Hindawi
Scientific Programming
Volume 2020, Article ID 7627290, 13 pages
https://doi.org/10.1155/2020/7627290

Research Article

Optimal Policy Learning for Disease Prevention Using Reinforcement Learning

Zahid Alam Khan,[1] Zhengyong Feng,[1] M. Irfan Uddin,[2] Noor Mast,[2] Syed Atif Ali Shah,[3,4] Muhammad Imtiaz,[5] Mahmoud Ahmad Al-Khasawneh,[4] and Marwan Mahmoud[6]

[1]*China West Normal University, Nanchong, China*
[2]*Institute of Computing, Kohat University of Science and Technology, Kohat, Pakistan*
[3]*Faculty of Engineering and Information Technology, Northern University, Nowshera, Pakistan*
[4]*Faculty of Computer and Information Technology, Al-Madinah International University, Kuala Lumpur, Malaysia*
[5]*Faculty of Computer Science, University of Swabi, Swabi, Pakistan*
[6]*King Abdulaziz University, Jeddah, Saudi Arabia*

Correspondence should be addressed to M. Irfan Uddin; irfanuddin@kust.edu.pk

Received 18 February 2020; Accepted 1 October 2020; Published 28 November 2020

Academic Editor: Shaukat Ali

Diseases can have a huge impact on the quality of life of the human population. Humans have always been in the quest to find strategies to avoid diseases that are life-threatening or affect the quality of life of humans. Effective use of resources available to human to control different diseases has always been critical. Researchers are recently more interested to find AI-based solutions to control the human population from diseases due to the overwhelming popularity of deep learning. There are many supervised techniques that have always been applied for disease diagnosis. However, the main problem of supervised based solutions is the availability of data, which is not always possible or not always complete. For instance, we do not have enough data that shows the different states of humans and different states of environments, and how all different actions taken by humans or viruses have ultimately resulted in a disease that eventually takes the lives of humans. Therefore, there is a need to find unsupervised based solutions or some techniques that do not have a dependency on the underlying dataset. In this paper, we have explored the reinforcement learning approach. We have tried different reinforcement learning algorithms to research different solutions for the prevention of diseases in the simulation of the human population. We have explored different techniques for controlling the transmission of diseases and its effects on health in the human population simulated in an environment. Our algorithms have found out policies that are best for the human population to protect themselves from the transmission and infection of malaria. The paper concludes that deep learning-based algorithms such as Deep Deterministic Policy Gradient (DDPG) have outperformed traditional algorithms such as Q-Learning or SARSA.

1. Introduction

Different types of diseases such as malaria, flu, dengue, and HIV have a huge impact on the quality of life of the human population [1–3]. If we consider malaria only, then according to the World Health Organization's report, approximately 3.2 billion people are infected with malaria. As per their report, in 2016 and 2017, there were 217 and 219 million malaria cases reported, which shows an increase in malaria cases in recent years [4]. Therefore, effective use of resources to get malaria under control has been critical. Insecticide-Treated Nets (ITNs) are the primary method of malaria prevention [5] because there is a type of mosquito called the anopheles mosquito; it bites after 9 p.m. When a mosquito sets on the net, it dies due to the insecticide, which disrupts the reproductive cycle. In addition to ITNs, the

other malaria preventive policies include Indoor Residual Spraying (IRS) [6], larvicide [7] in bodies of water, and malaria vaccination [8–11].

Machine Learning algorithms are applied in different domains and have made tremendous progress [12] where healthcare sector is particularly influenced by machine learning [13–15] in the past few years. These machine learning algorithms are focusing on the diagnosis of diseases [16] or forecasting future results [17], but the treatment of diseases is not explored [18]. It is a very important step to diagnose a disease and is considered as an important step to treat diseases, and machine learning techniques can support healthcare professionals in the treatment to some extent, but it has been a challenging problem to find the best policy to treat patients for medical professionals [19]. Recently, much popularity is gained by reinforcement learning (RL) [20] in video games [21–23], where good and bad actions are learned by the agent through interactions with the environment and the response of the environment. In the context of video games, RL has performed very well, but limited progress has been made in real-world domains like health care. In video games such as AlphaGo and StarCraft, the agent plays a large number of actions in the environment and learns the optimal policy. However, in the context of health care, it is considered unethical to use humans to train RL algorithms and not to mention that this process would be costly and takes years to complete. We are not able to observe everything happening the body of a person. We can measure blood pressure, temperature, and some other measurements at different intervals of time, but these measurements do not represent the complete state of a patient. Similarly, the data collected in health care about patients may exist for one time and may not exist for others. For example, chest X-rays that are used in the treatment of pneumonia [24] are collected before a person is infected and after the person is cured, but the RL model has to know all the estimates of the states the patient goes through. It is very challenging in health care where there are many unknown facts about patients at all time steps.

Reward function is one of the most important functions in RL, and it is challenging in many real-world applications to find a good reward function. In health care, it is even more challenging to search for the reward function that keeps balance between short-term success and overall long-term improvements. For example, in case of sepsis [25], improvements in blood pressure at different durations of time may not cause improvement in the overall success. Similarly, having only a single high reward at the end of an episode (i.e., survived or died) demonstrates that a long route is followed without different intermediary rewards [26, 27]. It is also difficult to know what actions result in reward and what actions result in penalty. All the major breakthroughs are possible by using simulated data in deep RL that is equal to many actual years [28]. When data are generated through simulators, it is not a problem, but in case of health care, it is not possible to generate simulated data for the treatment of different diseases. Generally, the data are very scarce to start with training supervised learning, and the data that exist take efforts to annotate to be used for supervised learning. Furthermore, hospitals are not willing to share data of patients mainly because of privacy reasons. All these facts further make the application of deep RL to health care challenging.

By nature, the health care data is nonstationary and dynamic [29]. For example, it is possible that patients' symptoms are stored at different intervals of time and maybe different records are stored for different patients. Over time, the objectives of treatments may also change. In literature, different studies [30–32] are focused on reducing the overall mortality. When the condition of a person improves, the focus shifts to a different objective such as the duration of the virus staying in the body. Similarly, viruses or infections may change much more rapidly and may evolve in different dynamics [33–35] that are most probably not observed in the training data used for supervised or semisupervised learning algorithms. Decision-making in medical diagnosis is inherently sequential [36, 37]. It means that a patient visits a health care centre for the treatment of a disease. The doctor, based on previous experiences, decides a treatment to be followed. Later, when the patient returns to the same doctor, the treatment that was previously suggested by the doctor decides the current state of the patient and also helps the doctor in which decision needs to be taken next. In the existing state-of-the-art AI strategies of dealing with disease treatment [38, 39], the sequential nature of the decisions is ignored [40]. These AI systems make decisions on the basis of the present state of the patients. The sequential nature of medical treatment can be effectively modelled as Markov Decision Process (MDP) [41–44] and better solved through RL. The RL algorithms will not only consider the instantaneous outcomes of treatment but also the long-term benefits of the patients [45].

An intervention of actions to avoid malaria are systematically explored in this paper. The paper demonstrates a real-world example of reinforcement learning, where simulated humans are trained to learn an effective technique to avoid malaria. In the literature, AI techniques are used for the prediction, diagnosis, and healthcare planning, but this paper takes a different approach by simulating an environment and using simulated humans to use different reinforcement learning techniques to avoid malaria. A combination of interventions is explored to control the transmission of malaria and learn techniques for malaria avoidance.

The paper is organized as follows: the related works are explained in Section 2. The problem of malaria avoidance and the methodology of reinforcement learning are given in Section 3. Experiments are performed, and their results are analysed in Section 4. Concluding remarks of the paper are given in Section 5.

