and finally converges to the reward of actions. The general framework of our work is shown in Fig. 1. It contains two modules with four stages iteratively cycling during the training process. First, the dueling module provides encoding of patient clinic indices as state and outputs state-action value parameters (discussed in Section 4.1). Then we use a risk-evaluation score to find samples with high-risk uncertainty (discussed in Section 4.2) and divide them into reliable set S and unreliable set $\hat{S}$. Next, samples from the unreliable set $\hat{S}$ are reconstructed with the transfer module via the MMD distance (discussed in Section 4.3). Finally, the model computes the Q-value with reconstructed samples and updates the state-action value function (discussed in Section 4.4).

3.2.2 Transfer learning and distillation

VAE is used to shift the sample distribution between different domains. We use the MMD distance (Long et al., 2015) to measure the gap between domains. Distillation via the privileged learning (PL) method (Hinton et al., 2015) attempts to learn additional information from high-quality samples or representations. Training on the PL framework aims

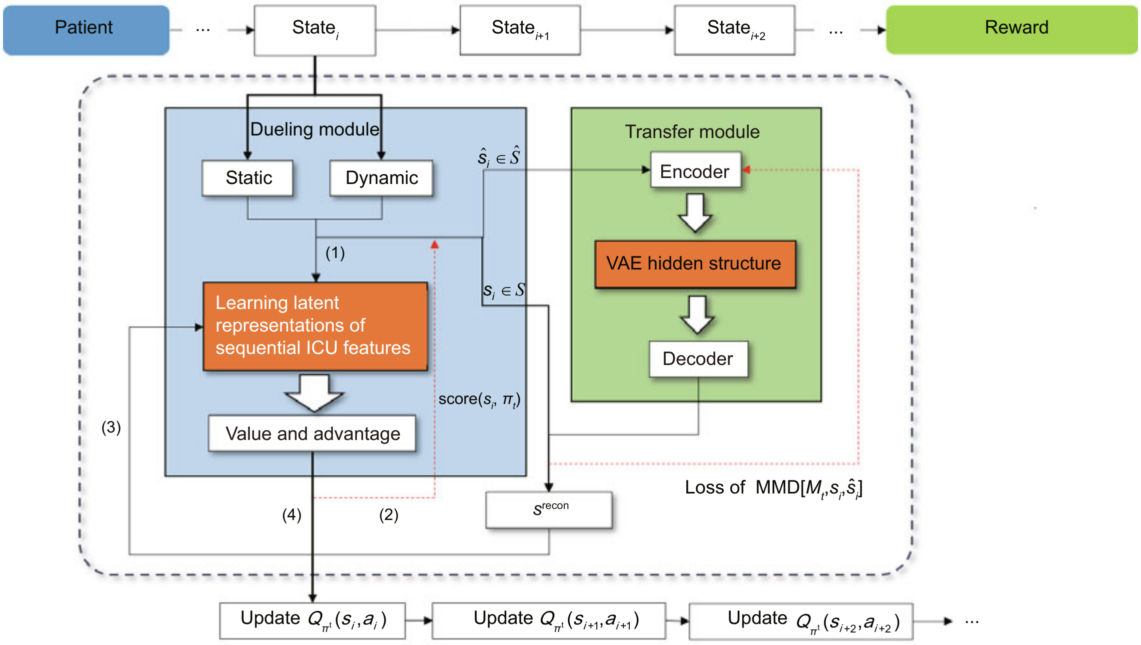

Fig. 1 General architecture of our work

Each state iteration includes four steps: (1) features go to the dueling structure to obtain the value and advantage; (2) the value and advantage are used to calculate uncertainty and a dividing score, and then the original states are divided into reliable set S and unreliable set $\hat{S}$; (3) ambiguous $\hat{s}$ is transferred to obtain $s^{\rm recon}$; (4) $s^{\rm recon}$ is used to obtain the final value and advantage, and Q-function is updated in a privileged learning form

to distill high-level information into the model. This benefits the performance on normal samples.

4 The proposed method

This section starts with an overview of our method. Our framework deals with the state value and action advantage separately using a part of the dueling structure, which we will discuss in Section 4.1. In Section 4.2, we propose an evaluation method to divide samples into a reliable domain with low uncertainty and an unreliable domain with high uncertainty, and use self-supervised learning with the help of the transfer module. In Section 4.3, we transfer samples to an ideal domain via the MMD distance. In Section 4.4, we reconstruct the sample and train the model via PL-based methods. We analyze our algorithm in Section 4.5.

4.1 Dueling net with sequence

Patient states can be represented by static features (such as demographics) and dynamic features (such as lab values). Dynamic features within one course of treatment are segmented into sequences corresponding to one specific action. We put a dynamic feature sequence in one course of treatment in LSTM to learn deep relevance and generate highly characteristic features, and use fully connected layers to combine those with static features. The feature extraction network architecture is shown in Fig. 2.

