Computer Networks 197 (2021) 108279

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Reinforcement learning approaches for efficient and secure blockchain-powered smart health systems

Abeer Z. Al-Marridi [a], Amr Mohamed [a,*], Aiman Erbad [b]

[a] Department of Computer Science and Engineering, College Engineering, Qatar University, Qatar
[b] Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Qatar

ARTICLE INFO

Keywords:
Reinforcement learning
Blockchain
e-Health
Multi-objective optimization
Healthchain-RL

ABSTRACT

Emerging technological innovation toward e-Health transition is a worldwide priority for ensuring people's quality of life. Hence, secure exchange and analysis of medical data amongst diverse organizations would increase the efficiency of e-Health systems toward elevating medical phenomena such as outbreaks and acute patients' disorders. However, medical data exchange is challenging since issues, such as privacy, security, and latency may arise. Thus, this paper introduces Healthchain-RL, an adaptive, intelligent, consortium, and secure Blockchain-powered health system employing artificial intelligence, especially Deep Reinforcement Learning (DRL). Blockchain and DRL technologies show their robust performance in different fields, including healthcare systems. The proposed Healthchain-RL framework aggregates heterogeneous healthcare organizations with different requirements using the power of Blockchain while maintaining an optimized framework via an online intelligent decision-making RL algorithm. Hence, an intelligent Blockchain Manager (BM) was proposed based on the DRL, mainly Deep Q-Learning and it is variations, to optimizes the Blockchain network's behavior in real-time while considering medical data requirements, such as urgency and security levels. The proposed BM works toward intelligently changing the blockchain configuration while optimizing the trade-off between security, latency, and cost. The optimization model is formulated as a Markov Decision Process (MDP) and solved effectively using three RL-based techniques. These three techniques are Deep Q-Networks (DQN), Double Deep Q-Networks (DDQN), and Dueling Double Deep Q-Networks (D3QN). Finally, a comprehensive comparison is conducted between the proposed techniques and two heuristic approaches. The proposed strategies converge in real-time adaptivity to the system status while maintaining maximum security and minimum latency and cost.

1. Introduction

The impact of healthcare systems on the quality of life becomes a priority worldwide. Due to the rapid increase in the number of patients with chronic diseases, deploying the traditional healthcare model to deliver continuous monitoring raises a burden for both the patients and the doctors since the number of doctors cannot scale with the number of patients. Besides, the risk of direct interaction between doctors and patients in epidemic outbreaks imposes several problems concerning stability, scalability, and delay in getting the needed services. Hence, increasing the possibility of losing the patients' and the doctors' lives. In the United States, the National Health Council [1] reported that more than 40 million of patients with chronic diseases could not perform their daily life activities. By the current pandemic of COVID-19, of course, these numbers increase swiftly. Consequently, the research community is directed toward leveraging recent intelligent technologies, including the Internet of Things (IoT), Reinforcement Learning (RL), Deep Learning (DL), Blockchain, and Edge Computing in healthcare systems to enhance remote health monitoring and increasing the capacity of healthcare systems.

Building an e-Health system that eliminates the need for direct physical interaction between doctors and patients is essential. A smooth Integration of the e-Health system into society can be applied as almost no one moves without his smart device (e.g., phone, watch, etc.), which is a critical element in conducting an e-Health system. On the other hand, e-Health systems raise several problems involving security, privacy, scalability, computational complexity, cost, inconsistent patient profiles, a central point of failure, and much more. Researchers work on investigating solutions that enable connecting several parties in a decentralized manner while considering the mentioned limitations [2].

In 2008, Blockchain technology was launched as a decentralized ledger shared across different entities while maintaining the shared

* Corresponding author.
E-mail address: amrm@qu.edu.qa (A. Mohamed).

https://doi.org/10.1016/j.comnet.2021.108279
Received 19 November 2020; Received in revised form 16 June 2021; Accepted 30 June 2021
Available online 7 July 2021

data's consistency and integrity. There are three types of Blockchain: public, private, and consortium Blockchain. This promising technology was used in several applications in different domains, including Industry 4.0 and Industrial Internet of Things (IIoT), finance, and education, as discussed in [3–7]. Blockchain was able to overcome central challenges in these applications due to its characteristics, which can be summarized as

- Facilitates trust between heterogeneous entities with different policies and regulations while eliminating a third party's need.
- Facilitates data recovery since a copy of the ledger is available for all the Blockchain entities.
- Facilitates resilience, integrity, security, and distributed storage of the data.
- Provides an immutable structure as it follows the append-only scheme. Therefore, the added block cannot be deleted, and it enables superior fraud detection.

These Blockchain systems are formed based on a specific consensus algorithm and smart contract. The consensus algorithm ensures the consistency and integrity aspects of Blockchain among all the entities. Various types of consensus algorithms are discussed in the literature, such as Proof of Work (PoW), Proof of Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), Delegated PoS (DPoS), and more [8]. Smart contracts allow the automatic execution of business logic based on specific conditions to encode the entities' rules and priorities without third-party involvements. Smart contracts are run as transactions on top of the blockchain ledger to have the same security guarantees. The verification process of the transactions to be added into a verified block is done by the miners. The miners are followers from the Blockchain entities to ensure enforcing smart contract rules coherently and correctly [9].