2. Related Work

Recent advancements in machine learning and big data have motivated researchers of different domains to use these algorithms in their problems. Biomedical and health care researchers are getting benefits from these algorithms in early disease recognition, community services, and patients

care. In [46], machine learning and MapReduce algorithms are used to effectively predict different diseases in disease-frequent societies. The paper demonstrated to achieve 94.8% accuracy and convergence speed that is faster than CNN (Convolutional Neural Network) based algorithms. Similarly, deep learning and big data techniques have been used in [47] to predict infectious diseases. The authors have combined Deep Neural Network (DNN) and Long Short-Term Memory (LSTM) and evaluated the performance with Autoregressive Integrated Moving Average (ARIMA) in making the prediction of different diseases one week in the future. Better results have been achieved compared to ARIMA. Automatic diagnosis of malaria enables us to provide reliability in health care services to areas where resources are limited. Machine learning techniques have been tried to investigate the process of automating malaria detection. In [48], malaria classification is performed using CNN. Similarly, in [49], CNN has been used to detect malaria classification and has demonstrated promising accuracy. Deep reinforcement learning (DRL) has recently attained remarkable success, notably in complex games like Atari, Go, and Chess. These achievements are mainly possible because of the powerful function approximation with the help of DNN. DRL has been proved as an effective method in the medical context. Several applications of RL have been found in the context of medicine. For instance, RL methods have been used to develop strategies of treatment for epilepsy [50] and lung cancer [51]. Authors have used the sepsis dataset which is a subset of the MIMIC-III dataset [25]. An action space consisting of vasopressors and IV fluid is selected. Each drug of varying amount is grouped in four bins. Double Deep Q-Network is used for the evaluation. SOFA score which is used for measurements of organ failure is used for the reward function. U-curve is used for evaluation. The mortality rate is used as a function of dosage of policy prescription versus the policy that is actually followed.

In [19], DRL is used to develop a framework that predicts an optimal strategy to deal with Dynamic Treatment Regimes using medical data. The paper has claimed that their RL model is more flexible and adaptive in high dimensional action and state spaces compared to other RL based approaches. The framework models real-world complexity in helping doctors and patients to make a personalized decision in making treatment choices and disease progression. The framework combines supervised learning and DRL using DNN. The dataset is taken from the database of the Centre for International Bone Marrow Transplant Research (CIBMTR) registry. The framework has demonstrated achieving promising accuracy to predict a human doctor's decision and at the same time compute a high reward function. In [52], an RL system is developed that helps diabetes patients to engage in different physical activities. Messages sent to patients were made personalized to patients and the results have demonstrated that participants receiving messages with the RL algorithm increased the number of physical activities and walking speed. A supervised RL with recurrent neural network (SRL-RNN) is combined in a framework to make different treatment recommendations by Wang et al. in [53]. Their results of experiments conducted on MIMIC-3 dataset have demonstrated that the RL based framework can reduce the estimated mortality and at the same time provide promising accuracy to match doctor's prescriptions. In [54], the authors describe a novel technique that can find the optimal policy that can treat patients with chemo using RL. The authors have used Q-Learning, and, for the action space, a mechanism is used to quantify doses for a given time period that an agent can choose from. The cycle of dose is initiated with a frequency as determined by an expert. At the end of each cycle, transition states are compared. The mean reduction in tumour diameter determines the reward function. Simulated clinical trials are used for the evaluation of the algorithm.

In [55], the authors have taken a different approach that uses the RL techniques to encourage healthy habits instead of looking for direct treatment. In [56], the authors focus on sepsis and RL, but a different approach is taken that uses the RL techniques to control glycemic. In [57], the authors have focused on counterfactual inference and domain adversarial Neural Networks. It is a complicated problem to solve the problem of decision-making under uncertainty. Health care practitioners are facing problems under challenging constraints, with limited tools to make data driven decisions. In [58], the authors have solved the problem of finding an optimal malaria policy as a stochastic multiarmed bandit problem and have developed three agent-based strategies to explore the space of policies. A Gaussian Process regression is applied to the finding of each agent, for compression and for stochastic results from simulating the spread of malaria in a fixed population. The policy generated by the simulation is compared with human experts in the field for direct reference. In [59], the authors have exposed subtleties associated with evaluating RL algorithms in health care. The focus is on the observational setting where RL algorithms have proposed a treatment policy and been evaluated based on historical data. A survey in [60] discusses the different applications of reinforcement learning in health care. The paper provides a systematic understanding of theoretical foundations, methods and techniques, challenges, and new insights into emerging directions. A context aware hierarchical RL scheme [61] has been shown to significantly improve the accuracy of symptom checking over traditional systems while reducing the number of inquiries. Another study that introduces basic concepts of RL and how RL could be effectively used in health care is given in [62].

Policy for malaria control using the reinforcement learning algorithm is explained in [63, 64]. The authors have applied the Genetic Algorithms [65], Bayesian Optimization [66], and Q-Learning with sequence breaking to search for optimal policy for a few years. Their experiments demonstrated the best performance by Q-Learning algorithm. A systematic review of agent-based models for malaria transmission is given in [67]. The paper covers an extensive array of topics covering the spectrum of transmission and intervention of malaria. Machine learning algorithms for the prediction of different diseases are studied in [68]. The authors have used Decision Tree and MapReduce algorithms and have claimed to achieve 94.8% accuracy. Machine learning algorithms have been used to automatically

diagnose malaria in [69]. Deep Convolutional Neural Networks have been used for classification. The authors in [70] have discussed safety applications related to AI in those domains where deep reinforcement learning is applied to the control of automatic mobile robots. An investigation of the risk associated with malaria infection to identify those bottlenecks in different malaria elimination techniques is discussed in [71]. Other relevant studies can be found in [72–74].

3. Methodology

Reinforcement learning (RL) [75] is an example of machine learning methods falling between supervised and unsupervised learning, where an agent learns by interacting with the environment. The agent performs certain actions and receives feedback from the environment. This feedback is in the form of negative or positive reward and determines the sequence of good or bad actions to be adapted within a particular situation. As a result, the agent can perform its operation efficiently without any intervention from a human. In other words, RL is a learning method where an agent learns a sequence of actions to eventually increase the reward function. The agent decides which action is the most appropriate and yields a maximum reward. It is possible that an action may not give a positive immediate reward but the long-term reward is also considered. In RL, we have two components, that is, agent and environment as shown in Figure 1. The agent represents the type of RL algorithm, and the environment represents what action returns which reward. The environment is established by sending a state at time t as $S_t \in S$, where S is the representation of the set of possible states to the agent. The action taken by the agent at time t is represented by $A_t \in A\ (S_t)$, where $A\ (S_t)$ is the representation of the set of actions possible to be taken at state S_t. The reward to be received by performing that action is represented as $R_{t+1} \in R$, where R is the set of rewards. After one time-step, the next state S_{t+1} will be sent to the agent by the environment along with reward R_{t+1}. This reward will eventually help the agent increase its knowledge to be used in evaluating its last action. This process of sending state and receiving reward as an outcome by the agent continues until the environment sends the last or terminal state to the agent.

In addition to the agent and environment, there are four components in a RL environment: (i) policy, (ii) reward, (iii) value function, and (iv) model of the environment.

(1) *Policy*. A policy defines the behaviour/reaction of an agent at a particular instance of time. Sometimes, a policy can be described as a simple function or as a lookup table, where a policy may involve a lot of computation, for example, the searching process. The policy is considered as a central part of the RL agent because it alone can describe the reaction of the agent. The policy may be stochastic, to determine possibilities for every action. The policy is represented by π_t, where $\pi_t\ (a \mid s)$ demonstrates the probability of $A_t = a$ if $S_t = s$

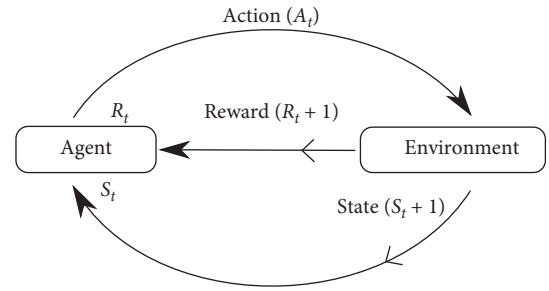

Figure 1: A typical reinforcement learning paradigm.