In Section 3.1, we use Eqs. (2) and (3) to extend the basic Q-learning problem to an unreliable environment with a mapping function f. In our proposed framework, the mapping f in Eq. (3) is instantiated using a VAE, and the final Q-function with VAE model $M(\cdot)$ is modified as follows:

$$Q_\pi(s,a) = V_\pi[M(s;\theta),a] + \Big\{A_\pi[M(s;\theta),a] - \frac{1}{|A_\pi|}\sum_{a'} A_\pi[M(s;\theta),a']\Big\}. \tag{6}$$

A key concern for directly optimizing the Q-value in a dueling framework using the original loss and transferred samples is that the model uses only the information from a part of samples. We thereby apply a privileged learning framework to distill

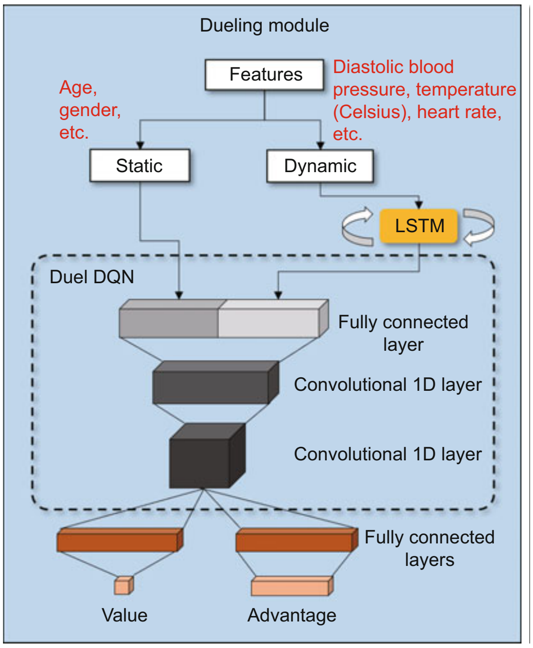

Fig. 2 Dueling module of our framework

Dynamic features in a sequence are handled by long-short term memory (LSTM) to obtain latent representations, which are combined with static features and turned into separate value-advantage vectors

information from transferred samples into original ones. This can be seen as self-supervised learning from reliable samples to unreliable ones.

4.2 Uncertainty evaluation

In this subsection, we analyze how states and actions affect the future reward. Let s_t denote a patient state sequence in RL settings and buffer $\mathcal{D} = \{e_1, e_2, \ldots, e_t\}$ consists of experience tuples $\boldsymbol{e}_t = (s_t, a_t, r_t, s_{t+1})$ which are segmented from sequence $\mathcal{P} = \{s_1, a_1, r_1, \ldots, s_N, a_N, r_N\}$. Mini-batches of experience (s, a, r, s') are sampled from $\mathcal{D}$ uniformly at random. The evaluation items take the forms as

$$\begin{cases} v(s,\pi) = \mathbb{E}_{a\in\mathcal{A}(s)}[(Q_\pi(s,a) - \hat{Q}_\pi(s,a))^2], \\ c(s,\pi) = \dfrac{Q_\pi(s,\hat{a})}{\int_{\mathcal{A}(s)} Q_\pi(s,a)}, \end{cases} \tag{7}$$

where

$$\begin{cases} \hat{Q}_\pi(s,a) = \mathbb{E}_{a\in\mathcal{A}(s)}[Q_\pi(s,a)], \\ \hat{a} = \arg\max\limits_{a\in\mathcal{A}(s)} Q_\pi(s,a). \end{cases} \tag{8}$$

Here, $v(s,\pi)$ and $c(s,\pi)$ denote the variance and confidence for the action on state s on policy π, respectively.

In most cases, a patient in a desirable condition can be cured using a proper treatment. However, the choice of the treatment plan in some extreme situations is worthless in the policy learning process. Consider the following two extreme situations:

Situation 1: a patient cannot be cured even using the ideal treatment;

Situation 2: a patient is in a good condition and can be cured using any treatment.

Reflected in the clinic log, the variance of expected return in prescriptions taken for a bad situation like situation 1 is low and often leads to a zero reward (dead). On the contrary, in situation 2, the expected return is high, but a doctor's policy gains little effect. However, prescriptions that perform effectively and have a significant impact on patient states often result in a higher variance of actions. If the policy is effective in this circumstance, then we will have high confidence in the chosen action.

The variance of actions evaluates the probability that the state drops into extreme cases. The confidence of the optimal action evaluates the quality of the chosen treatment policy. Using the definition of variance v and confidence c, we define the proposed evaluation metric as high-risk uncertainty (HRU) in the form of d-score:

$$\text{d-score}(s,\pi) = v(s,\pi)\exp\left[\frac{c(s,\pi^{\text{opt}}) - c(s,\pi^{\text{doc}})}{T}\right], \tag{9}$$

where T is the temperature that controls the influence on choice of actions; i.e., a lower T makes the model care more about states with choices differing from a doctor's, and a higher T makes the model care more about states with large variance on action values. π^{doc} is the policy with chosen actions, and π^{opt} is the greedy policy with the best actions. If there is a big difference between actions with maximal confidence under the optimal policy and a physician's policy, the score tends to be large. On the contrary, if the variance of actions is small, the policy's choice of actions makes less difference to the expected value. In this case, it is not necessary to pay much attention to choosing the right action in the policy. The treatment for this state is ambiguous and needs deeper discussion. In this case, d-score tends to be small. We use d-score to divide samples into two distributions mentioned above by the threshold ϵ, and transfer ambiguous ones into a clear distribution. ϵ is set according to the distribution of samples' d-score on cross-validation via grid search or the elbow method.