Traditional healthcare systems suffer from the redundancy problems that arise from the patients' misidentification and duplication of medical records between different organizations. Blockchain has been used heavily in the healthcare domain, where data can be shared timely and securely among various organizations, private clinics, insurance companies, pharmacies, the ministry of health, and hospitals to overcome traditional healthcare systems' limitations. According to [10], by 2022, the Blockchain market in the healthcare sector will exceed $500 million.

Blockchain works toward mitigating the chances of facing inconsistent medical data, which improves the healthcare systems in terms of better data quality, less processing time, and reduced manual processing steps and reconciliation costs. Furthermore, accessibility, trust, transparency, traceability, and auditability are Blockchain features that can be efficiently deployed in healthcare systems. For instance, Blockchain can facilitate medical data analysis to investigate the root causes of medical phenomena such as virus infections by correlating data and events from multiple entities [11].

Meanwhile, Blockchain's usage raises some challenges, for example, processing sophisticated transactions while ensuring privacy and security requirements. Besides, processing, sharing, and storing enormous collected real-time health data while allowing secure authorized access by the Blockchain entities with acceptable latency and cost. Therefore, it is essential to monitor the security, latency, and cost as conflicting objectives in any healthcare system and specifically in a system that uses Blockchain technology in facilitating healthcare services. The authors in [12–14] consider the latency and security only while optimizing in Blockchain. However, the cost is an important aspect that cannot be ignored.

An intelligent learning-assisted decision-making approach is needed to change the Blockchain configuration adaptively to the incoming transaction characteristics and optimize the blockchain performance. The rapid development of Artificial Intelligent (AI) in the last few years proves its ability to learn from a massive amount of data effectively. AI approaches were deployed heavily in different domains, including the healthcare sector, to facilitate building smart health systems. It was used for various tasks such as data analysis, preprocessing, recognition, classification, development of new medications, etc. [15]. Reinforcement Learning (RL) is an AI approach under Machine Learning (ML), which was used in healthcare applications, as discussed [16–18]. RL is a decision-driven approach that learns the environments' dynamics and relates the states' transmission to make the right decision, maximizing the long-term reward. RL approaches can outperform other decision-making techniques as it considers the instantaneous (short-term) reward at a particular state, along with finding a long-term policy that maximizes the system's reward over time [19].

Deep Learning was integrated with the traditional RL, forming Deep Reinforcement Learning (DRL) to enhance RL's performance and overcome its limitations, such as managing huge state space smoothly. DRL was used in different optimization frameworks, as mentioned in [20, 21], due to its ability to make a real-time decision based on a trained model. The trained model maps the states of the environment and the decisions to maximize the reward; while eliminating the need to execute the complex traditional multi-objective optimization problem at every step, causing useful implications on the average complexity.

In this paper, we propose Healthchain-RL, an optimized, decentralized e-Health Blockchain framework that adopts Deep Reinforcement Learning (DRL) to dynamically adapt the behavior of the Blockchain network to provide optimal QoS based on health application requirements. The framework optimizes the trade-off among the system conflicting objectives: latency, security, and the cost, where we aim to minimize both latency and cost while maximizing security. Deep Q-Network (DQN), Double Deep Q-Network (DDQN), and Dueling Double Deep Q-Network (D3QN) are the potential DRL approaches proposed where one can be deployed based on the system administrator preference. The main contributions are listed as follows:

- Propose Healthchain-RL, a multi-objective optimization Blockchain framework in e-Health systems that maps application requirements such as urgency, security, and transaction age to Blockchain configuration, including the number of transactions per block and minor's selection.
- Formulate the Markov Decision Process (MDP) of our proposed Healthchain-RL and consider the temporal aspects of Blockchain, in addition to introducing the reputation of Blockchain miners.
- Propose an intelligent Blockchain manager (BM) based on reinforcement learning techniques: Deep Q-Network (DQN), Double Deep Q-Network (DDQN), and Dueling Double Deep Q-Network (D3QN) toward optimizing latency, security and cost in real-time while considering the requirements of Blockchain entities and reacting efficiently to system dynamics.
- Evaluate the proposed Healthchain-RL against multiple approaches, including the Greedy and Random-Selection approaches, while showing the proposed BM's superior performance.

The rest of the paper is organized as follows: Section 2 explores some related works. Healthchain-RL explanation presented in Section 3. In Section 4, we present the problem formulation considering the trade-off between the security, latency, and cost in Healthchain-RL by leveraging Deep Reinforcement Learning. Blockchain Manager (BM) Agent-based techniques are discussed in Section 5. In Section 6, the proposed framework's performance evaluation is presented and compared with other heuristics, and finally, Section 7 concludes the paper.

2. Related works

This section presents some related works concerning Blockchain and Reinforcement Learning in e-Health systems and highlights how to leverage Reinforcement Learning for Blockchain optimization.

Table 1
Blockchain-powered applications using DRL.