(2) *Reward*. A reward signal indicates the target of an RL problem. As a result of an action taken by the agent, the environment returns a number, called a reward, at every time step. The objective of the agent is to get most of the total reward over time. Thus, the reward signal identifies that an action is good or bad. The rewards signal determines the action to be taken. If an action returns a low reward, then the policy will be changed to select another action in a similar situation. So generally, a reward signal is the stochastic function of the state and action.

(3) *Value Function*. A reward signal identifies what is good at the current time, while a value function describes what is good in the long run. In almost all RL algorithms, the most important component to be considered is the method to efficiently estimate the values. More precisely, the current value of the earlier state is adjusted to be closer to the value of the later state. This can be done by moving the earlier state's value a fraction toward the value of the later state. Let s denote the state before the move, and s is the state after the Agent Environment moves; then, the update to the estimated value of s, denoted as $V(s)$, can be written as shown in equation (1), where α' is a small positive fraction called the step-size parameter, which influences the rate of learning. $r + \gamma V(s\prime)$ is called Temporal Difference target and is an unbiased estimate for $V(s\prime)$. In equation (1), r represents reward and γ represents the discounting factor. This update rule is an example of a Temporal Difference learning method, called so because its changes are based on a difference, $V(s') - V(s)$, that is, difference between estimates at two different times:

$$V(s) \longleftarrow V(s) + \alpha\left[r + \gamma V\left(s'\right) - V(s)\right]. \tag{1}$$

(4) *Model*. A model allows inferences of the actions in an environment. Suppose a state and action are given; then, it is possible that the model determines the resultant next state and reward. The methods that use the models and planning to solve RL Problems are known as model-based methods. Those techniques which are explicitly trail-and-error learner are called model-free methods.

Let us assume that there are finite states and rewards. Let us consider an environment that may respond at time $t+1$ to the action taken at time t. This response actually depends on everything that happened earlier. The complete probability distribution of the dynamics of the system can be defined in equation (2), for all r, S, and all possible values of the actions in the past represented in the form of action, states, and rewards, that is, S_t, A_t, and R_t. However, due to the Markovian property, we can represent the response of the environment at $t+1$ that depends only on the state and action at time t. The dynamics of the environment can be defined as given in equation (3), for all r, s', S_t, and A_t. It means that a state or an environment has a Markovian property if and only if equations (2) and (3) are equal. The Markovian property is very important in RL, as decisions and values are a function of the current state. These decisions and values can be effective and carry more information when the state representation carries enough information:

$$\begin{aligned}\Pr\{&R_{t+1}=r, S_{t+1}=S'|S_0, A_0, R_1, \ldots,\\ &S_{t-1}, A_{t-1}, R_t, S_t, A_t\},\end{aligned} \tag{2}$$

$$p(s', r|s, a) = \Pr\{R_{t+1}=r, S_{t+1}=s'|S_t, A_t\}. \tag{3}$$

The task of RL that satisfied the Markovian property is known by the name Markov Decision Process (MDP). Given a state s and action a, the computation of probability of next state s' along with reward r is denoted as given in equation (4). The expected value of rewards for the state-action pairs can be computed given in equation (5). The expected rewards for state-action-next-state is given in equation (6):

$$p(s', r|s, a) = \Pr\{S_{t+1}=s', R_{t+1}=r|S_t=s, A_t=a\}, \tag{4}$$

$$\begin{aligned}r(s,a) &= E[R_{t+1}|S_t=s, A_t=a]\\ &= \sum_{r\in R} r \sum_{s'\in S} p(s', r|s, a),\end{aligned} \tag{5}$$

$$\begin{aligned}r(s,a,s') &= E[R_{t+1}|S_t=s, A_t=a, S_{t+1}=s']\\ &= \frac{\sum_{r\in R} r\cdot p(s', r|s, a)}{p(s'|s, a)}.\end{aligned} \tag{6}$$

Value functions, which is a function of states or state-action pairs, are used to estimate the performance of an agent in a given state. This performance is computed in terms of future rewards to be collected. The state value is denoted by $V_\pi(s)$ given a policy π and state s and is computed as shown in equation (7), where E_π [.] represents the expectation of variable when an agent follows a policy π at time step t. Similarly, the action value of a state s following a policy π represented by q_π (s, a) is given in equation (8), where q_π is the function of action-value when π policy is used:

$$\begin{aligned}V_\pi(s) &= E_\pi[G_t|S_t=s]\\ &= E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s\right],\end{aligned} \tag{7}$$

$$\begin{aligned}q_\pi(s,a) &= E_\pi[G_t|S_t=s, A_t=a]\\ &= E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t=s, A_t=a\right].\end{aligned} \tag{8}$$

RL problem is solved by searching for a policy that helps the agent to collect maximum possible rewards over the execution of the simulation. A given policy π is treated as a better policy or equal to another policy π', it the expectation of the π is greater or equal to the expectation of π' for all states. In other words, $\pi \geq \pi'$ if and only if $V_\pi(s) \geq V'_\pi(S)\ \forall\, s \in S$. An optimal policy is the policy that is considered good or equal to all possible policies. Optimal policies are represented by π^*. The same state-value function is shared by optimal policies as V^* and defined as $V^*(S) = \max V_\pi(S)\ \forall\, s \in S$. They also share same optimal action-value function, represented by q^* defined as $q^*(s, a) = \max q_\pi(s, a)\ \forall\, s \in S$ and $a \in A(s)$.

The model-based RL means the simulation of the dynamics of a given environment. The model learns the probability of moving from the current state s_0, taking action a and ending in next state s_1. Given the learning of transition probability, the agent can determine the probability to enter a state given the current state and action. However, model-based algorithms are not practical because the state space and action space grow. On the other side, the model-free algorithms depend on trial-and-error to update its knowledge. Therefore, space is not required to store all combination of states and actions. In this paper, we are using model-free algorithms. Classification of RL algorithms are made based on on-policy and off-policy. When the value is based on the current action a and derived from the current policy, it is known as on-policy. When an action a^* is obtained from a different policy, then it is known as off-policy.

3.1. Q-Learning. A well-known algorithm in RL is Q-Learning developed by Watkins [76]. Its proof of convergence is given by Jaakkola [77]. Q-Learning is a simple technique, and it can compute optimal action value without the involvement of intermediary evaluation of cost and the usage of a model [78]. This algorithm is model-free and is considered as off-policy algorithm, which is derived from Bellman Equation as shown in equation (9), where expectation is given by E and discounting factor is represented by λ. This update equation is shown in Algorithm 1 on line 10. Learning rate is represented by α. The next state's Q value determine the next action a instead of using the current

Input:
States: $S = 1, \ldots, n$
Actions: $A = 1, \ldots, n$
Rewards: R: $S \times A \longrightarrow R$ Transitions: T: $S \times A \longrightarrow S$
$\alpha \in [0, 1]$ and $\gamma \in [0, 1]$
Randomly Initialize $Q(s, a) \forall s \in S, a \in A(s)$
while For every episode do
    Initialize $S \in S$
    Select a from s on the basis of exploration strategy (e.g. ε-greedy)
    while For every step in the episode do
        //Repeat until s is terminal
        Compute π on the basis of Q and strategy of exploration (e.g. $\pi(s) = \text{argmax}_a Q(s, a)$)
        $a \longleftarrow \pi(s)$
        $r \longleftarrow R(s, a)$
        $s \longleftarrow T(s, a)$
        $Q(s', a) \longleftarrow (1 - \alpha).Q(s, a) + \alpha [r + \max'_a Q(s', a')]$
        $s \longleftarrow s$

Algorithm 1: Q-Learning.

policy. The overall objective of the algorithm is to maximize the Q-value:

$$Q^{\pi}(s, a) = E_{s'}\left[r + \lambda Q^{\pi}\left(s', a'\right) | s, a\right]. \tag{9}$$

3.2. SARSA. A similar algorithm to Q-Learning is SARSA [79, 80]. In case of Q-Learning, greedy policy is followed, but in case of SARSA on-policy is followed. SARSA learns Q-value by performing actions using the current policy. Algorithm 2 shows the algorithm of SARSA. Current policy is used to carry out selection of actions.