4.3 Transfer learning via MMD

In Section 4.2, we assume that a doctor's poor choice attributes to the poor observation of the real patient state s^*. Samples that perform undesirably under a specific physician policy have low evaluation d-score s, which results in a bad observation on the patient's real state features. In a self-supervised learning setting, samples learn from themselves. We divide sample states using the evaluation metric d-score and use TL on divided domains to mitigate the observation shift. The MMD distance is used to measure the gap between domains. This can be formulated as follows:

$$\mathrm{MMD}[f,p,q]=\sup\{\mathbb{E}_{x\sim p}[f(x)]-\mathbb{E}_{y\sim q}[f(y)]\}, \quad (10)$$

where p is the distribution of x and q is the distribution of y. f is the mapping function which can be a fully connected network or a convolutional layer. We divide the samples into two distributions and minimize the following MMD distance to transfer the auto-encoder's outputs to a reliable domain:

$$\mathrm{MMD}[M,S,\hat{S}]=\sup\{\mathbb{E}_{s\sim S}[s]-\mathbb{E}_{\hat{s}\sim\hat{S}}[M(\hat{s};\theta_i)]\}. \quad (11)$$

We denote clear samples with low uncertainty by s in set S and ambiguous ones by $\hat{s}$ in set $\hat{S}$. Details of the transfer module can be seen in Fig. 3. The dueling structure provides latent representations for sample features. As the training process progresses, value and advantage converge to the optimal Q-function. This makes the d-score dividing more clear. Threshold-based methods using the temperature T in Eq. (9) and threshold ϵ on d-score control the gap between source distribution and target distribution. The flexibility makes the model pay attention to different Q-value distributions.

4.4 Self-supervised training via PL

In the reconstruction process, we use weight-based methods and knowledge distillation based methods to extract information from transferred samples and help the whole training process on the proposed framework.

4.4.1 Naive weight-based training

One way to treat samples of different importance is sample weighting. We could train the model to use a sample weight calculated using the aforementioned d-score and normalized by a softmax function σ(·):

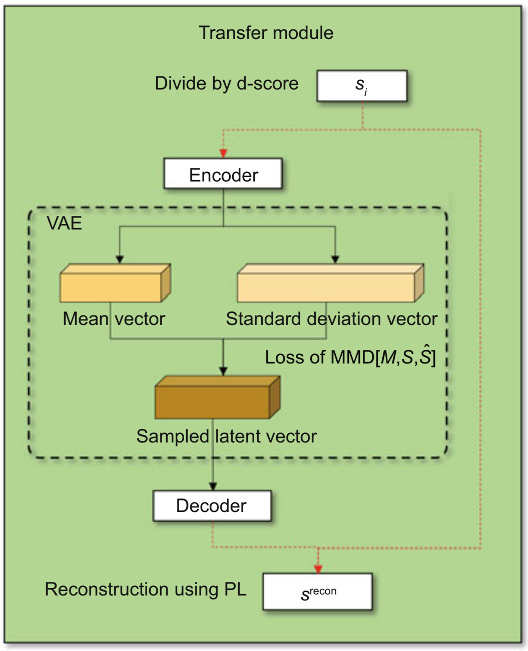

Fig. 3 Transfer module of our framework
Samples are divided by d-score and transferred through VAE using MMD loss between sample distributions

$$\mathrm{weight}(s)=\sigma(\text{d-score}(s,\pi)). \quad (12)$$

This naive approach attempts to use different importance on samples, but may contaminate clear samples.

4.4.2 Distillation in PL

In the reconstruction process, samples reconstructed with weight-based methods could lose variety of information; the difference between sample domains could trace back to information asymmetry. Distillation (Hinton et al., 2015) and privileged information (Vapnik and Izmailov, 2015) are two techniques that enable machines to learn from other ones. Moreover, Lopez-Paz et al. (2016) unified these two into one framework and extended it to semi-supervised scenarios. Specifically, it regards high-quality samples as "teachers" and low-quality samples as "students," and extracts high-level knowledge of "teacher" samples into "student" sample models. We divide the sample into different domains and use TL approaches to acquire an image into the source domain. To use the learning effect of samples in different distribution domains, we use privileged learning methods to guarantee an optimal reconstruction process of deep models.

Sample features are transferred into a new domain. They can be seen as “teachers” to our model f. We train model f with both original samples and transferred ones in the form of PL. The general PL framework takes the forms as

$$\begin{cases} f^{\mathrm{t}} = \underset{f\in\mathcal{F}_t}{\arg\min} \sum_{i=1}^{n} L\{y_i, \sigma(f(x_i))\}, \\ f^{\mathrm{s}} = \underset{f\in\mathcal{F}_t}{\arg\min} \sum_{i=1}^{n}\{(1-\lambda)L[y_i, f(x_i)] \\ \qquad + \lambda L[\hat{y}_i, f(x_i)]\}, \\ \hat{y}_i = \sigma\left(\dfrac{f^{\mathrm{t}}(x_i)}{T_{\mathrm{p}}}\right), \end{cases} \tag{13}$$

where f^{t} denotes the “teacher” model from high-quality samples and f^{s} denotes the student model from normal samples. $\sigma(\cdot)$ is an activation function. T_{p} is the temperature. We use the PL framework to distill information from transferred samples provided by VAE as follows:

$$\begin{aligned} \pi^{\mathrm{t}} = \underset{\pi}{\arg\min} \sum_{i=1}^{n} &\mathbb{E}_{(s,a,r,s')\sim\mathcal{U}(\mathcal{D})}\{L[y_i, \\ &Q_\pi(M(s;\theta_i), a)]\}, \end{aligned} \tag{14}$$

$$\begin{cases} \pi^{\mathrm{s}} = \underset{\pi}{\arg\min} \sum_{i=1}^{n} \mathbb{E}_{(\hat{s},a,r,s')\sim\mathcal{U}(\mathcal{D})}\{(1-\lambda) \\ \qquad \cdot L[y_i, Q_\pi(\hat{s}, a)] + \lambda L[\hat{y}_i, Q_\pi(\hat{s}, a)]\}, \\ \hat{y}_i = Q_{\pi^{\mathrm{t}}}(s^{\mathrm{recon}}, a), \end{cases} \tag{15}$$