Ref.	Field	Trade-off objectives	RL-Approach
[22]	Vehicular Ad Hoc Networks	Trust features of Blockchain nodes and vehicles, # of consensus nodes,Blockchain computational capability	DDQN
[23]	Vehicular Edge Computing	Transmission energy consumption, content caching energy consumption and content delivery latency	DQN
[24]	Industrial Internet of Things	Geographical fairness,and total power consumption	Distributed DQN
[13]	Industrial Internet of Things	scalability, decentralization, latency, and security	DQN
[25]	Wireless Networks	Energy consumption, caching resources and cost	DQN
[26]	IoT Monitoring Applications	Auditability, delay and cost	DQN
[27]	Cognitive Radio Network	Network throughput, time scheduling, and cost	D3QN
Healthchain-RL	**e-Health**	Latency, security and cost	DQN / DDQN / D3QN

2.1. Blockchain in e-Health systems

Due to Blockchain's characteristics, it was beneficial to integrate it into healthcare applications, which requires high confidentiality while sharing medical information and records. In [28], the authors' proposed Gem Health Network, a decentralized medical record ledger based on Ethereum Blockchain, allows real-time access by different healthcare operators. However, Gem has some drawbacks, including recognition of the patient identity, lack of key replacement capabilities, and scalability. MedShare [29], is a medical Blockchain-based framework assures provenance, access control, auditing, and privacy of medical data while reducing malicious activities on data access by the entities. However, it suffers from the same obstacles as Gem. The work in [30] proposed a healthcare Blockchain system that enables encrypted searching of electronic health records (EHRs), resulting in scalability problems. PHS Blockchain-based on artificial systems, computational experiments and parallel execution (ACP) approach framework proposed by [31] also struggles with scalability and latency and some security threats. BSPP [32] and BLOCHIE [33] use a dual Blockchain framework for different healthcare systems usage. Both approaches have these drawbacks: scalability, latency, computation cost, and storage capacity. A private healthcare Blockchain framework called MedChain suffers from scalability and requires patient approval, which increases the overhead on the framework [34]. OmniPHR framework [35] tries to solve the scalability problem in MedChain, but the patient's authentication is still needed. Many other proposed Blockchain frameworks in healthcare, such as [36–39] suffer from administrative scalability problems.

2.2. Reinforcement learning in e-Health systems

Markov decision process (MDP) is a formulation of decision-making problems, where a decision-maker (the agent) decides an action for a given state while interacting with the environment (formulation of the problem), causing a transition to a new state. At the same time, the agent receives a reward based on the action taken. Therefore, MDP consists of five main components: the agent, the environment, the states, the actions, and the reward. Reinforcement Learning (RL) is an AI approach under Machine Learning (ML) introduced by [40], which solves problems formulated as MDP. The agent has mainly two objectives; first, effective interaction with the environment by selecting an action based on a behavior policy. Second, evaluation of the action taken for a given state to maximize the cumulative reward received over time based on a target policy. Eventually, this later policy is considered as the optimum.

The learning process of the agent will follow either the on-policy or the off-policy approach. On-policy learning means that both the behavior and target policies are the same (e.g., SARSA). Otherwise, it is called off-policy learning (e.g., Q-Learning). During on-policy learning, the agent learns the value function based on the current policy used to derive the action; thus, it tries to evaluate and improve the same policy to reach the sub-optimal policy. Meanwhile, in the off-policy learning, the agent learns the value function independently from the action by continuously updating the policy, as part of the exploration, to learn the optimal policy [19].

In this paper, the off-policy approach is considered, precisely, the Q-Learning approach, where the agent attempts to find the optimal action for a particular state and save it in a Q-table. Neural networks are introduced to this approach to replace the Q-table concept as function approximators, especially when the action and state spaces are enormous, known as Deep Q-Networks (DQN). DQN is used in the healthcare sector, including medical images report generations, extraction of clinical concepts, and optimizes the treatment policies for chronic illnesses as in [16–18].

2.3. Reinforcement learning for blockchain optimization

RL and Blockchain were used to serve different fields such as vehicular Ad Hoc networks, wireless networks, and the Internet of Things (IoT) [22–26,41]. In [27], the authors proposed the Dueling Double Deep Q-Network (D3QN) approach to manage the selection of the optimal Blockchain network based on the transaction fees considering different constraints. The authors in [22] used the Double Deep Q-Learning (DDQN) prioritized experience replay approach to find the optimal number of consensus nodes and the trusted neighbor vehicles. RL was utilized in [13] to increase the throughput, consensus algorithm and block producer, block size, and block interval. The authors in [42] solved the mining problem in public Blockchain without knowing the Blockchain parameters using RL. As discussed in the previous subsections, RL and Blockchain technologies were heavily used to serve healthcare applications, which motivate us to integrate the power of both in e-Health systems. In [43], the authors presented the importance of integrating Blockchain in e-Health systems considering different stakeholders participating in it, such as the ministry of public health (MOPH), pharmacies, insurance companies, external edge (EE), hospitals, etc. The trade-off between security, latency, and the cost was considered in [43], where the optimization was solved based on a Greedy policy without considering the temporal aspects of the transactions and the overall system's future state.

In this paper, we introduce an adaptive, secure, and intelligent healthcare system leveraging reinforcement learning and Blockchain to optimize three conflicting objectives, namely, the latency, security, and cost based on urgency, security levels of the transactions, and the age of the transactions, along while considering the optimal number of miners and transactions per block.