3.3. Deep Deterministic Policy Gradient. An actor-critic architecture is called Deep Deterministic Policy Gradient (DDPG) [81, 82]. The parameter x is tuned for policy by actor as given in equation (10). Using Temporal Difference error, the policy computed by the action is evaluated by critic as demonstrated in equation (11). The policy decided by the actor is shown by v. The idea of experience replay and separate target network as utilized by Deep Q Network (DQN) [83] is used by DDPG. Algorithm 3 shows the algorithm of DDPG.

$$\pi_{\theta}(s, a) = P[a | s, \theta], \tag{10}$$

$$r_{t+1} + \gamma V^{v}\left(S_{t+1}\right) - V^{v}\left(S_t\right). \tag{11}$$

$$A(s) = \left[a_{\text{ITN}}, a_{\text{IRS}}\right], \quad \text{where } a_{\text{ITN}} \in [0, 1] \text{ and } a_{\text{IRS}} \in [0, 1]. \tag{12}$$

4. Simulation and Discussion

In this section, we present the results of algorithms explained in Section 3 obtained in a simulated human population and see which algorithm performs better to prevent humans from diseases. For the evaluation, we need an environment where we have different states, actions, and agents (representative of human population) looking for the best policy to avoid diseases such as malaria, flu, and HIV. In this section, results are shown for malaria avoidance only, but similar environment with sufficient information can be used for the avoidance of other types of diseases such as flu, HIV, and dengue. An environment where a human, mosquito, and other factors that can influence the transmission of malaria virus to spread to human is shown in Figure 2. The box on the left contains factors relevant to human and the box on the right contains factors pertaining to mosquitoes. Different factors that can influence the disease are shown inside the arrows linking the boxes for humans and mosquitoes. Environment factors and interventions are shown on the top and bottom of the boxes for human and mosquitoes.

The IBM Africa research team has taken steps to control malaria by developing a world-class environment to distribute bed nets and repellents. Their goal is to develop a custom agent that will help identify the best policies for rewards based on the simulation environment. Our work leverages the environment developed by IBM Africa research for reinforcement learning competition on hexagon-ml (https://compete.hexagon-ml.com/practice/rl_competition/38/) where an agent learns the best policy for the control of diseases, that is, malaria. The environment provides stochastic transmission models for malaria and different researchers can evaluate the impact of different malaria control interventions. In the environment, an agent may explore optimal policies to control the spread of the malaria virus. A diagram representing the environment developed by Hexagon-ML for finding the best policy for avoiding malaria is given in Figure 3. The environment contains five years. Every year is a state. At every state, we take different actions in the form of ITN and IRS.

States are represented as $S \in \{1, 2, 3, 4, 5\}$, where each number shows the number of the year. We are trying to solve the problem of making one-shot policy recommendations for the simulation intervention period of 5 years. The main control methods used in different regions are mass-distribution of long-lasting ITNs, IRS with pyrethroids, and the prompt and

Input:
States: $S = 1, \ldots, n$
Actions: $A = 1, \ldots, n$
Rewards: R: $S \times A \longrightarrow R$
Transitions: T: $S \times A \longrightarrow S$
$\alpha \in [0, 1]$ and $\gamma \in [0, 1]$
$\lambda \in [0, 1]$ this shows the trade-off between Temporal Difference and Monte Carlo methods.
Randomly Initialize $Q(s, a) \forall s \in S, a \in A(s)$
while For every episode do
    Randomly initialize $s \in S$
    Initialize e with 0
    Randomly select $(s, a) \in S \times A$
    while For every step in the episode do
        //Repeat until s is terminal
        $r \longleftarrow R(s, a)$
        $s' \longleftarrow T(s, a)$
        Compute π based on Q using exploration strategy (e.g. ε-greedy)
        $a' \longleftarrow \pi(s')$
        $e(s, a) \longleftarrow e(s, a) + 1$
        $\delta \longleftarrow r + \gamma.Q(s', a') - Q(s, a)$
        for $(s', a') \in S \times A$ do
            $Q(s', a') \longleftarrow Q(s', a') + \alpha.\delta.e(s', a')$
            $e(s', a' \longleftarrow \gamma.\lambda.e(s', a'))$
        $s \longleftarrow s'$
        $a \longleftarrow a'$

Algorithm 2: SARSA.

(1) Randomly initialize critic network $Q(s, a|\theta^Q)$ with weight θ^Q
(2) Randomly initialize actor $\mu(s \mid \theta^\mu)$ with weight θ^μ
(3) Initialize target network Q' with weight $\theta^{Q'} \longleftarrow \theta^Q$
(4) Initialize target network μ' with weight $\theta^{\mu'} \longleftarrow \theta^\mu$
(5) Initialize replay buffer R
(6) **while** *For every episode* **do**
(7) Randomly initialize N for exploration
(8) Get initial observation state s_1
(9) **while** *For every step in the episode* **do**
        //Repeat until s is terminal
(10) Section action $a_t = \mu(s_t \mid \theta^\mu) + N_t$ as per the current policy and exploration strategy
(11) Perform action a_t and monitor rewards r_t and new states$_{t+1}$
(12) Store (s_t, a_t, r_t, s_{t+1}) in R
(13) Sample a randomly selected minibatch of N transition (s_i, a_i, r_i, S_{i+1}) from R
(14) $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} \mid \theta^{\mu'} \mid \theta^{Q'}))$
(15) $L = 1/N \sum_i (y_i - Q(s_i, a_i \mid \theta^Q))^2$
  //Update rule for critic to minimize the loss
(16) $\Delta_{\theta\mu} J \approx 1/N \sum_i \Delta_\alpha Q(s, a \mid \theta^Q)|_{s=s_i, a=\mu(S_i)} \Delta_{\theta^\mu} \mu(s \mid \theta^\mu)|_{s_i}$
  //Update rule for actor policy using the sampled policy gradient
(17) $\theta^{Q'} \longleftarrow \gamma\theta^Q + (1 - \gamma)\theta^{Q'}$
  //Update rule for target network
(18) $\theta^{\mu'} \longleftarrow \gamma\theta^\mu + (1 - \gamma)\theta^{\mu'}$

Algorithm 3: Deep Deterministic Policy Gradient.

effective treatment of malaria. Actions, represented by $A(s)$, are performed in the form of ITN and IRS, where the values of ITN and IRS are infinite real numbers between 0 and 1.

