Workshops at 18th International Conference on Intelligent Environments (IE2022)
H.H. Alvarez Valera and M. Luštrek (Eds.)

doi:10.3233/AISE220052

Inverse Reinforcement Learning Based Approach for Investigating Optimal Dynamic Treatment Regime

Syed Ihtesham Hussain Shah [a,b,1], Antonio Coronato [b,c] and Muddasar Naeem [b]
[a] *Dept. of ICT and Engineering, University of Parthenope, Italy*
[b] *Institute for high performance computing and networking (ICAR), National Research Council, Italy*
[c] *Università Telematica Giustino Fortunato*

Abstract. In recent years, the importance of artificial intelligence (AI) and reinforcement learning (RL) has exponentially increased in healthcare and learning Dynamic Treatment Regimes (DTR). These techniques are used to learn and recover the best of the doctor's treatment policies. However, methods based on existing RL approaches are encountered with some limitations e.g. behavior cloning (BC) methods suffer from compounding errors and reinforcement learning (RL) techniques use self-defined reward functions that are either too sparse or need clinical guidance. To tackle the limitations that are associated with RL model, a new technique named Inverse reinforcement learning (IRL) was introduced. In IRL reward function is learned through expert demonstrations. In this paper, we are proposing an IRL approach for finding the true reward function for expert demonstrations. Result shows that with rewards through proposed technique provide fast learning capability to existing RL model as compared to self-defined rewards.

Keywords. Reinforcement learning (RL), Inverse Reinforcement Learning (IRL), Dynamic Treatment Regime (DTR).

Introduction

Statistical research in the field of healthcare are mostly concerns about the comparison of pre-determined treatment Regimes and the estimation of the optimal treatment. The information that is used in comparing or constructing Dynamic Treatment Regime (DTR) are usually from sequentially randomized studies, longitudinal observational studies or from dynamical system models. Data from sequentially randomized studies are used more extensively because of the large number of randomized clinical tests. Adaptive treatment strategy is a set of guidelines for picking up an effective treatment for individual patients [1]. Treatment choices by following a dynamic regime are dependent on individual's history and characteristics, with the goal of optimizing his/her long-term

[1]Corresponding Author: Syed Ihtesham Hussain Shah, Institute for high performance computing and networking (ICAR), National Research Council (CNR), Via Pietro Castellino, 111 Napoli – 80131, Italy. E-mail: ihtesham.shah@icar.cnr.it

clinical outcome [2]. We have considered Sequential Multiple Assignment Randomized Trial (SMART) [3,4] based on the addiction management [5] for our experiment. A hypothetical based design is shown in the figure-1. There are two initial possible treatments i.e. medicine(MED) and psychology (PSY) that are randomly assigned in each trail. After initial treatment participants are divided into two categories responders (Res) or non-responder (NR) that depends on whether they do or do not experience more than two heavy-drinking days during a specific period of time. A non-responder to MED is either switch to one of the two subsequent treatment options i.e. PSY or an augmentation (PSY + MED). In same way non-responder to PSY is randomized into either MED or an augmentation of PSY with MED (MED + PSY). On the other hand responders (Res) to the initial treatment are observed through telephone monitoring (TM) for an additional time period.

On the other hand, Reinforcement Learning (RL) algorithm is a decision making framework that specify a high-level objective function and learn a policy that satisfies these objectives [6]. A policy in RL is similar to this treatment regime [7]. Most of the time researchers have supposed to specify reward function manually to infer optimal DTR [8,9]. However, the reward function chosen in such a way are extremely scattered which create the problems in recognition of the optimal actions [10]. Clinically guided reward functions may overcome this issue but problem with this technique is that it needs the information of an expert and can not be used into different domains [11].

Inverse Reinforcement Learning (IRL) technique [12] can mitigate this problem and avoid a manual specification of the reward function. True reward function is necessary for the reproduction of the expert's demonstrations [13]. In IRL the reward function is derived from the demonstration of the expert's behavior [14] as represented in section 3.

Rest of the paper is organised as follows: Related work in emerging and existing fields are deliberated in section-1. Section-2 comprise a quick review of background and problem formulation, where basics about the Markov decision process (MDP) is discussed. Proposed approach and system model is detailed in section-3. This section describes the problem of learning the reward function not explicitly but through observing an expert demonstration. Discussion about experiment model, data description and result are stated in section-4. We summarize the paper in Section-5 by giving a conclusion followed by the future directions.