To the best of our knowledge, this is the first work leveraging deep reinforcement learning to optimize Blockchain in e-Health systems. Three state-of-art off-policy-based learning algorithms, DQL, DDQN, and D3QN, are used to implement the agent and discussed in Section 5. Table 1 summarizes the Blockchain-Powered applications in several domains while leveraging Deep Reinforcement Learning.

3. System model

Healthchain-RL is our proposed solution to enable secure, adaptive, and online healthcare data sharing between different entities. The framework is represented in Fig. 1, where consortium medical Blockchain is proposed, among several healthcare entities can access,

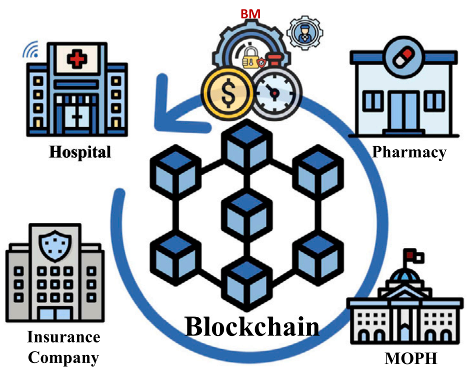

Fig. 1. Our Proposed Healthchain-RL. General representation where the roles of the BM circulate between all the entities to ensure fairness.

share, and process the available medical data on a distributed ledger, based on their predefined eligibility level in the smart contract. Besides, the entities might have their local network to collect, process, and prepare transactions holding critical data that needs to be shared in the Blockchain. Pre-processing the local networks' data includes summarization, clustering, or compression [44] using AI techniques. However, this paper will focus on Blockchain network optimization and the trade-off between security, latency, and cost considering the transaction's constraints: security and urgency levels.

The concept of the Blockchain manager is proposed to address the urgency and security of medical data along with optimizing the three mentioned objectives. This paper proposes an intelligent Blockchain manager using reinforcement learning techniques to adapt the system status fluctuation while considering future behavior.

In the following subsections, we explain the Blockchain entities, the Blockchain network, the intelligent Blockchain manager, along with the optimization problem we are tackling.

3.1. Blockchain entities

Different stakeholders can be part of this framework to facilitate a fast, scalable, secure, and intelligent distributed healthcare system. These entities can either share their healthcare data or access the available data on the Blockchain to analyze, research, adopt new health laws, and much more. Hospitals, pharmacies, insurance companies, and the Ministry of Public Health (MOPH) are examples of possible entities in such a framework. The smart contract introduces the business logic behind Blockchain, including all the rules, instructions, authority, and priority levels of each entity. All the entities should sign and apply the smart contract to all executable transactions before being saved into the Blockchain ledger.

3.2. Blockchain network

Healthchain-RL Blockchain is a consortium, immutable, and decentralized healthcare data sharing network among the entities mentioned above to securely transmitting, accessing, processing, and sharing medical data. We focus on the consortium Blockchain framework with the Delegated Proof of Stake (DPoS) consensus scheme, an alternative to Proof of Stake (PoS) to ensure scalability and defend the Blockchain from malicious usage and centralization. It uses the election and voting process to select the miners with moderate costs.

In this paper, the Blockchain configuration is adaptable to the requirements of the transactions held by different entities and manages the trade-off between three conflicting objectives, security, latency, and cost. The main affected configurations are the number of transactions per block and the number of miners. The intelligent Blockchain manager decides these two parameters.

3.3. Intelligent Blockchain Manager (BM) — optimizer

Blockchain Manager (BM) holds the most critical role in the proposed framework, where it optimizes the Blockchain configuration based on three main attributes of the transaction: urgency, security, and transaction age. All available transactions at a particular time step will be collected by the BM to decide the number of transactions admitted into a block and verified through a certain number of miners based on their storage, computation resources, and prices in order to be shared on the Blockchain network. However, those configurations should not change continuously to avoid any further additional computational and monetary costs. Therefore, the frequency of changing the Blockchain configuration can be decided by all entities unless a sudden event occurs. Such actions could be specified in the smart contract of the Blockchain and agreed upon by all entities. For example, the algorithm can run every certain amount of time unless a price-changing event occurs. The role of BM installed at the entity level of the Blockchain can either be centralized (not recommended for security aspects) or decentralized by circulating the role between the Blockchain entities. Circulation should be on a defined and agreed timeline, where at a particular time step, the BM tasks will be delegated to the following entity in the Blockchain sequentially or following a specific approach decided by the smart contract [43,45]. Circulation is recommended to maintain fairness and security; therefore, it is applied in the proposed Healthchain-RL framework.

As mentioned before, we plan to optimize the trade-off between security, latency, and cost in our proposed Healthchain-RL, where both latency and cost should be minimized while maximizing security based on the submitted characteristics transactions. Security, urgency, and the transactions' age are the primary considered transaction's characteristics defined based on different data compression, classification, summarization, and event detection at the edge level of the architecture (local network). The age concept describes the transaction duration of waiting until being processed and added to the Blockchain network (ledger). More details are explained in Section 4. The following scenarios can illustrate the effect of urgency and security: at high-level urgency, transactions could require low latency and minimum security, such as emergency notifications. In the mentioned case, the Blockchain becomes restricted by a minimum number of miners. Meanwhile, high-level security transactions require more miners to participate in the verification stage; however, the latency and the cost will be affected as the verification delay, and fees increase by increasing the number of miners.