The agent trained on a reinforcement learning algorithm will explore a policy space made up of the first two components, that is, ITNs and IRS, which are strategies for direct intervention. The prompt and effective treatment is given by the environment parameters and impacts the rewards. The first component. That is, ITN, is the development of nets, defining the population coverage ($a_{\text{ITN}} \in (0, 1]$). The second

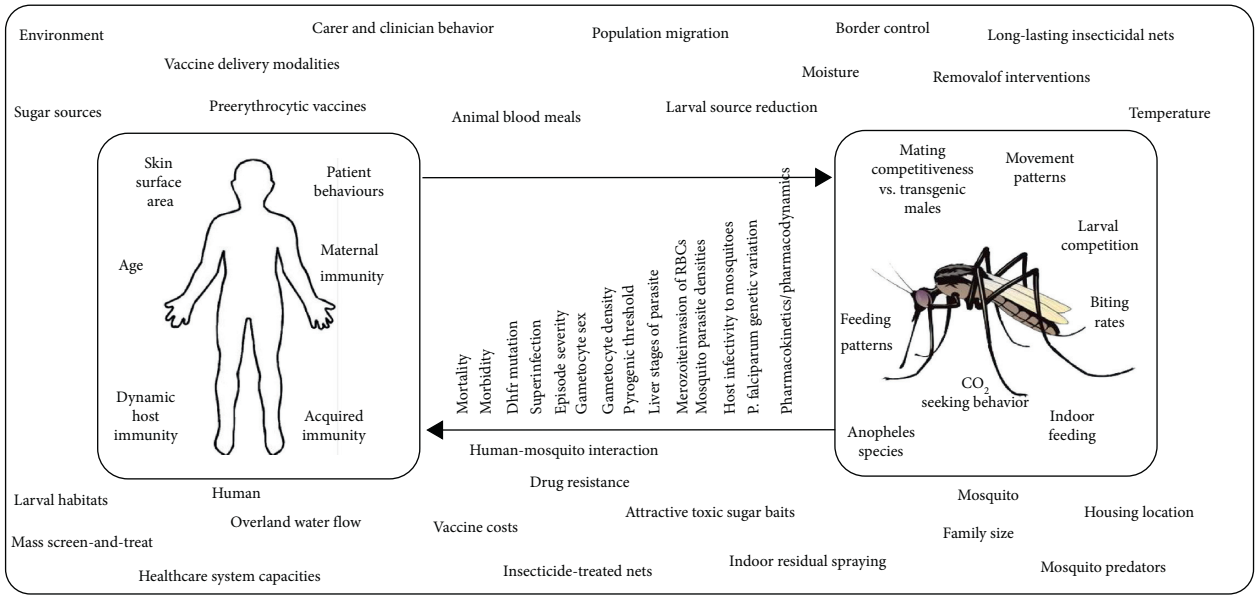

Figure 2: Different factors related to humans, mosquitos, and the environment that influence malaria transmission.

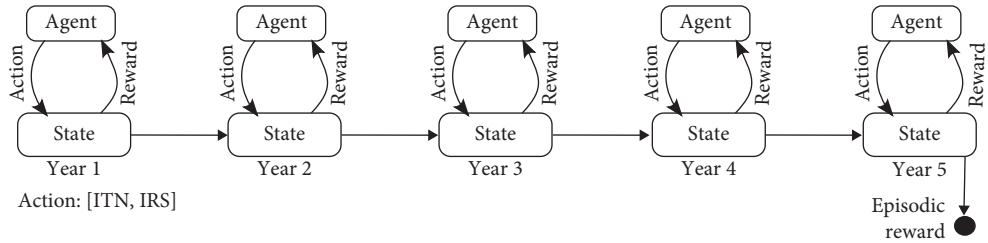

Figure 3: A block diagram showing the environment developed by Hexagon-ML for reinforcement learning algorithms to learn malaria intervention.

component is the use of seasonal spraying, and it defines the proportion of population coverage for this intervention ($a_{\text{IRS}} \in (0, 1]$). The seasonal spraying is performed through alternating the intervention between April and June every year in different regions. The policy decision is framed in a way of the simulated population to be covered by a particular intervention; the space of policy A is designed through $a_i \in A = [a_{\text{ITN}}, a_{\text{IRS}}]$.

Health care organizations should be able to explore all possible set of actions for appropriate malaria interventions within the populations. These policies include a mix of actions, like the distribution of ITNs, IRS, larvicide in water, and vaccination for malaria control. The space of possible policies for the control of malaria is not complete and inefficient for health care experts to explore without an adequate decision support system. The environment in simulation handles the distribution of the interventions in the simulated population. The agent is in charge of the complex actions of targeted interventions, which are not reported previously. Although the action space is finite (i.e., finite number of people in the simulation environment) the space size grows exponentially as more interventions are added. The computation time of simulation will also grow linearly with the number of populations. Therefore, a complex exploration of the entire action space becomes impossible as complexity goes to a real-world equivalent simulation.

The agent learns different rewards during the learning process. The idea of learning is to collect as much reward as possible during the process of execution of the experiment. These rewards are infinite and usually represented by $R_\pi \in (-\infty, +\infty)$, where the policy is represented by π. Every policy is associated with a reward represented by $R_\theta(ai)$ and is a stochastic parameterization of the simulation shown as θ which produces random distribution of parameters for the simulated environment.

The environment is executed for 100 episodes, and rewards are collected. An episode consists of five consecutive years. The rewards collected by different algorithms are demonstrated in Figure 4. The random selection algorithm when there is no learning for 100 episodes is given in Figure 4(a). In random policy learning, every time one episode is finished, the environment is initiated with different random states and different policy is tried at random to go from one state to another to collect rewards. In this algorithm, no learning is involved, and this experiment is performed only

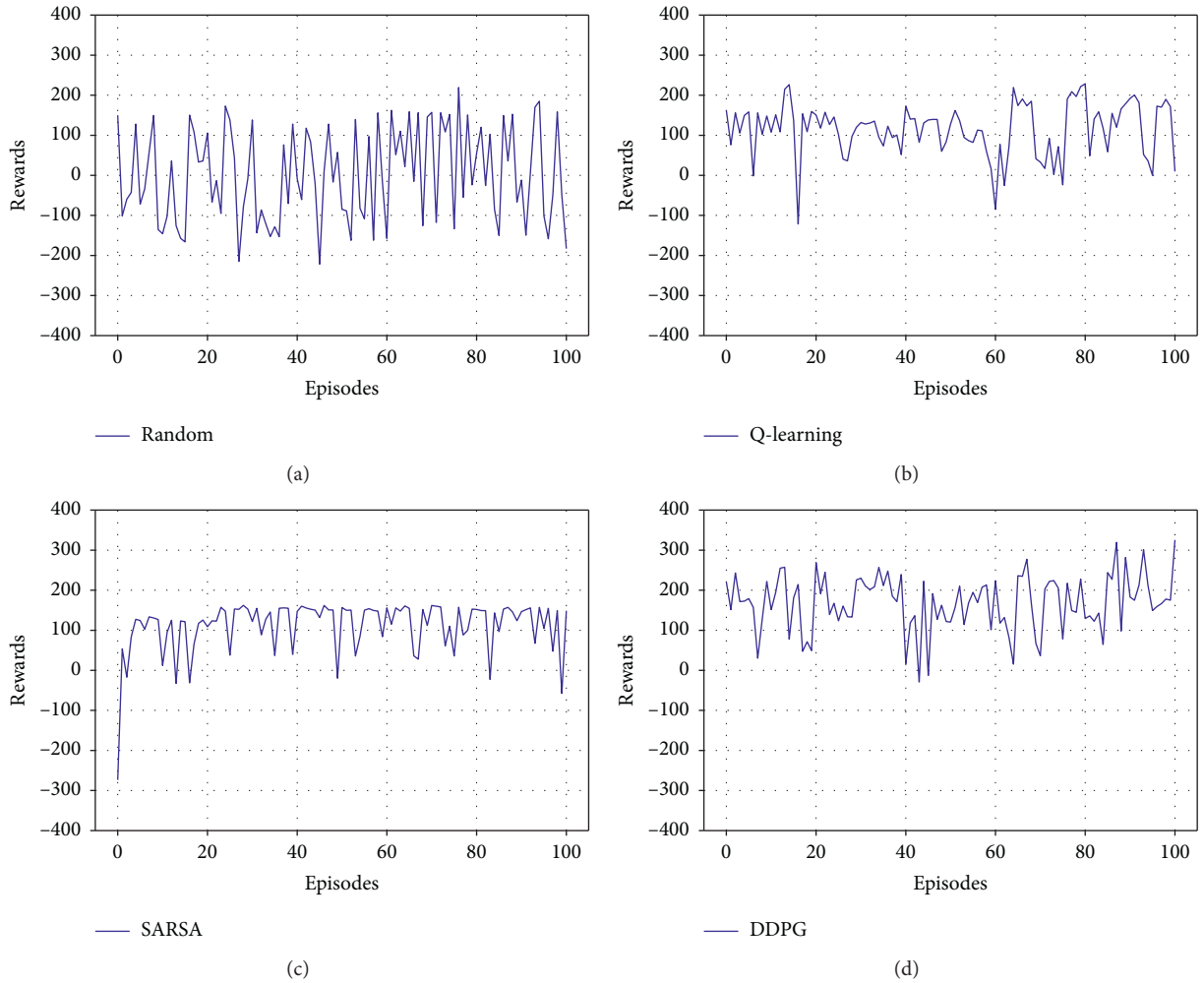

Figure 4: Reward collection by agent trained with different reinforcement learning algorithms in 100 episodes. (a) Reward collection when the agent randomly chooses action. (b) Reward collection when the agent is trained with Q-Learning. (c) Reward collection when the agent is trained with SARSA. (d) Reward collection when the agent is trained with DDPG.