where s is the original state, a the action, and $\hat{y}_i$ the soft label on the reconstructed sample s^{recon}. Model $M(\hat{s};\theta_i)$ uses the VAE model $M(\cdot)$ with parameter θ_i to obtain the transferred state. For the clear state set S, the training process needs only to optimize the “teacher” model, which is equivalent to the basic Q-learning process. For the ambiguous state set $\hat{S}$, the training process goes according to the aforementioned PL framework, and the reconstructed sample s^{recon} takes the form as

$$s^{\mathrm{recon}} = l\hat{s} + (1-l)M(\hat{s};\theta_i), \tag{16}$$

where l is the reconstruction parameter and controls the portion of the information distilled from transferred samples. If $l \to 1$, the sample domain tends to be the original sample distribution, and the transfer module is not working. The Q-learning model learns the original Q-function (4). If $l \to 0$, the sample domain tends to be the transferred sample distribution, and function f^{s} reduces to

$$\pi^{\mathrm{s}} = \underset{\pi}{\arg\min} \sum_{i=1}^{n} \mathbb{E}_{(\hat{s},a,r,s')\sim\mathcal{U}(\mathcal{D})}\{L[\hat{y}_i, Q_\pi(\hat{s}, a)]\}. \tag{17}$$

In the dueling structure, we optimize Q-function (4) and update the network parameter using loss (5). The “teacher” model includes rich information of ambiguous samples. The updating process takes the forms as

$$\begin{aligned} \pi^{\mathrm{t}} = \underset{\pi}{\arg\min} \sum_{i=1}^{n} &\mathbb{E}_{(s,a,r,s')\sim\mathcal{U}(\mathcal{D})}\{[y_i \\ &- Q_\pi(M(s;\theta_i), a)]^2\}, \end{aligned} \tag{18}$$

$$\begin{aligned} \pi^{\mathrm{s}} = \underset{\pi}{\arg\min} \sum_{i=1}^{n} &\mathbb{E}_{(\hat{s},a,r,s')\sim\mathcal{U}(\mathcal{D})}\{(1-\lambda) \\ &\cdot [y_i - Q_\pi(s,a)]^2 + \lambda[\hat{y}_i - Q_\pi(s,a)]^2\}. \end{aligned} \tag{19}$$

4.5 Algorithm

With all the aforementioned parts, we propose the HRU-control method, as shown in Algorithm 1.

The HRU-control framework trains two deep networks, dueling net for Q-function $Q_\pi(s,a)$ optimization and VAE $M(\cdot)$ for TL. The dueling net is trained with sample states provided by LSTM $f(\cdot)$, learns the relationship between state sequences of one action course, and gives latent representations. Survival status is used as the final reward

Algorithm 1 The proposed HRU-control method

```
Read the patient state sequence of one course like
  $\mathcal{P} = \{s_1, a_1, r_1, \ldots, s_N, a_N, r_N\}$
Split the sequence into $(s, a, r, s')$ tuples stored in
  the buffer $\mathcal{D}$
for $t = 0$ to $T$ do
  Sample $(s, a, r, s')$ in $\mathcal{D}$ at random
  if d-score$(s, \pi) > \epsilon$ then
    Divide $s$ into $\hat{S}$
  else
    Divide $s$ into $S$
  end if
  Minimize the loss of MMD$[M, S, \hat{S}]$
  Update VAE model $M(\cdot)$
  Reconstruct $s^{\mathrm{recon}}$ for an ambiguous sample set
  Minimize PL loss to obtain the optimal $Q_\pi$
end for
```

of $+15/-15$ to help Q-function converge to policy π. The evaluation d-score(s,π) for high-risk uncertainty divides the samples in buffer $\mathcal{D}$ into sets S and $\hat{S}$. The VAE model is trained on the loss of MMD$[M,S,\hat{S}]$ using backpropagation to update model parameters. Finally, we update the dueling module parameters using the Q-learning updating function with reconstructed sample s^{recon} in the form of privileged learning.

5 Experiments

In this section, we present experimental results of our method on the open dataset MIMIC-III.

5.1 Dataset and cohort

Our experiments were carried out on the MIMIC-III v1.4 database (Johnson et al., 2016), which is large and publicly available. It includes all patient admissions to ICU from 2001 to 2012. Sepsis-3 criteria were set to identify patients with sepsis (Singer et al., 2016). Following Raghu et al. (2017), our cohort consists of 15 415 patients with the age ranging from 18 to 91. The dataset is summarized in Table 1.

Table 1 Dataset cohort statistics for subjects fulfilling the sepsis-3 criteria

Subject	Female (%)	Mean age	Total number of patients
Survivor	44.1	63.9	13 535
Non-survivor	44.3	67.4	1880

5.2 RL settings and preprocessing

We present RL settings in medical environment experiments where patient features are extracted into states and treatment plans as actions, and final patient status is used to evaluate the delayed reward.

5.2.1 Features and states

Following Raghu et al. (2017), relevant clinical features include static variables and dynamic variables (time-series variables), which are sliced in a given four-hour window. Forty-eight physiological features used in our experiments include 8 demographics/static features, 24 lab values, 12 vital signs, 3 intake/output events, and 1 miscellaneous variable; this yields a 48 × 1 feature vector, which is denoted as the state s in the RL setting. The physiological features used in our model are shown in Table 2. Features are standardized and rescaled into 0–1. Those features with large values are dealt with by log transformation. Samples with ≥ 8 missing variables are excluded.