In the proposed framework, we could try to map the patient's conditions into different Blockchain configuration modes while considering three conflicting objectives: security SE, latency L, and cost C. Eq. (1), represents the utility multi-objective function, where at a particular time step t, the selected number of miners m_t and the number of transactions tr_t are constrained by $1 \leq m_t \leq M$ and $1 \leq tr_t \leq T_r$. M is the maximum number of miners participating in the verification process, and T_r is the maximum number of transactions allocated in a block. Moreover, the transaction age $\breve{a}_t$ should not exceed a certain threshold $\breve{a}_{th}$, which is defined as the maximum duration of time that the transaction can wait in the BM submission queue Q before it is added to a block and submitted to the Blockchain network. α, β, and γ are the weighting factors that decide how valuable the latency, security, and cost are as a part of the total objective, based on the system administrator requirements, where their summation equals one. The system administrator requirements can be based on the business logic and the nature of the data.

To ensure having comparable, unitless, and normalized equations, the objectives' maximum values were used. l_m, s_m, and c_m symbolizes the maximum latency, security, and cost, respectively.

$$\begin{aligned} \underset{m_t, tr_t}{\text{minimize}} \quad & \alpha \cdot \left(\frac{L}{l_m}\right) + \beta \cdot \left(\frac{s_m}{SE}\right) + \gamma \cdot \left(\frac{C}{c_m}\right) \\ \text{subject to} \quad & 1 \leq m_t \leq M, \\ & 1 \leq tr_t \leq T_r, \\ & \breve{a}_t \leq \breve{a}_{th}. \end{aligned} \tag{1}$$

Eq. (2) indicates the security SE, where g is a system coefficient, m is the number of miners selected by the BM, and q is an indicator factor representing the network scale and its' value greater than or equal to two [43].

$$SE = g.m^q \tag{2}$$

Eq. (3) refers to the total latency (L) of creating, verifying, broadcasting and uploading a block [43]. tr is the number of transactions per block, B is the size of the transaction, G is the computation resources needed to verify a block, x_i refers to the computational resources of miner i, ψ is predefined parameter described in details in [46]. O is the size of the verification feedback, r_u is the uplink transmission rate from the miners to the BM, which is defined in (4) and r_d is the downlink transmission rate from the BM to the miners, which is defined in (5). In (4) and (5), w is the bandwidth, SNR_d and SNR_u are the signal to noise ratio of the downlink and uplink [43].

$$L = \frac{tr.B}{r_d} + \max_{i\in 1,\dots,M}\left(\frac{G}{x_i}\right) + \psi.tr.B.m + \frac{O}{r_u} \tag{3}$$

$$r_u = w\log(1+SNR_u) \tag{4}$$

$$r_d = w\log(1+SNR_d) \tag{5}$$

The last objective that we are targeting to minimize is the cost C represented in Eq. (6), where for each miner i, the computational cost is c_i considering the number of selected transactions. c_i equals to its available resources x_i multiplied by the prices of using the resources p_i, $c_i = x_i \times p_i$ [43].

$$C = \frac{\sum_i^m c_i}{tr} \tag{6}$$

4. Deep reinforcement learning-based approach

The architecture in [43] was adopted and modified to enable running the system online under different circumstances. This paper's main objective is to ensure an adaptable, intelligent, and secure healthcare system by mapping the patients' status to the optimized Blockchain configuration considering the trade-off between the three conflicting objectives in any healthcare system, particularly security, latency, and cost. This optimization problem is considered an NP-hard problem. The authors in [43] solved it via a Greedy approach where it does not consider the temporal behavior of getting transactions in the Blockchain framework, which has a direct impact on the latency as well as the future state of the system. Furthermore, the time and computational complexity of using the Greedy approach for solving the optimization at every time step is costly and unreliable in real-time.

Therefore, reinforcement learning approaches, specifically Q-learning and its variations, are adopted to overcome the mentioned limitations in speed, complexity, and future accumulated performance. At each time step, the BM queue Q of the transactions will be checked to decide on Blockchain's configurations. The decision is made based on the predefined entities' requirements, security, and urgency levels, along with the age of the transactions in the BM queue Q to optimize our utility function. The utility function represents the trade-off in a consortium Blockchain by maximizing security while minimizing latency and cost.

The multi-objective optimization problem is formulated as a Markov Decision Process (MDP) represented by five-tuples $\langle \mathcal{S},\mathcal{A},\mathcal{T},\mathcal{R},\lambda\rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the state transition, $\mathcal{R}$ is the reward function and $\lambda \in [0,1)$ is the discount factor. At every time step t, a representation of the environment state $s \in \mathcal{S}$ is sent to the agent. The agent is expected to execute an action $a \in \mathcal{A}$ using a policy $\pi(a|s)$ to receive a reward $r_t \in \mathbb{R}$. Hence, a transition to next state s' occurs with a probability $P(s'|s,a) = \mathcal{T}(s,a,s')$. The deep expectation is introduced, where state–action of a specific policy π is function defined as $Q^\pi(s,a) = \mathbb{E}_{a\sim\pi,s'\sim P}[R_t|s_t = s, a_t = a]$, which encapsulates the sum of rewards resulting from taking an action a in a state s, following the policy π.