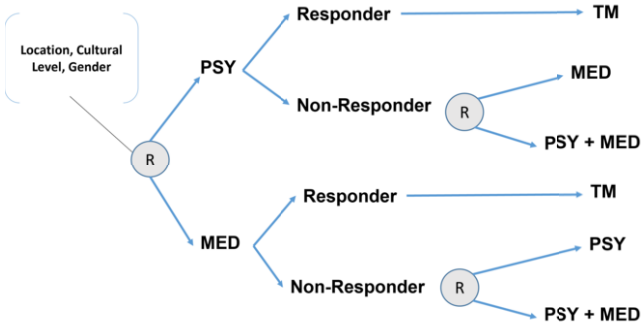

Figure 1. Hypothetical sequential multiple assignment randomized trial (SMART) design schematic for Alcohol addiction management (an R within a circle denotes randomization at a critical decision point)

1. Related Work

Recently, the development in computing technology and the introduction of new machine learning algorithms e.g. reinforcement learning [15], neural network [16] the goal of Artificial Intelligence (AI) has become a step closer. AI has important application in diverse fields including:healthcare [17,18], robotics and autonomous control, vision enhancing method for low vision impairments [19], risk management [20,21], communication [22], and Social humanoid robot [23].
Learning from Demonstrations or Imitation learning [24] may usually be divided into three types: IRL,Adversarial Imitation Learning (AIL) and Behavior Cloning (BC). BC [25] is the technique of learning the policy by direct mapping of states to the actions. It can avoid interaction with the environment. However, it introduces the compounding error along the trajectory length when there is a lack of improvement during training[26]. On the other hand, IRL [27] learns the reward function based on the expert demonstrations that models the preference and intention of the demonstrator.
Preference of the Imitation Learning (IL) [28] is to minimize the Jensen-Shannon divergence between expert policy and learned policy. Different techniques has been adopted for this purpose i.e. Gaussian Process (GP) [29] is utilized in a continuous state space to recover both uncertainty and rewards information. Deep GP model [30] has the capability of learning complicated reward structures With limited number of expert demonstrations.
Application of Machine Learning (ML) techniques in healthcare and biomedical [31] [32] field have increased exponentially in recent years. Alternatively, DTRs are known as treatment policies [33] or adaptive treatment strategies [3,34]. DTR [9,35] oversimplified medicine to time-varying treatment in which treatment is tailored to a patient's dynamic-state. Researchers have successfully applied Deep Learning (DL) techniques to Prognostics and Health Management (PHM) system [36]. RL [37,38] and BC are the two techniques which can be preferred to learn DTRs. BC can efficiently recover the expert (doctor's) policies when the Electronic Health Record (EHR) is sufficient and optimal. RL methods, on the other side, directly learns a policy that are based on maximizing the long-term reward of patients [37,10] . However, pre-defined reward function plays an impotent role in learning the optimal policy.

2. Theoretical background

2.1. Markov Decision Process

A Markov Decision Process (Markov Decision Process (MDP)) can be represented as a complex decision making process that satisfies Markov property. It means that the current decision in MDP is based on the current state or action and not on previous ones. An MDP may be adopted to describe the dynamic of an uncertain environment while an Agent interacts with it by performing actions. A MDP is a form of tuple (S,A,T,R,γ) [39]. Essential elements for a MDP are as under:
-$S = s_1, s_2, s_3, ...s_n$ is the set states for the given environment. State transitions depend on the chosen action and the probability transition function.
-$A = a_1, a_2, a_3, ...a_n$ represents the set of all actions that the agent can take in a state at

Table 1. Features of (SMART) design schematic for DTR

Features	*Stages*
Pre-treatment Features	*First Step*
Location, Gender, Cultural Level	PSY , MED
First Response Features	*Second Step*
Participant's Binary Response, Side Effects, and Adherence	TM, MED, PSY, MED+PSY

every time step.
-$T(s_t, a_t, s_(t+1))$ is a transition function. It states the probability of getting into state $s_(t+1)$ by taking action a in state s in time step t.
-$\gamma \in [0,1]$: is a discount factor. It takes the value in between zero and one. A value close to one gives more weight to long term rewards and a value of zero gives more weight to immediate rewards.