to show a base line for comparison with other algorithms. The Q-learning algorithm is shown in Figure 4(b). Compared to random search algorithm, this algorithm has shown improvements as the agent is learning through Q-learning mechanism to collect rewards in the learning process. SARSA algorithm is used, and the result of reward collection is shown in Figure 4(c). The SARSA trained agents are used to look to policy to avoid malaria in a simulated human environment and has shown improvements over simple Q-learning algorithm. An even more sophisticated algorithm known as DDPG is used in the environment to collect rewards, and results are demonstrated in Figure 4(d). This algorithm shows improvements compared to all other three algorithms and demonstrated that deep learning methods can potentially collect better results in reinforcement learning algorithms.

We have combined the results of the algorithms trained in this paper in Figure 5. In random searching process, there is no learning, and therefore reward is not maximized. But in other algorithms such as Q-learning, SARSA, and DDPG, there is learning involved, and therefore reward is maximized. The overall rewards collected by different algorithms are combined in one figure (Figure 5(b)). The maximum rewards are collected by DDPG because a complex algorithm is used for collection of rewards. This comparison of three algorithms is shown in Table 1. This comparison

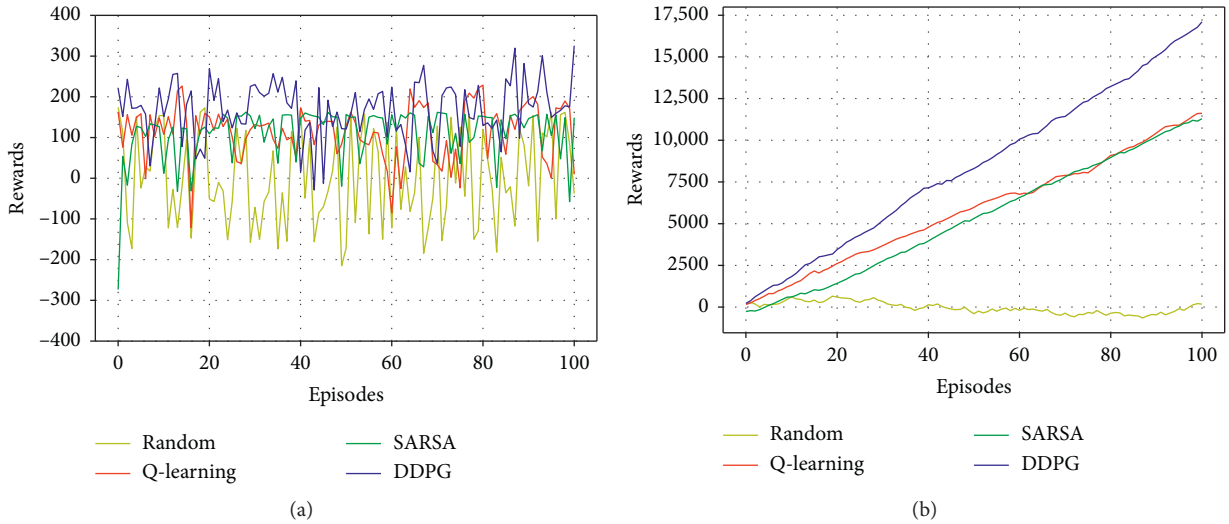

Figure 5: Comparison of reward collection by agent trained with different reinforcement learning algorithms, that is, Q-Learning, SARSA, and DDPG in 100 episodes. (a) Reward collection when the agent is trained with different reinforcement learning algorithms, that is, Q-Learning, SARSA, and DDPG. (b) Sum of rewards over time when the agent is trained with different reinforcement learning algorithms Q-Learning, SARSA, and DDPG.

Table 1: The comparison of three reinforcement learning algorithms explained in the paper in terms of best rewards and best policy when the agent is executed for 100 episodes.

Algorithm	Best reward	Optimal policy				
		Year 1	Year 2	Year 3	Year 4	Year 5
Random	174.16	[0.2, 0.7]	[0.6, 0.9]	[0.1, 0.8]	[0.4, 0.6]	[0.3, 0.1]
Q-Learning	228.77	[0.3, 0.1]	[0.3, 0.2]	[0.5, 0.2]	[0.9, 0.5]	[0.5, 0.1]
SARSA	161.74	[0.3, 0.1]	[0.3, 0.1]	[0.3, 0.1]	[0.3, 0.1]	[0.3, 0.1]
DDPG	325.55	[1.0, 0.8]	[0.1, 0.0]	[0.1, 0.8]	[0.6, 1.0]	[0.6, 1.0]

demonstrates the best policy obtained by operating in the environment to avoid malaria and the related reward collected by performing the best policy. This table demonstrates that DDPG has outperformed traditional learning algorithms.

5. Conclusion

Since the development of human civilizations, humans have always been in the quest to improve the quality of life from different perspectives. We are looking for the most comfortable accommodation, fast and secure transport, clean and healthy food, comfortable clothes, and many other things. But because of the environmental changes and different actions taken by humans, there are possibilities of different viruses entering the body of humans and affecting the quality of life of humans. For instance, malaria, flu, HIV, and dengue are some diseases that not only affect a single individual but also can affect the whole population, as the virus spreads from one person to another person. Humans over time have learned different methods to treat these diseases. There are doctors, who prescribe medicine to treat diseases, and hence diseases are in control. But the problem is that the decision of a doctor requires a huge amount of knowledge and experience, to effectively cure a disease. We think it is possible that the human effort is minimized, and some AI-based solutions are explored. Different AI-based solutions have also been explored by researchers, in the form of supervised learning such as ANN, KNN, and SVM. However, the problem with these supervised learning is that the model is trained on the existing data to make similar decisions when a similar data is presented as testing. There is a huge gap to further generalize the solution. Therefore, unsupervised learning algorithms and reinforcement learning are becoming popular. In this paper, we have explored reinforcement learning-based algorithms, where an agent interacts with the environment to get feedback and improves its state of knowledge. We have experimented with three different algorithms in reinforcement learning. These algorithms are Q-Learning, SARSA, and DDPG. All these algorithms perform better than random search, as there is learning involved. Q-Learning and SARSA are based on traditional methods of reinforcement learning. However, because of the popularity of deep learning, researchers are interested in introducing deep learning in reinforcement learning. DDPG is a deep learning-based algorithm. Our experiments have demonstrated that deep learning-based

algorithms are the most suitable algorithm for such type of complex environment, where human, their actions, environments, and their feedback play a very important role.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under Grant no. DF-458-156-1441. The authors, therefore, gratefully acknowledge DSR technical and financial support.

References

[1] A. Bowling, "The effects of illness on quality of life: findings from a survey of households in great britain," *Journal of Epidemiology and Community Health*, vol. 50, pp. 149–155, 1996.

[2] C. L. Lam and I. J. Lauder, "The impact of chronic diseases on the health-related quality of life (HRQOL) of Chinese patients in primary care," *Family Practice*, vol. 17, pp. 159–166, 2000.

[3] R. Somrongthong, D. Hongthong, S. Wongchalee, and N. Wongtongkam, "The influence of chronic illness and lifestyle behaviors on quality of life among older thais," *BioMed Research International*, vol. 2016, pp. 1–7, 2016.