Subsequently, Deep Reinforcement Learning (DRL) is utilized to solve the MDP. The following subsections discuss the elements of our goal-directed learning framework in details, the state space $\mathcal{S}$, the action space $\mathcal{A}$, the environment design including the state transition, and the formulation of the multi-objective reward function in the context of our optimization problem, and finally the design of the agent.

4.1. State space

At every time step, the system state s belongs to the state space $\mathcal{S}$, which is represented by two vectors. The first vector represents the queue of the BM, Q, which includes the transactions considering their three characteristics: security, urgency, and age. The second vector denotes the reputation score for all Healthchain-RL miners. Reputation gives a favor of fairness between the miners, and it ensures having more realistic scenarios. The reputation concept was addressed in the literature, and different approaches were explicitly proposed to address their application requirements and solve a particular limitation of their system as in [47–52].

In this paper, all the miners' reputation score varies between zero and one, where zero is the worst, and one is the best as recommended in [53]. The reputation score changes every time step randomly based on Gaussian distribution and directly affects the reward function based on the taken action. The security and urgency values are between zero and one set by entities, where zero represents the minimum security and utmost urgency and one represents maximum security and minimum urgency.

We set $s_t = \left\langle \left[\check{S}_1,\check{U}_1,\check{a}_1 \cdots \check{S}_T,\check{U}_T,\check{a}_T\right],\left[R_1 \cdots R_M\right]\right\rangle$ for all $s_t \in S$ where $\check{S},\check{U}$ and $\check{a}$ refer to security, urgency and transaction age accordingly. Moreover, the reputation of each miner participate in the verification process is considered and represented by $R_{(m)}$ where $m \in M$ and M is the maximum number of miners.

4.2. Action space

At every time step t, the BM needs to select an action $a_t \in M \times T_r$ actions, where M is the maximum number of miners participating in Healthchain-RL, and T_r is the maximum number of transactions allocated in one block which is also equal to the queue size of the BM Q. Therefore, a_t represents both the selected number of transactions tr_t and number of miners m_t, which are calculated as stated in Eqs. (7) and (8).

$$tr_t = \begin{cases} Q, & a_t \mod Q = 0 \\ a_t \mod Q, & \text{otherwise} \end{cases} \tag{7}$$

$$m_t = \left\lceil \frac{a_t}{Q} \right\rceil \tag{8}$$

4.3. Environment design

The MDP environment represents the received transactions from the entities, where BM needs to interact along with the Blockchain network. The environment receives the action a_t from BM at time step t, then a transition to the next state s' occurs, and it emits a reward signal r, which is explained in detail in the following subsections. In our Healthchain-RL, a series of episodes of experiences (States, Actions, Rewards, Transitions) will be generated by the environment to support BM's learning process. Note that BM is not aware of the environment design. It uses those experiences to interact and observe the rewards for various decisions in different states and then learn the optimal policy π^* to map the states with their best actions.

4.3.1. Age concept and state transition

As mentioned earlier, the transactions have the age as a parameter updated at every time step $\check{a}_t$. Introducing the age concept is essential to facilitate the state transition, considering the Blockchain's temporal behavior and the requirements of deploying reinforcement learning. Accordingly, the age concept maintains a manageable state transition. It provides the BM with samples for the next state s' following the probability distribution over the next states conditioned on the current state s_t and action a_t without directly modeling the state transition distribution. It is worth mentioning that considering the transaction age prevents the case when a transaction will never be allocated into a block (i.e., Zombie transaction problem) if its urgency requirements are low. Therefore, deciding the number of allocated transactions in future actions will consider the transaction age since the time it has been added to the queue Q of the BM until being added to the Blockchain network. In the typical Greedy approach [43], the temporal structure is not considered.

4.3.2. Reward function formulation

The reward function's formulation represents the ultimate goal of optimizing the trade-off between security, latency, and cost while meeting the application-level requirements, involving security and urgency levels. The reward r_t is a function based on state s_t and action a_t where t is the current time step. In RL, the algorithm maximizes the reward, unlike our optimization, which aims to minimize the trade-off. Therefore, the minimization function explained earlier is converted to maximization, as stated in Eq. (9). α, β, and γ are the weighting factors to indicate the importance. The three conflicting objectives are calculated using Eq. (2), (3), and (6) and affected by the selected miners' reputation. The BM selects the value of tr_t and m_t as the current state's action. (10) represents the reward function of our framework restricted by five main conditions, (10a), (10b), (10c), (10d), and (10e). Starting with the transaction age threshold; it is an application-level parameter, the reward function is assigned to zero if the transaction age exceeds the threshold $\check{a}_{th}$. Otherwise, there are four conditions where a penalty will be applied to the reward if the selected transactions and miners do not align with the requested security and urgency levels. The penalty concept was introduced to avoid the sparse reward problem, and its value is an application-level decision. $\hat{P}_s$ and $\hat{P}_u$ represent the security and urgency penalties, respectively. l_s and l_u are the permissible limits of the excepted requirement for security and urgency, accordingly.