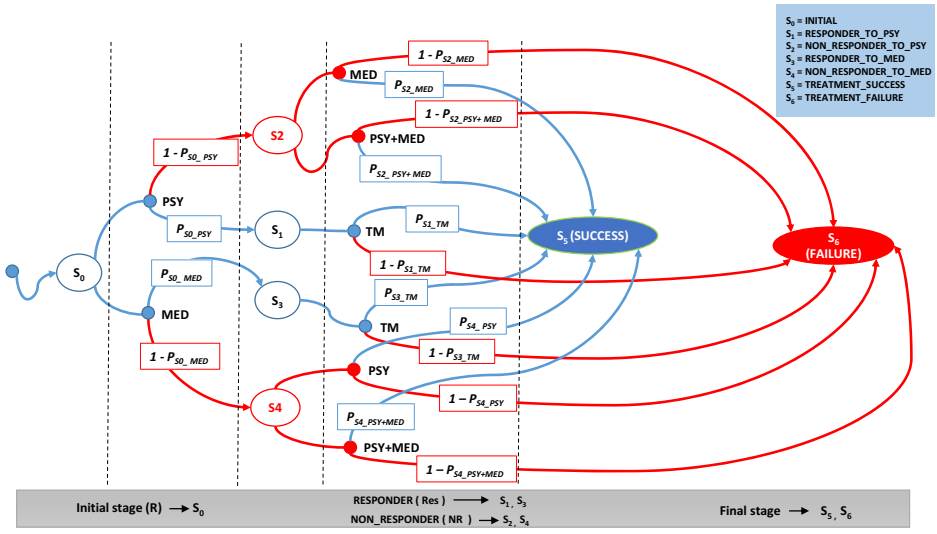

Figure 2. A Markov Decision Process mapping the DTR.

3. Inverse Reinforcement Learning

Let consider an MDP without reward (MDP/R) in this context we have same component as regular MDP except for the reward model. Given some expert's trajectories τ^E as

Table 2. Rewards collected at each state against demonstrations

Pre-treatment Features	Reward Function						
	s_0	s_1	s_2	s_3	s_4	s_5	s_6
M-O-L	-0.07608	0.02531	-0.01483	0.02352	0.02860	0.02662	-0.01314
M-O-M	-0.01065	0.04971	-0.00019	0.03434	-0.06962	0.01591	-0.01949
M-O-H	0.00094	0.01239	-0.04246	-0.01280	0.07577	0.01017	-0.04401
M-D-L	0.00627	0.05076	-0.05029	0.04778	-0.0464	0.00158	-0.00969
M-D-M	0.02857	0.01935	-0.07938	0.02066	-0.00079	0.02684	-0.01525
M-D-H	0.02801	0.05016	-0.01598	-0.06860	-0.00954	0.02766	-0.01169
F-O-L	-0.04651	-0.01826	0.07030	0.00517	0.00423	0.02440	-0.03933
F-O-M	-0.01335	-0.05419	0.03662	0.06278	-0.02876	0.00148	-0.00457
F-O-H	-0.01656	-0.00396	0.07294	-0.04237	-0.00378	0.0242	-0.03047
F-D-L	-0.05002	-0.00568	-0.04073	0.06213	0.04086	0.00047	-0.00703
F-D-M	-0.00041	0.03286	-0.07824	0.01592	0.03171	0.01384	-0.01565
F-D-H	-0.03154	-0.00355	-0.00352	-0.04633	0.06648	0.04019	-0.02174

below:

$$\begin{aligned}\tau^E &= [(s_1^1, a_1^1, s_2^1, a_2^1, .., s_d^1, a_d^1), (s_1^2, a_1^2, s_2^2, a_2^2, .., s_d^2, a_d^2), ...] \\ &= [\tau^1, \tau^2, \tau^3 ...]\end{aligned} \tag{1}$$

An episode τ^i of expert trajectory represents the expert action a at state s for length of d states. Where $s \in [s_0, s_1, s_2, s_3, s_4, s_5, s_6]$ and $a \in [MED, PSY, PSY + MED]$ as stated in figure-2. It is a three stage decision model. Set of all actions that an agent can take are given as $A = ['MED', 'PSY', 'TM', 'PSY + MED']$ and the set of states to which an agent can moves in by taking a specific action are $S = ['s_0', s_1, s_2, s_3, s_4, s_5, s_6]$. Expert trajecto-