[4] B. Torto, "Innovative approaches to exploit host plant metabolites in malaria control," *Pest Management Science*, vol. 75, no. 9, pp. 2341–2345, 2019.

[5] F. Binka and P. Akweongo, "Prevention of malaria using ITNs: potential for achieving the millennium development goals," *Current Molecular Medicine*, vol. 6, pp. 261–267, 2006.

[6] B. B. Tukei, A. Beke, and H. Lamadrid-Figueroa, "Assessing the effect of indoor residual spraying (IRS) on malaria morbidity in northern Uganda: a before and after study," *Malaria Journal*, vol. 16, no. 1, 2017.

[7] Y. A. Derua, E. J. Kweka, W. N. Kisinza, A. K. Githeko, and F. W. Mosha, "Bacterial larvicides used for malaria vector control in Sub-Saharan Africa: review of their effectiveness and operational feasibility," *Parasites & Vectors*, vol. 12, no. 1, 2019.

[8] T. L. I. Diseases, "Malaria vaccination: a major milestone," *The Lancet Infectious Diseases*, vol. 19, p. 559, 2019.

[9] S. J. Draper, B. K. Sack, C. R. King et al., "Malaria vaccines: Recent advances and new horizons," *Cell Host & Microbe*, vol. 24, pp. 43–56, 2018.

[10] M. Fatima, A. Baig, and I. Uddin, "Reliable and energy efficient MAC mechanism for patient monitoring in hospitals," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, 2018.

[11] I. Uddin, A. Baig, and A. Ali, "A controlled environment model for dealing with smart phone addiction," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 9, 2018.

[12] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, "Recent advances and applications of machine learning in solid-state materials science," *NPJ Computational Materials*, vol. 5, no. 1, 2019.

[13] K. Y. Ngiam and I. W. Khor, "Big data and machine learning algorithms for health-care delivery," *The Lancet Oncology*, vol. 20, no. 5, p. e262, 2019.

[14] E. Loh, "Medicine and the rise of the robots: a qualitative review of recent advances of artificial intelligence in health," *BMJ Leader*, vol. 2, no. 2, pp. 59–63, 2018.

[15] F. Jiang, Y. Jiang, H. Zhi et al., "Artificial intelligence in healthcare: past, present and future," *Stroke and Vascular Neurology*, vol. 2, no. 4, pp. 230–243, 2017.

[16] O. Frunza, D. Inkpen, and T. Tran, ""A machine learning approach for identifying disease-treatment relations in short texts," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 801–814, 2011.

[17] X. Liu, L. Faes, A. U. Kale et al., "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis," *The Lancet Digital Health*, vol. 1, no. 6, pp. e271–e297, 2019.

[18] S. Syed, M. Al-Boni, M. N. Khan et al., "Assessment of machine learning detection of environmental enteropathy and celiac disease in children," *JAMA Network Open*, vol. 2, no. 6, Article ID e195822, 2019.

[19] Y. Liu, B. Logan, N. Liu, Z. Xu, J. Tang, and Y. Wang, ""Deep reinforcement learning for dynamic treatment regimes on medical registry data," in *Proceedings of the 2017 IEEE International Conference on Healthcare Informatics (ICHI)*, Park City, UT, USA, August 2017.

[20] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, *An Introduction to Deep Reinforce-Ment Learning*, http://arxiv.org/abs//1811.12560, 2018.

[21] Y. Zheng, *Reinforcement Learning and Video Games*, MSc thesis, University of Sheffield, Sheffield, UK, 2019.

[22] I. Szita, *Reinforcement Learning in Games*, in M. Wiering, M. van Otterlo (eds) Reinforcement Learning, Adaptation, Learning, and Optimization, vol. 12, Springer, Berlin, Germany, 2012, https://doi.org/10.1007/978-3-642-27645-3_17.

[23] R. R. Torrado, P. Bontrager, J. Togelius, J. Liu, and D. Perez-Liebana, "Deep reinforcement learning for general video game AI," in *Proceedings of the 14th IEEE Conference on Computational Intelligence and Games, CIG 2018*, Maastricht, Netherlands, August 2018.

[24] B. A. Kwambana-Adams, E. K. Mulholland, E. K. Mulholland, and C. Satzke, "State-of-the-art in the pneumococcal field: proceedings of the 11th International Symposium on pneumococci and pneumococcal diseases (ISPPD-11)," *Pneumonia*, vol. 12, no. 1, 2020.

[25] A. Raghu, M. Komorowski, and S. Singh, "Model-based reinforcement learning for sepsis treatment," 2018, http://arxiv.org/abs//1811.09602.

[26] C. P. Janssen and W. D. Gray, "When, what, and how much to reward in reinforcement learning-based models of cognition," *Cognitive Science*, vol. 36, no. 2, pp. 333–358, 2012.

[27] I. Uddin, "High-level simulation of concurrency operations in microthreaded many-core architectures," *GSTF Journal on Computing*, vol. 4, no. 3, p. 21, 2015.

[28] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: a brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[29] M. Hengge and S. Leonard, *Factor Models for Non-Stationary Series: Estimates of Monthly U.S. GDP*, IHEID Working Papers 13-2017, Economics Section, The Graduate Institute of International Studies, 2017.

[30] H. Burnett, A. Earley, A. A. Voors et al., "Thirty years of evidence on the efficacy of drug treatments for chronic heart failure with reduced ejection fraction," *Circulation: Heart Failure*, vol. 10, 2017.

[31] R. P. Steeds and K. S. Channer, "Drug treatment in heart failure," *BMJ*, vol. 316, no. 7131, pp. 567-568, Feb. 1998.

[32] I. Uddin, "One-IPC high-level simulation of microthreaded many-core architectures," *International Journal of High Performance Computing Applications*, vol. 31, no. 2, pp. 152–162, 2015.

[33] S. D. W. Frost, B. R. Magalis, and S. L. Kosakovsky Pond, "Neutral theory and rapidly evolving viral pathogens," *Molecular Biology and Evolution*, vol. 35, no. 6, pp. 1348–1354, 2018.

[34] R. G. Webster and E. A. Govorkova, "Continuing challenges in influenza," *Annals of the New York Academy of Sciences*, vol. 1323, no. 1, pp. 115–139, 2014.

[35] S. Duffy, "Why are RNA virus mutation rates so damn high?" *PLoS Biology*, vol. 16, no. 8, Article ID e3000003, 2018.

[36] D. J. Hockstra and S. D. Miller, "Sequential games and medical diagnosis," *Computers and Biomedical Research*, vol. 9, no. 3, pp. 205–215, 1976.

[37] D. Hausmann, C. Zulian, E. Battegay, and L. Zimmerli, "Tracing the decision-making process of physicians with a decision process matrix," *BMC Medical Informatics and Decision Making*, vol. 16, no. 1, 2016.

[38] M. Uddin, Y. Wang, and M. Woodbury-Smith, "Artificial intelligence for precision medicine in neurodevelopmental disorders," *NPJ Digital Medicine*, vol. 2, no. 1, 2019.

[39] A. S. Ahuja, "The impact of artificial intelligence in medicine on the future role of the physician," *PeerJ*, vol. 7, Article ID e7702, 2019.

[40] D. Zois, "Sequential decision-making in healthcare IOT: real-time health monitoring, treatments and interven- tions," in *Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 24–29, Reston, VA, USA, December 2016.

[41] O. Alagoz, H. Hsu, A. Schaefer, and M. Roberts, ""Markov decision processes: a tool for sequential decision mak- ing under uncertainty," Medical decision making," *An International Journal of the Society for Medical Decision Making*, vol. 30, pp. 474–483, 2010.

[42] C. C. Bennett and K. Hauser, "Artificial intelligence framework for simulating clinical decision-making: a markov decision process approach," *Artificial Intelligence in Medicine*, vol. 57, no. 1, pp. 9–19, 2013.