$$\check{R} = \frac{\sum_{i=1}^{m} R_i}{m} \cdot (\alpha \cdot (1 - \frac{L}{l_m}) + \beta \cdot \frac{SE}{s_m} + \gamma \cdot (1 - \frac{C}{c_m})) \tag{9}$$

$$r_t = \begin{cases} 0, & \text{(a)} \\ \check{R}, & \text{(b)} \\ \check{R} - (\check{R} \times (\hat{P}_u) \times (\alpha)), & \text{(c)} \\ \check{R} - (\check{R} \times (\hat{P}_s) \times (\beta)), & \text{(d)} \\ \check{R} - (\check{R} \times (\hat{P}_u + \hat{P}_s) \times (\alpha + \beta)), & \text{(e)} \end{cases} \tag{10}$$

Knowing that the average security requirement of the selected transaction is $avg_s = \sum_{i=1}^{t_r} \check{S}/t_r$ and the average urgency requirement of the selected transaction is $avg_u = \sum_{i=1}^{t_r} \check{U}/t_r$.

The lower bound of the security is $LS = avg_s(1 - l_s)$ and the upper bound is $US = avg_s(1 + l_s)$. The lower bound of the latency is $LU = avg_u(1 - l_u)$ and the upper bound is $UU = avg_u(1 + l_u)$. Our objectives in Healthchain-RL are restricted by application-level bounds, where they should not go beyond them. The conditions' definition in the reward function (10) are discussed in the below list.

- (10a): $\check{a}_t > \check{a}_{th}$
  The age of the one or more of the transactions exceeds the age threshold.
- (10b): $\check{a}_t \leq \check{a}_{th} \& (LS \leq SE \leq US) \& (LU \leq L \leq UU)$
  The age of all the transactions in Q is below $\check{a}_{th}$ and both the security SE and the latency L are within the accepted range.
- (10c): $\check{a}_t \leq \check{a}_{th} \& (LU > L | L > UU)$
  The age of all the transactions in Q is below $\check{a}_{th}$ and the latency L is beyond the accepted range.
- (10d): $\check{a}_t \leq \check{a}_{th}$ & $(LS > SE | SE > US)$
  The age of all the transactions in Q is below $\check{a}_{th}$ and the security SE is beyond the accepted range.
- (10e): $\check{a}_t \leq \check{a}_{th}$ & $(LS > SE | SE > US)$ & $(LU > L | L > UU)$
  The age of all the transactions in Q is below $\check{a}_{th}$ and both the security SE and the latency L are beyond the accepted range.

4.4. Agent design

Finding the best mapping between the state and the action (optimal policy π^*) that gives the maximum reward is the agent's main objective. Therefore, the agent needs to use the available state's samples and builds a precise estimation of the optimal action-value function $Q^{\pi^*}(s, a)$ by observing the expected reward of choosing an action a for a state s. Hence, the defined expectation of the Q^* function in Eq. (11) can be evaluated by implementing the recursive Bellman equation [19] as stated in Eq. (12). As Q_k is the estimation of Q^* at iteration k and will converge to Q^* as $k \to \infty$.

Due to the high dimensionality in the Q-function (including the optimal Q^*), it is worth taking advantage of the neural networks as general function approximators during the implementation. The following section explains our proposed Deep Q-Network comprehensively.

$$Q^*(s, a) = \mathbb{E}_{s' \sim \mathcal{T}}[r_t + \lambda \max_{a'} Q^*(s', a' | s_t = s, a_t = a)] \tag{11}$$

$$Q^*_{k+1}(s, a) = \mathbb{E}_{s' \sim \mathcal{T}}[r_t + \lambda \max_{a'} Q^*_k(s', a' | s_t = s, a_t = a)] \tag{12}$$

5. Blockchain manager (BM) agent-based techniques

Deep Q-network, with its variations, are considered in our proposed solution where the Q function takes into account the neural network parameters θ. As mentioned in [19], the parameters of the neural network are optimized toward the approximated Q^* using the Temporal Difference error (TD-error) loss function $\mathcal{L}$, where at each iteration k, the loss $\mathcal{L}$ is given by Eq. (13). Y_k represents the intermediate estimate of expected future rewards, which enables online learning from the samples. It is called TD-target and is defined in Eq. (14).

$$\mathcal{L}_k(\theta_k) = (Y_k - Q(s, a; \theta_k))^2 \tag{13}$$

$$Y_k = r + \lambda \max_{a'} Q(s', a'; \theta_k) \tag{14}$$

5.0.1. Proposed reinforcement learning approaches

Aforementioned, the BM does not know the state transition model; hence, it is essential to use a model-free algorithm such as Deep Q-Network (DQN) in order to estimate Q^*. However, the convergence of this approach is not guaranteed; thus, it is beneficial to add some enhancements that help improve the convergence of the weight of the neural network and stabilize the training (e.g., experience replay and soft updates of the neural networks).