5.2.2 Actions and rewards

Treatments are discretized into 25 actions, which yields a 5 × 5 action space. The 5 × 5 action space includes two axes of IV fluid and maximum VP dosage in the given four-hour time slice. At the terminal state of a patient's trajectory, the reward is set to +15 if the patient is discharged; otherwise, the reward is −15. The learned policy in the action space is shown in Fig. 4. The physician's policy follows mainly the rule that IV dose and VP dose increase with the severity of the disease (organ failure). For mild symptoms, vasopressors are not usually prescribed unless the symptoms reach a certain degree. For our model, there are slight differences from the physician's policy. Despite some treatments with no drugs given to mild symptom patients (left bottom), most decisions follow the physician's rule that dosages align with symptoms. The vasopressor dosages seem less conservative, which results in a higher dosage of vasopressors in the middle part. It might be a signal of treatment trade-off between relying on external medication and the body's immune system, including the other side-effects that could not be reflected in the mortality rate. Deeper

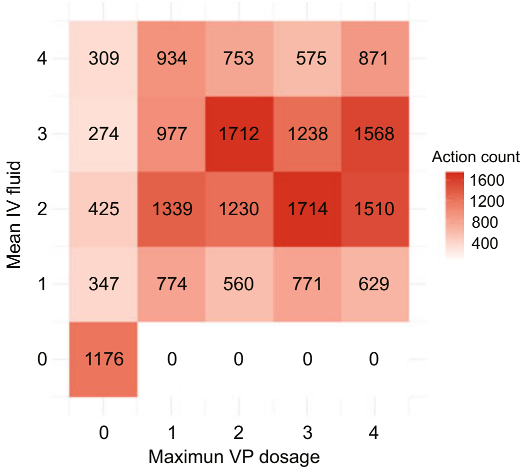

Fig. 4 Learned policy in the action space

Table 2 Physiological features in the model

Physiological feature	Detailed information
Demographics/Static features (8)	Shock index, elixhauser, SIRS, gender, re-admission, Glasgow coma scale, sequential organ failure assessment, and age
Lab values (24)	Albumin, arterial pH, calcium, glucose, hemoglobin, magnesium, partial thromboplastin time, potassium, arterial blood gas, blood urea nitrogen, serum glutamic-pyruvic transaminase, chloride, bicarbonate, international normalized ratio, sodium, arterial lactate, CO_2, creatinine, ionised calcium, prothrombin time, platelet count, serum glutamic-oxaloacetic transaminase, total bilirubin, and white blood cell count
Vital signs (12)	Diastolic blood pressure, systolic blood pressure, mean blood pressure, $PaCO_2$, PaO_2, FiO_2, PaO/FiO_2 ratio, respiratory rate, temperature, weight, heart rate, and SpO_2
Intake/Output events (3)	Fluid output (4 hourly period), total fluid output, and mechanical ventilation
Miscellaneous variable (1)	Timestep

The corresponding number of features is also given in the bracket in the first column

clinical insights and considerations need to be further discussed in future work.

5.3 Metrics and evaluation

As the patient's health status is the clinician's most prominent indicator, the reward is related to the patient's survival. Following Raghu et al. (2017), Weng et al. (2017), and Wang L et al. (2018), we use in-hospital mortality rate to evaluate the performance of our method. These works showed that the expected return Q^* learned by the optimal policy π^* is negatively correlated with the mortality rate with high correlation. The empirically estimated mortality rate of each Q-value unit is calculated using a mortality-expected return function acquired from learned representations with a probabilistic output ranging from 0 (discharged) to 1 (died). Finally, the estimated mortality rate is obtained by averaging all these values in the corresponding units (Raghu et al., 2017; Weng et al., 2017). Although there is still difference between the estimated mortality rate and the real one, it is still a widely used metric for computational experiments.

Another evaluation metric is used to compare the difference between the agent and a doctor's policies. We use the Jaccard coefficient to measure the consistency. The Jaccard coefficient is defined as $(1/M)\sum_{i=1}^{M}(1/T_i)\sum_{t=1}^{T_i}|U_t^i \cap \hat{U}_t^i|/|U_t^i \cup \hat{U}_t^i|$, where M is the number of patients and T_i is the number of ICU days of the i^{th} patient. The medication and dosage at day t for the i^{th} patient are defined as U_t^i and $\hat{U}_t^i$, respectively, one from the learned policy and the other from the doctor.

Furthermore, we try to fairly compare evaluations among different Q-value regions so that performance on cases with different severities could be discussed. The learned Q-value units are divided into regions with small intervals and used to calculate averaging estimated mortality separately. The estimated mortality in the region with most Q-value units is defined as estimated major mortality. This part of comparison will be discussed in Section 5.5.

5.4 Competitors

Competitors of the experiments include three parts.

1. Basic-LSTM (baseline, BLSTM)

This baseline uses LSTM to deal with a sequence of states and outputs the final action as a result of supervised learning in sequence. While the LSTM structure focuses on longitudinal records, states are equally treated using two-layer fully connected networks and concatenated to obtain per-step action prediction.

2. Reward-LSTM (RLSTM)

Reward-LSTM is a variant of basic-LSTM. It has extra reward signals for the feedback of mortality, which makes the prediction a Q-learning policy. It uses a tabular Q-learning approach to learn the Q-values that fit a doctor's prescriptions and survival situations.

3. Dueling deep Q-network (Q-learning, DDQN)

The DQN method (Mnih et al., 2015) uses a deep neural network to learn the corresponding Q-value function, and DDQN (Wang ZY et al., 2016) combines DQN with the dueling structure which divides the Q-value into separate values and advantages to revise the value function.