Algorithm 1 Proposed algorithm for the reward estimation

Given: Demonstrations τ^π generated by behavior policies π^E, discount factor γ, termination criteria ε,
Initialize: Feature matrix ϕ, number of iteration $n = \infty$,
Randomly pick a policy $\hat{\pi}^0$
set: $w^1 = \mu_E - \hat{\mu}^0$ and $\hat{\mu^{(0)}} = \mu^{(0)}$
Calculate the feature expectation of demonstration $\mu_E = E[\sum_{t=1}^{\infty} \gamma^t \phi(s_t) | \pi_E]$
for $(i = 1; i \leq No.of\ iteration; i^{++})$ **do**
    Calculate $R = (w^{(i)})^T \phi_{(i)}$
    Apply RL to find policy $\hat{\pi}^i$ by using reward R.
    Compute feature expectation: $\mu = E[\sum_{t=1}^{\infty} \gamma^t \phi(s_t) | \hat{\pi}^i]$;
    Compute $\hat{\mu}^{i-1}$ through Equation-8.
    Set $w^i = \mu_E - \hat{\mu}^{(i-1)}$, $t^i = ||\mu_E - \hat{\mu}^{(i-1)}||_2$;
    if $t^i \leq \varepsilon$ **then**
        Break **for**
    end if
end for
Return: $R(s,a) = \sum_i \{w_i \phi_i(s,a)\}$

ries that represents expert action at each state are dependent on the information of some pre-treatment parameters as listed in table-1. It includes the information about gender $\{Male(M), Female(F)\}$, location $\{downtown(d), hill-station(o)\}$ and Cultural-Level $\{high(h), medium(m), low(l)\}$. Actions that are taken at the second stage and also dependent on participant's binary response, adherence and side effects of the policy carried out at 1st step. Expert trajectories can be represented as $\tau^{\pi} = \{[{}'R', {}'MED', {}'Res', {}'TM'], [{}'R', {}'MED', {}'NR', {}'PSY + MED'], [.....]\}$. IRL algorithm is proposed to estimate the true underlying reward function for these expert trajectories.

Value of a policy V is the sum of all the discounted rewards by following that policy. Expectation of the expert value function $V^E(s)$ is defined as:

$$E[V^E(s)] = E[\sum_{t=1}^{\infty} {}^t R(s_t)|\pi_E] \tag{2}$$

$$= E[\sum_{t=1}^{\infty} {}^t w.\phi(s_t)|\pi_E] \tag{3}$$

$$= w\,.\,E[\sum_{t=1}^{\infty} {}^t \phi(s_t)|\pi_E] \tag{4}$$

$$= w\,.\,\mu_{\pi} \tag{5}$$

Where ϕ represents k fixed, known and bounded basis or feature function $\phi : S \rightarrow [0,1]^k$ and weights $w_i \varepsilon R$.

On the other hand, some policies are generated randomly $\{\hat{\pi}^1, \ldots, \hat{\pi}^d\}$ and by mixing them according to mixer weight λ_i we can generate a new policy whose feature expectations μ is convex combination of these policies.

$$\mu = \sum_i \lambda_i \mu^{(i)};\ \ \lambda_i \geq 0, \sum_i \lambda_i = 1. \tag{6}$$

Where λ_i is the probability of picking $\hat{\pi}^i$. One policy (atleast) among randomly generated policies has the performance are as good as expert policy has. It can be found by solving the $\min ||\mu_E - \mu||_2$.

Solution is acceptable if $||\mu_E - \mu||_2 \leq \varepsilon$. It means that, at that point, expert expectation μ_E separated a margin of at most ε from μ. We can reach to that solution after exploring $\mu^1, \mu^2, ...\mu^n$ random feature expectation. In large state space it becomes computationally complex.