[43] S. A. A. Shah, I. Uddin, F. Aziz, S. Ahmad, M. A. Al-Khasawneh, and M. Sharaf, "An enhanced deep neural network for predicting workplace absenteeism," *Complexity*, vol. 2020, Article ID 5843932, 12 pages, 2020.

[44] M. I. Uddin, N. Zada, F. Aziz et al., "Prediction of future terrorist activities using deep neural networks," *Complexity*, vol. 2020, Article ID 1373087, 16 pages, 2020.

[45] S. Parisi, D. Tateo, M. Hensel, C. D'Eramo, J. Peters, and J. Pajarinen, "Long-term visitation value for deep exploration in sparse reward reinforcement learning," 2020, http://arxiv.org/abs//2001.00119.

[46] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869–8879, 2017.

[47] S. Chae, S. Kwon, and D. Lee, "Predicting infectious disease using deep learning and big data," *International Journal of Environmental Research and Public Health*, vol. 15, no. 8, p. 1596, 2018.

[48] Y. Dong, Z. Jiang, H. Shen, and W. D. Pan, "Classification accuracies of malaria infected cells using deep convolutional neural networks based on decompressed images," in *Proceedings of the 2017 Southeastern Conference*, pp. 1–6, Charlotte, NC, USA, 2017.

[49] S. Rajaraman, S. K. Antani, M. Poostchi et al., "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," *PeerJ*, vol. 6, Article ID e4568, 2018.

[50] J. Pineau, A. Guez, R. Vincent, G. Panuccio, and M. Avoli, "Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach," *International Journal of Neural Systems*, vol. 19, no. 4, pp. 227–240, 2009.

[51] Y. Zhao, D. Zeng, M. A. Socinski, and M. R. Kosorok, "Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer," *Biometrics*, vol. 67, pp. 1422–1433, 2011.

[52] E. Yom-Tov, G. Feraru, M. Kozdoba, S. Mannor, M. Tennenholtz, and I. Hochberg, "Encouraging physical activity in patients with diabetes: intervention using a reinforcement learning system," *Journal of Medical Internet Research*, vol. 19, no. 10, p. e338, 2017.

[53] 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, KDD'18*, pp. 2447–2456, New York, NY, USA, 2018.

[54] G. Yauney and P. Shah, "Classification accuracies of malaria infected cells using deep convolutional neural networks based on decompressed imagesreinforcement learning with action-derived rewards for chemotherapy and clinical trial dosing regimen selection," in *Proceedings of the 3rd Machine Learning for Healthcare Conference*, pp. 161–226, Palo Alto, CA, USA, August 2018.

[55] I. Hochberg, G. Feraru, M. Kozdoba, S. Mannor, M. Tennenholtz, and E. Yom-Tov, "A reinforcement learning system to encourage physical activity in diabetes patients," 2016, http://arxiv.org/abs//1605.04070.

[56] W.-H. Weng, M. Gao, Z. He, S. Yan, and P. Szolovits, *Representation and Reinforcement Learning for Person-Alized Glycemic Control in Septic Patients*, 2017.

[57] O. Atan, W. R. Zame, and M. van der Schaar, "Learning optimal policies from observational data," 2018, http://arxiv.org/abs//1802.08679.

[58] O. Bent, S. Remy, S. Roberts, and A. Walcott-Bryant, *Novel Exploration Techniques (Nets) for Malaria Policy Interventions*, 2017, https://arxiv.org/abs/1712.00428.

[59] O. Gottesman, F. Johansson, J. Meier et al., *Evaluating Reinforcement Learning Algorithms in Observational Health Settings*, 2018, https://arxiv.org/abs/1805.12298.

[60] C. Yu, J. Liu, and S. Nemati, *Reinforcement Learning in Healthcare: A Survey*, 2019, https://arxiv.org/abs/1908.08796.

[61] H.-C. Kao, K.-F. Tang, and E. Y. Chang, "Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LO, USA, February 2018.

[62] A. Jonsson, "Deep reinforcement learning in medicine," *Kidney Diseases*, vol. 5, pp. 18–22, 2018.

[63] V. B. Nguyen, B. M. Karim, B. L. Vu, J. Schlötterer, and M. Granitzer, *Policy Learning for Malaria Control*, 2019, https://arxiv.org/abs/1910.08926.

[64] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869–8879, 2017.

[65] K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithms: concepts and applications in engineering design," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 5, pp. 519–534, Oct 1996.

[66] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pp. 2951–2959, New York, NY, USA, 2012.

[67] N. R. Smith, J. M. Trauer, M. Gambhir et al., "Agent-based models of malaria transmission: a systematic review," *Malaria Journal*, vol. 17, 2018.

[68] S. Vinitha, S. Sweetlin, H. M. Vinusha, and S. Sajini, "Disease prediction using machine learning over big data," *SSRN Electronic Journal*, 2018.

[69] Y. Dong, Z. Jiang, H. Shen, and W. D. Pan, "Classification accuracies of malaria infected cells using deep convolutional neural networks based on decompressed images," in *Proceedings o the SoutheastCon 2017*, pp. 1–6, Charlotte, NC, USA, 2017.

[70] T. Namba and Y. Yamada, "Risks of deep reinforcement learning applied to fall prevention assist by autonomous mobile robots in the hospital," *Big Data and Cognitive Computing*, vol. 2, no. 2, p. 13, June 2018.

[71] G. Tiburce, S. Laurentine, H. N. Ngum, I. C. Etso, and C. N.-D. Hugues, "Investigating risk factors associated with the persistence of malaria in the obang valley, north west region, Cameroon," *Journal of Public Health and Epidemiology*, vol. 10, no. 10, pp. 380–386, 2018.

[72] J.-e. Liu and F.-P. An, "Image classification algorithm based on deep learning-kernel function," *Scientific Programming*, vol. 2020, pp. 1–14, Article ID 7607612, 2020.

[73] E. Torti, M. Musci, F. Guareschi, F. Leporati, and M. Piastra, "Deep recurrent neural networks for edge monitoring of personal risk and warning situations," *Scientific Programming*, vol. 2019, pp. 1–10, Article ID 9135196, 2019.

[74] B. Ramzan, I. S. Bajwa, N. Jamil et al., "An intelligent data analysis for recommendation systems using machine learning," *Scientific Programming*, vol. 2019, pp. 1–20, Article ID 5941096, 2019.

[75] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, A Bradford Book, Cambridge, MA, USA, 2018.

[76] C. Watkins, *Learning from Delayed Rewards*, Ph.D. thesis, University of Cambridge, Cambridge, UK, 1989.

[77] T. Jaakkola, M. I. Jordan, and S. P. Singh, "Convergence of stochastic iterative dynamic programming algo- rithms," in *Proceedings of the 6th International Conference on Neural Information Processing Systems (NIPS'93)*, pp. 703–710, San Francisco, CA, USA, 1993.

[78] C. H. C. Ribeiro, *A Tutorial on Reinforcement Learning Techniques*, University of Michigen, Ann Arbor, MI, USA, 1999.

[79] D. Zhao, H. Wang, K. Shao, and Y. Zhu, "Deep reinforcement learning with experience replay based on SARSA," in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–6, Athens, Greece, 2016.

[80] Z.-x. Xu, L. Cao, C. Xiliang, C.-x. Li, Y.-l. Zhang, and J. Lai, "Deep reinforcement learning with sarsa and Q-learning: a hybrid approach," *IEICE Transactions on Information and Systems*, vol. E101, pp. 2315–2322, 2018.

[81] G. Yang, F. Zhang, C. Gong, and S. Zhang, "Application of a deep deterministic policy gradient algorithm for energy-aimed timetable rescheduling problem," *Energies*, vol. 12, no. 18, p. 3461, 2019.

[82] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8577–8588, 2019.

[83] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A theoretical analysis of deep Q-learning," 2020, http://arxiv.org/abs//1901.00137.