5.5 Results and analysis

We present the results and analysis for comparison models and case studies for the effect of hyperparameters that control the model flexibility.

5.5.1 Model comparison

Table 3 shows the results of the estimated mortality rate for all the chosen comparison models on MIMIC-III. The results show that Q-learning based approaches (DDQN and the proposed HRU-control method) outperform recurrent deep approaches (basic-LSTM and reward-LSTM) in mortality evaluation, and that our proposed method is significantly better than all the adopted baselines in terms of both total estimated mortality and estimated major mortality. The total estimated mortality is calculated among all the Q-value intervals; the estimated mortality of our framework is 12.3%. Simultaneously, our method improves the performance in conservative Q-value regions. As discussed in Section 5.3, the expected return Q is negatively correlated with the mortality rate with a high correlation. We consider that cases with too high or too low mortality are less important than those in the middle region. So, we divide Q-value units into different regions. The estimated major mortality is calculated in the major Q-value region, which includes most (over 80%) of the decisions in typical cases. This metric is aimed to evaluate the performance on regular decisions, and the exceptional cases with extremely high or low Q-value predictions are not considered. The estimated major mortality of our framework is 9.5%. The Q-value distribution is influenced by reconstruction parameter l, which will be discussed in Sections 5.5.3 and 5.5.4.

Table 3 Performance comparison of different methods

Method	Expected return	Estimated mortality	Major mortality	Jaccard coefficient
BLSTM	–	22.1%	–	0.376
RLSTM	–	21.3%	–	**0.378**
DDQN	14.3	14.6%	12.4%	0.289
Our method	**15.1**	**12.3%**	**9.5%**	0.357

The estimated mortality in expected Q-value's adjacent interval is entitled the estimated major mortality. Jaccard coefficient is calculated between the physician's and policy's decisions. "–" means that the expected return is not used in the non-RL-based approaches for evaluation. Best results are in bold

5.5.2 Training with the temperature parameter

As mentioned in Section 4.2, sample risk uncertainty is evaluated by d-score(s, π) in Eq. (9). The temperature parameter is used to control d-score's volatility. If the d-score strictly distinguishes samples, the model takes good care of ambiguous samples but with less attention to clear samples. We want the d-score to be strict to help the model learn the reliability of samples. Still, strict d-score judgment ignores information of the clear samples. This means that the model would abandon information from a portion of samples. Therefore, we use a strict parameter setting at the beginning of the training and gradually shrink parameter T to 1 at the end of the iteration. Our model's performance converges to a final expected Q-value of 15.1, significantly better than that of the state-of-the-art method DDQN, which is 14.3.

5.5.3 Q-value distribution

The transfer module turns high-uncertainty samples into a projection on the low-uncertainty domain, which keeps the final policy away from high-risk decisions. Fig. 5a is produced by estimating the mortality-expected return function in a similar way to that in Weng et al. (2017). The Q-values are provided by the model for each observed state, and mortalities are calculated by the final survival status of the corresponding Q-value. Fig. 5b is produced by simply calculating the distribution of Q-values for each observed state representation predicted by the model. As shown in Fig. 5, if the reconstruction parameter l is large, the Q-value distribution tends to be more conservative. Otherwise, it tends to be more erratic. The spike in Fig. 5a is caused mainly by the transfer module, which makes the model make more conservative decisions around the overall expected return, thus sacrificing the performance in low- and high-confidence regions. In our experiments, the model performs best in the metric of the total estimated mortality when l=0.5. The influence of different choices of l on the final performance of estimated mortality is shown in Table 4.

5.5.4 Choice of l

Fig. 5b shows the influence of l on the estimated mortality in different Q-value predictions. Over 80% of policy prediction Q-values range from 10 to 20.

With a large reconstruction parameter l, Q-values are controlled not to make high-risk predictions, which ensures conservative Q-value predictions to have desirable mortality. Still, small Q-value predictions would be less stable. We use grid search to find the ideal parameter of $l = 0.5$ and have the lowest estimate mortality of 12.3% around an expected Q-value of 15 (which is obtained by model training). As shown in Fig. 5b, a large reconstruction parameter l controls the model to make most of the decisions around the expected value and sacrifice the performance on other regions.

5.5.5 Performance trade-off

The trade-off between efficiency and conservativeness provides flexibility for different scenarios. In a clinical environment that has accurate measurements with a mature and stable treatment process, the model finds it easy to make clear prescriptions. It is efficient for adjusting the model to maintain a desirable accuracy on high-confidence regions; there would be fewer cases in low-confidence regions. From another perspective, if the clinical environment is complicated and messy, measurements and observations could be ambiguous for a model to predict a high-confidence prescription. Under this circumstance, it is important to maintain the accuracy of the most prescription-confident regions around the expected value. Conservative models leave more space for doctor intervention for detailed treatment plans. This brings more randomness for policy exploration and could be seen as a trade-off between exploitation and exploration. Fig. 6 shows the performance in continuous Q-value intervals. An efficient parameter setting improves the performance on high Q-value regions over the expected Q-value of 15 while trading off the performance in low-confidence regions. A conservative parameter setting maintains the performance around the expected Q-value of 15 and low-confidence regions with most of their decisions.