Alternatively, according to the Caratheodory's theorem [40]:

$$\mu = \arg \min_{\mu \in C0\{\mu(\hat{\pi}^i)\}_{i=0}^n} ||\mu_E - \mu||_2 \tag{7}$$

where $C0$ denotes convex hull. By doing so we can obtain a set of $k+1$ policies that are equally close to the expert feature expectation and can be calculated as:

$$\hat{\mu}^{i-1} = \hat{\mu}^{i-2} + \frac{(\mu^{i-1} - \hat{\mu}^{i-1})^T (\mu_E - \hat{\mu}^{i-2})}{(\mu^{i-1} - \hat{\mu}^{i-1})^T (\mu^{i-1} - \hat{\mu}^{i-1})} [(\mu^{i-1} - \hat{\mu}^{i-1})] \tag{8}$$

we set

$$w^i = \mu_E - \hat{\mu}^{(i-1)} \tag{9}$$

Reward function might be a linear combination of features ϕ. Where it can be calculated by the dot product of feature function ϕ and weight vector w^i

$$R(s,a) = w_1\phi_1(s,a) + w_2\phi_2(s,a) + \ldots\ldots + w_k\phi_k(s,a)$$

$$R(s,a) = \sum_i \{w_i\phi_i(s,a)\} \tag{10}$$

Value of thresh-hold t defines the termination criteria. Algorithm terminates if its value gets less then some predefined parameter ε.

$$t = ||\mu_E - \hat{\mu}^{(i-1)}||_2 \leq \varepsilon \tag{11}$$

In the beginning, we set the value of $w^1 = \mu_E - \hat{\mu}^0$ where $\hat{\mu^{(0)}} = \mu^{(0)}$.
Algorithm-1 represents the proposed technique. We have some set of expert trajectories that defines the expert policy at each state. We are trying to estimate the underlying reward function for these expert policies. Patient model randomly generates policies (choose action against each state). Feature expectation for these combinations (trajectories) are calculated through the reinforcement learning algorithm. The goal here is to find a policy whose feature expectation are nearly same as the feature expectation of the expert. By finding such a policy we can find the reward function for expert policies.
In this section We have proposed an IRL approach that can be utilized by mixing up estimated policies $\hat{\pi}^{(i)}$ corresponding to their mixture weights λ_i. This new generated policy achieves performance near to that of the expert's policies on the unknown reward function.

4. Experiment

In this section, we have conducted experiments to evaluate the proposed model. Data trajectory are based on observable data set that can be represented as $(O_1, A_1, O_2, A_2, O_3)$. Where O_1 is pre-treatment information, O_2 is intermediate outcomes and O_3 represents the final outcomes information. $A1$ and $A2$ are the randomized treatment actions. In an addiction management study, O_1 may include gender of patient, location, addiction severity and comorbid conditions. O_2 may include the participant's binary response status, side effects and adherence to the initial treatment. Whereas, O_3 may be the number of non-heavy-drinking days for an under-observed period of time.

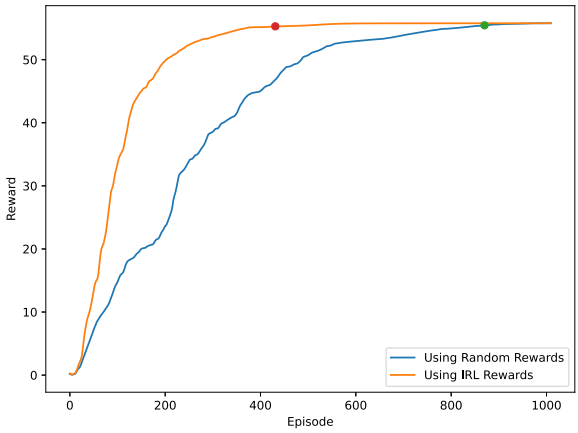

Figure 3. RL model (Q-Learning) learns the DTR environment by adopting IRL rewards and random rewards

4.1. Results

We have tested proposed IRL algorithm-1 to recover under-laying reward function for some generated expert trajectories by setting up a value of 0.7 for discount factor γ and 0.1 for threshhold ε. Rewards at each state calculated for different combination of pre-treatment parameters as represented in table-2. These rewards portray the preference of expert at each state. Another experiment is conducted to see how good these rewards are. For this purpose we tested RL model with learned rewards (collected through proposed approach) and random rewards (usually adopted by existing methods). A very famous RL model (Q-learning) is used for this experiment as shown in figure3. The comparison on both the rewards is presented here. Learning curves show how quickly RL model learns the DTR environment by adopting each of the rewards. A fast learning curve with "IRL rewards" can be seen compared with "random rewards". Almost 431 episodes were needed to learn the environment With IRL rewards. While, in other case it needed almost 870 episodes.