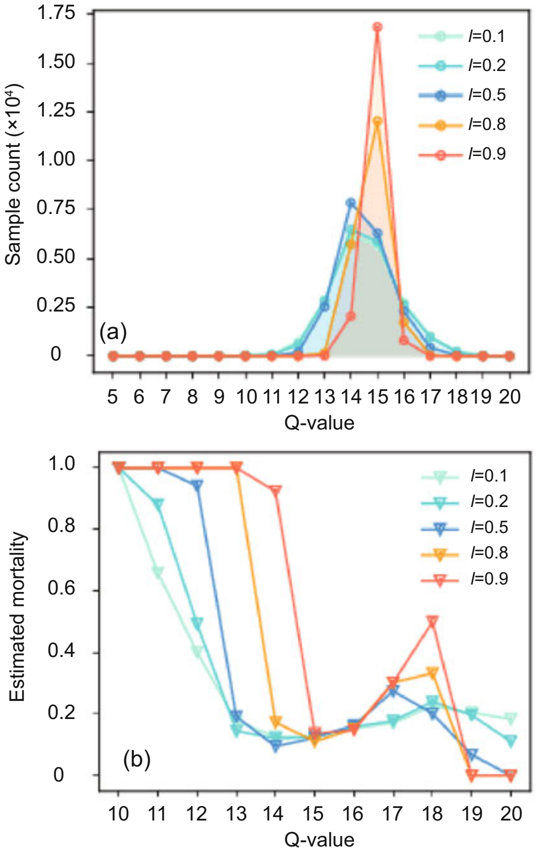

Fig. 5 Expected Q-value distribution under different reconstruction parameter l settings (a) and estimated mortality in different Q-value intervals under the setting of reconstruction for the transfer module (b)

In (a), l is the reconstruction parameter. In (b), l stands for the portion of transferred state feature vectors which are projected into the ideal domain in the reconstruction process. References to color refer to the online version of this figure

Table 4 Performance with different reconstruction parameter l settings used in the transfer module $M(\cdot)$

l	Estimated mortality (%)	Expected return
0.1	12.5	15.00
0.2	12.6	15.01
0.5	**12.3**	14.98
0.8	14.3	15.28
0.9	14.7	**15.41**

Best results are in bold

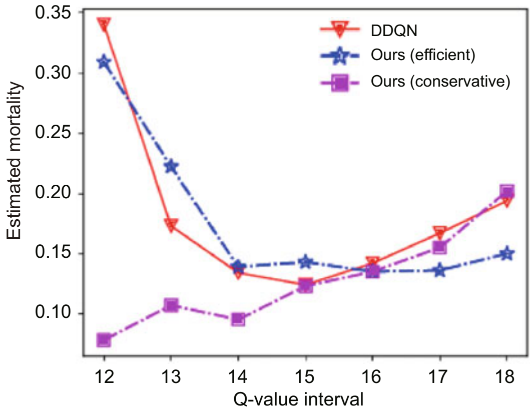

Fig. 6 Estimated mortality distribution in DDQN prediction and our model with a conservative setting $l = 0.3$ and an efficient setting $l = 0.7$

6 Conclusions

In this study, we have proposed a self-supervised RL method with privileged learning to tackle the treatment recommendation problem. The proposed method provides flexibility on the trade-off between efficiency and accuracy. The transfer module, with the evaluation of high-risk uncertainty, helps the model better learn the state representation of the patient's physiological features. The model uses reconstruction parameter l to switch between a conservative policy and an efficient policy. On one hand, the conservative policy centralizes most predictions around the expected Q-value and retains the ensemble mortality, and predictions are more likely to be regarded as guiding suggestions. On the other hand, the efficient policy provides exact treatment prescriptions on easy-handing cases while sacrificing accuracy on ambiguous ones, which then can be left to doctors.

Contributors

Jian PU designed the research. Sihan ZHU processed the data and drafted the manuscript. Jian PU helped organize the manuscript. Sihan ZHU and Jian PU revised and finalized the paper.

Compliance with ethics guidelines

Sihan ZHU and Jian PU declare that they have no conflict of interest.

Data usage notes

The data that supports the findings of this paper employs the MIMIC-III data, which is openly available at https://physionet.org/content/mimiciii/1.4/. The authors of this paper declare that they have signed a data use agreement, which outlines the data usage and security standards and prohibits any effort to identify the patients in MIMIC.

References

Almirall D, Compton SN, Gunlicks-Stoessel M, et al., 2012. Designing a pilot sequential multiple assignment randomized trial for developing an adaptive treatment strategy. *Stat Med*, 31(17):1887-1902. https://doi.org/10.1002/sim.4512

Asiain E, Clempner JB, Poznyak AS, 2018. A reinforcement learning approach for solving the mean variance customer portfolio in partially observable models. *Int J Artif Intell Tools*, 27(8):1850034. https://doi.org/10.1142/S0218213018500343

Bajor JM, Lasko TA, 2017. Predicting medications from diagnostic codes with recurrent neural networks. Int Conf on Learning Representations, p.1-19.

Chen JG, Li KL, Rong HG, et al., 2018. A disease diagnosis and treatment recommendation system based on big data mining and cloud computing. *Inform Sci*, 435:124-149. https://doi.org/10.1016/j.ins.2018.01.001

Chen Z, Marple K, Salazar E, et al., 2016. A physician advisory system for chronic heart failure management based on knowledge patterns. *Theory Pract Log Progr*, 16(5-6):604-618. https://doi.org/10.1017/S1471068416000429

Futoma J, Hariharan S, Heller KA, et al., 2017. An improved multi-output Gaussian process RNN with real-time validation for early sepsis detection. Proc 2nd Machine Learning for Healthcare Conf, p.243-254.

Gidaris S, Singh P, Komodakis N, 2018. Unsupervised representation learning by predicting image rotations. Int Conf on Learning Representations, p.1-16.