5. Conclusion

In this paper, we proposed IRL algorithm to learn the rewards and hence optimal policies for dynamic treatment regimes. It is shown that finding out new policy by mixing up all the estimated policies corresponding to their mixture weights is a quick and easy way to reach out to an optimal policy. Results shows that existing RL models learns the environment more quickly with the rewards obtained through proposed technique as compared to randomly defined rewards.
It has been seen that IRL methods only consider the positive trajectories (e.g. cured or survived patients) and learn rewards to recover these trajectories. The information in the negative trajectories (e.g. deceased patients) has been largely ignored, which could potentially help the learned policy to avoid repeating mistakes. One can use both pos-

itive and negative trajectories to deduce the best practices and avoid negative practices for dynamic treatment regimes. It may provides better dynamic treatment regimes and improves the likelihood of patient survival with the information from both positive and negative trajectories.

List of Acronyms

DL Deep Learning
MDP Markov Decision Process
RL Reinforcement Learning
DL Deep Learning
DTR Dynamic Treatment Regime
IRL Inverse Reinforcement Learning
IL Imitation Learning
ML Machine Learning
BC Behavior Cloning
AIL Adversarial Imitation Learning
GP Gaussian Process
SMART Sequential Multiple Assignment Randomized Trial

References

[1] X. Chen, S. Li, H. Li, S. Jiang, Y. Qi, and L. Song, "Generative adversarial user model for reinforcement learning based recommendation system," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1052–1061.
[2] E. H. Wagner, B. T. Austin, C. Davis, M. Hindmarsh, J. Schaefer, and A. Bonomi, "Improving chronic illness care: translating evidence into action," *Health affairs*, vol. 20, no. 6, pp. 64–78, 2001.
[3] P. W. Lavori and R. Dawson, "A design for testing clinical strategies: biased adaptive within-subject randomization," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 163, no. 1, pp. 29–38, 2000.
[4] "Reinforcement learning for intelligent healthcare applications: A survey," *Artificial Intelligence in Medicine*, vol. 109, p. 101964, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S093336572031229X
[5] B. Chakraborty and S. A. Murphy, "Dynamic treatment regimes," *Annual review of statistics and its application*, vol. 1, pp. 447–464, 2014.
[6] M. Naeem, S. T. H. Rizvi, and A. Coronato, "A gentle introduction to reinforcement learning and its application in different fields," *IEEE Access*, vol. 8, pp. 209 320–209 344, 2020.
[7] J. D. Co-Reyes, Y. Liu, A. Gupta, B. Eysenbach, P. Abbeel, and S. Levine, "Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings," *arXiv preprint arXiv:1806.02813*, 2018.
[8] M. K. Bothe, L. Dickens, K. Reichel, A. Tellmann, B. Ellger, M. Westphal, and A. A. Faisal, "The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas," *Expert review of medical devices*, vol. 10, no. 5, pp. 661–673, 2013.
[9] S. A. Murphy, "Optimal dynamic treatment regimes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 65, no. 2, pp. 331–355, 2003.
[10] A. Raghu, M. Komorowski, L. A. Celi, P. Szolovits, and M. Ghassemi, "Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach," *arXiv preprint arXiv:1705.08422*, 2017.
[11] A. Raghu, M. Komorowski, I. Ahmed, L. Celi, P. Szolovits, and M. Ghassemi, "Deep reinforcement learning for sepsis treatment," *arXiv preprint arXiv:1711.09602*, 2017.