Gunlicks-Stoessel M, Mufson L, Westervelt A, et al., 2016. A pilot smart for developing an adaptive treatment strategy for adolescent depression. *J Clin Child Adolesc Psychol*, 45(4):480-494. https://doi.org/10.1080/15374416.2015.1015133

Hendrycks D, Mazeika M, Kadavath S, et al., 2019. Using self-supervised learning can improve model robustness and uncertainty. Proc 33rd Conf on Neural Information Processing Systems, p.1-13.

Hinton G, Vinyals O, Dean J, 2015. Distilling the knowledge in a neural network. https://arxiv.org/abs/1503.02531

Jiang N, Li LH, 2016. Doubly robust off-policy value evaluation for reinforcement learning. Proc 33rd Int Conf on Machine Learning, p.652-661.

Johnson AEW, Pollard TJ, Shen L, et al., 2016. MIMIC-III, a freely accessible critical care database. *Sci Data*, 3:160035. https://doi.org/10.1038/sdata.2016.35

Kaelbling LP, Littman ML, Moore AW, 1995. An introduction to reinforcement learning. In: Steels L (Ed.), The Biology and Technology of Intelligent Autonomous Agents. Springer, Berlin, p.90-127. https://doi.org/10.1007/978-3-642-79629-6_5

Katzman JL, Shaham U, Cloninger A, et al., 2018. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Med Res Method*, 18(1):24. https://doi.org/10.1186/s12874-018-0482-1

Kingma DP, Welling M, 2014. Auto-encoding variational Bayes. Int Conf on Learning Representations Ithacap, p.1-14.

Kingma DP, Salimans T, Jozefowicz R, et al., 2016. Improved variational inference with inverse autoregressive flow. Proc 30th Int Conf on Neural Information Processing Systems, p.4743-4751.

Komorowski M, Celi LA, Badawi O, et al., 2018. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nat Med*, 24(11):1716-1720. https://doi.org/10.1038/s41591-018-0213-5

Li Y, Zeng JB, Shan SG, et al., 2019. Self-supervised representation learning from videos for facial action unit detection. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.10924-10933. https://doi.org/10.1109/CVPR.2019.01118

Long M, Cao Y, Wang J, et al., 2015. Learning transferable features with deep adaptation networks. Int Conf on Machine Learning, p.97-105.

Lopez-Paz D, Bottou L, Schölkopf B, et al., 2016. Unifying distillation and privileged information. https://arxiv.org/abs/1511.03643

Mnih V, Kavukcuoglu K, Silver D, et al., 2015. Playing Atari with deep reinforcement learning. https://arxiv.org/abs/1312.5602

Nemati S, Ghassemi MM, Clifford GD, 2016. Optimal medication dosing from suboptimal clinical examples: a deep reinforcement learning approach. Proc 38th Annual Int Conf of the IEEE Engineering in Medicine and Biology Society, p.2978-2981. https://doi.org/10.1109/EMBC.2016.7591355

Peng XF, Ding Y, Wihl D, et al., 2018. Improving sepsis treatment strategies by combining deep and kernel-based reinforcement learning. American Medical Informatics Association® Annual Symp, p.887-896.

Raghu A, Komorowski M, Ahmed I, et al., 2017. Deep reinforcement learning for sepsis treatment. Proc 31st Conf on Neural Information Processing Systems, p.1-9.

Raghu A, Komorowski M, Singh S, 2018. Model-based reinforcement learning for sepsis treatment. https://arxiv.org/abs/1811.09602

Saria S, 2018. Individualized sepsis treatment using reinforcement learning. *Nat Med*, 24(11):1641-1642. https://doi.org/10.1038/s41591-018-0253-x

Shortreed SM, Laber E, Lizotte DJ, et al., 2011. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Mach Learn*, 84(1-2):109-136. https://doi.org/10.1007/s10994-010-5229-0

Singer M, Deutschman CS, Seymour CW, et al., 2016. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA*, 315(8):801-810. https://doi.org/10.1001/jama.2016.0287

Vapnik V, Izmailov R, 2015. Learning using privileged information: similarity control and knowledge transfer. *J Mach Learn Res*, 16(1):2023-2049.

Vondrick C, Pirsiavash H, Torralba A, 2016. Anticipating visual representations from unlabeled video. IEEE Conf on Computer Vision and Pattern Recognition, p.98-106. https://doi.org/10.1109/CVPR.2016.18

Wang L, Zhang W, He XF, et al., 2018. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. Proc 24th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining, p.2447-2456. https://doi.org/10.1145/3219819.3219961

Wang ZY, Schaul T, Hessel M, et al., 2016. Dueling network architectures for deep reinforcement learning. Proc 33rd Int Conf on Machine Learning, p.1995-2003.

Weng WH, Gao MW, He Z, et al., 2017. Representation and reinforcement learning for personalized glycemic control in septic patients. Proc 31st Conf on Neural Information Processing Systems, p.1-5.

Yu C, Liu JM, Nemati S, 2019. Reinforcement learning in healthcare: a survey. https://arxiv.org/abs/1908.08796

Zhai XH, Oliver A, Kolesnikov A, et al., 2019. S^{4}L: self-supervised semi-supervised learning. IEEE/CVF Int Conf on Computer Vision, p.1476-1485. https://doi.org/10.1109/ICCV.2019.00156

Zhang YT, Chen R, Tang J, et al., 2017. LEAP: learning to prescribe effective and safe treatment combinations for multimorbidity. Proc 23rd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.1315-1324. https://doi.org/10.1145/3097983.3098109

Zhao SJ, Song JM, Ermon S, 2017. InfoVAE: information maximizing variational autoencoders. https://arxiv.org/abs/1706.02262