[12] S. Russell, "Learning agents for uncertain environments," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 101–103.
[13] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.
[14] S. Zhifei and E. M. Joo, "A survey of inverse reinforcement learning techniques," vol. 5, no. 3, pp. 293–311, 2012.
[15] M. Naeem, S. T. H. Rizvi, and A. Coronato, "A gentle introduction to reinforcement learning and its application in different fields," *IEEE Access*, 2020.
[16] M. Naeem, G. Paragiola, A. Coronato, and G. De Pietro, "A cnn based monitoring system to minimize medication errors during treatment process at home," in *Proceedings of the 3rd International Conference on Applications of Intelligent Systems*, 2020, pp. 1–5.
[17] M. Naeem, G. Paragliola, and A. Coronato, "A reinforcement learning and deep learning based intelligent system for the support of impaired patients in home treatment," *Expert Systems with Applications*, p. 114285, 2020.
[18] M. Naeem and A. Coronato, "An ai-empowered home-infrastructure to minimize medication errors," *Journal of Sensor and Actuator Networks*, vol. 11, no. 1, p. 13, 2022.
[19] C. Lodato and P. Ribino, "A novel vision-enhancing technology for low-vision impairments," *Journal of medical systems*, vol. 42, no. 12, pp. 1–13, 2018.
[20] G. Paragliola and M. Naeem, "Risk management for nuclear medical department using reinforcement learning algorithms," *Journal of Reliable Intelligent Environments*, vol. 5, no. 2, pp. 105–113, 2019.
[21] G. Paragliola, A. Coronato, M. Naeem, and G. De Pietro, "A reinforcement learning-based approach for the risk management of e-health environments: A case study," in *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2018, pp. 711–716.
[22] M. Naeem, G. De Pietro, and A. Coronato, "Application of reinforcement learning and deep learning in multiple-input and multiple-output (mimo) systems," *Sensors*, vol. 22, no. 1, p. 309, 2021.
[23] M. Bonomolo, P. Ribino, and G. Vitale, "Explainable post-occupancy evaluation using a humanoid robot," *Applied Sciences*, vol. 10, no. 21, p. 7906, 2020.
[24] H. Burgsteiner, "Imitation learning with spiking neural networks and real-world devices," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 7, pp. 741–752, 2006.
[25] D. A. Pomerleau, "Efficient training of artificial neural networks for autonomous navigation," *Neural computation*, vol. 3, no. 1, pp. 88–97, 1991.
[26] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 627–635.
[27] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.
[28] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in neural information processing systems*, 2016, pp. 4565–4573.
[29] Z.-J. Jin, H. Qian, and M.-L. Zhu, "Gaussian processes in inverse reinforcement learning," in *2010 International Conference on Machine Learning and Cybernetics*, vol. 1. IEEE, 2010, pp. 225–230.
[30] M. Jin, A. Damianou, P. Abbeel, and C. Spanos, "Inverse reinforcement learning via deep gaussian process," *arXiv preprint arXiv:1512.08065*, 2015.
[31] S. Deepika and T. Geetha, "Pattern-based bootstrapping framework for biomedical relation extraction," *Engineering Applications of Artificial Intelligence*, vol. 99, p. 104130, 2021.
[32] M. P. Fanti, S. Mininel, W. Ukovich, and F. Vatta, "Modelling alarm management workflow in healthcare according to ihe framework by coloured petri nets," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 4, pp. 728–733, 2012.
[33] J. K. Lunceford, M. Davidian, and A. A. Tsiatis, "Estimation of survival distributions of treatment policies in two-stage randomization designs in clinical trials," *Biometrics*, vol. 58, no. 1, pp. 48–57, 2002.
[34] A. Oetting, J. Levy, R. Weiss, and S. Murphy, "Statistical methodology for a smart design in the development of adaptive treatment strategies," *Causality and psychopathology: Finding the determinants of disorders and their cures*, vol. 8, pp. 179–205, 2011.
[35] J. M. Robins, "Optimal structural nested models for optimal sequential decisions," in *Proceedings of the second seattle Symposium in Biostatistics*. Springer, 2004, pp. 189–326.
[36] O. Fink, Q. Wang, M. Svensén, P. Dersin, W.-J. Lee, and M. Ducoffe, "Potential, challenges and future

directions for deep learning in prognostics and health management applications," *Engineering Applications of Artificial Intelligence*, vol. 92, p. 103678, 2020.

[37] L. Wang, W. Zhang, X. He, and H. Zha, "Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2447–2456.

[38] Y. Zhang, R. Chen, J. Tang, W. F. Stewart, and J. Sun, "Leap: Learning to prescribe effective and safe treatment combinations for multimorbidity," in *proceedings of the 23rd ACM SIGKDD international conference on knowledge Discovery and data Mining*, 2017, pp. 1315–1324.

[39] M. L. Puterman, "Markov decision processes: discrete stochastic dynamic programming," 2014.

[40] K. Hrbacek, "Nonstandard set theory," *The American Mathematical Monthly*, vol. 86, no. 8, pp. 659–677, 1979.