Experience replay ($\mathcal{D}$) is an important concept used in model-free algorithms to enable training, concerning not only the current experience but also the previous experiences accumulated by the agent (BM). Consequently, the data efficiency increases by using saved experience samples during the updates and ensures reducing the correlation amongst experience samples in the update using uniform sampling, which implies a reduction in the variance [54]. Fig. 2 represents the interaction flowchart between the environment and the agent (BM) considering the integration of experience replay and soft updates of the neural networks. The interaction starts by observing a state by the agent (BM) from the environment. Accordingly, the agent takes an action that represents the number of selected transactions in the state and the number of miners to verify the transactions. The resulting reward for the action taken and the new state will be calculated and

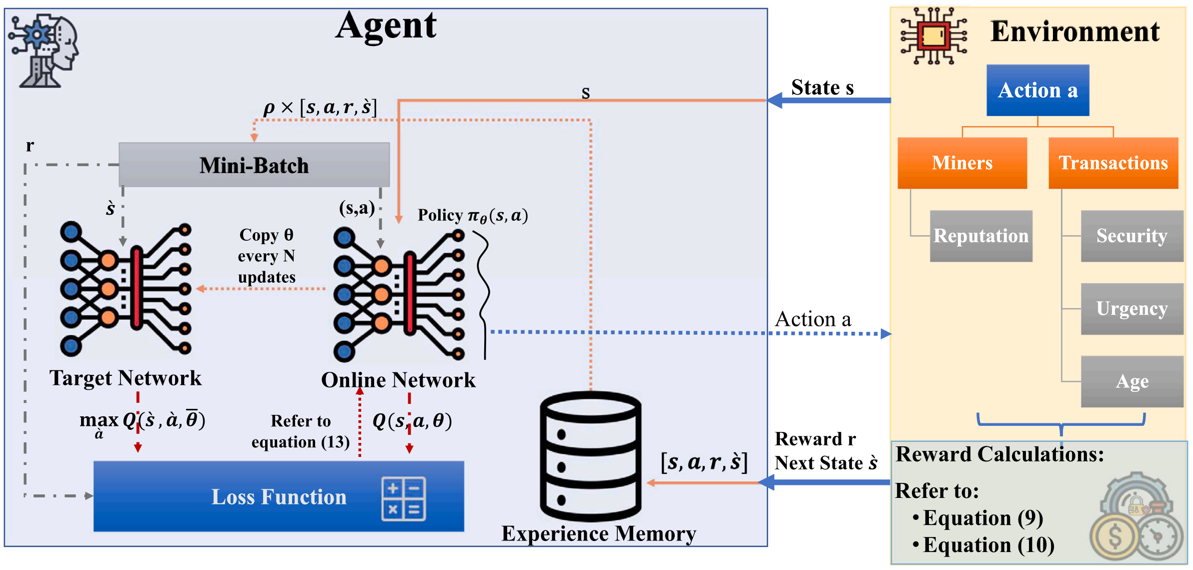

Fig. 2. Flowchart of Deep Q-Learning Algorithm in Healthchain-RL system considering the integration of experience buffer and soft updates.

analyzed by the environment and sent back to the agent for the next action. Learning the optimal action for a particular state starts by taking several random actions for exploration purposes. All previous experiences, including the state,taken action, corresponding reward, and new state, will be stored in an experience replay memory to enhance the stabilization of the learning process. The target network parameters will be updated every certain number of time steps N.

In healthcare applications, making the right decision and adapting to sudden changes significantly impact the patient's status and system performance. Fast response and efficient decision-making are critical factors in healthcare systems, which cannot be handled effectively by the traditional decision-making approaches. Therefore, this paper investigates the performance of the state-of-art off-policy approaches in making an online decision and their adaptability to sudden changes. DQN with its variations, including Double Deep Q-Network (DDQN) and Dueling Double Q-Network (D3QN), are studied. DQN is the original representation where the neural networks replace the Q-table. DQN may raise an overoptimistic issue and false-positive problem, which is critical, especially in healthcare systems. Hence, DDQN is an enhancement of DQN to overcome the overoptimistic problem. D3QN approach considers not only the state–action value but also the likelihood of being in a particular state. Hence, the architecture of the neural network model used in D3QN is different. The following subsections discuss each technique in detail.

Deep Q-Network (DQN). DQN is a model-free algorithm proposed by [55] to eliminate the need for Q-tables in the traditional Q-learning and replaces it with neural networks that estimate the action. DQN introduces two networks, the online and target networks. The network's input is the state, along with the number of possible actions $\mathcal{A}$. The online network will be continuously updated using gradient descent, while the target network parameters are updated after a certain number of episodes [54]. The main advantage of DQN compared to the traditional Q-learning approach is that DQN uses machine learning models as function approximators to eliminate the need for lookup tables.

$$Y_k^{DQN} = r + \lambda \max_{a'} Q(s', a'; \bar{\theta}) \tag{15}$$

$$\mathcal{L}_k(\theta_k) = \mathbb{E}_{(s,a,r,s')\sim \mathcal{U}(D)}[(Y_k^{DQN} - Q(s,a;\theta_k))^2] \tag{16}$$

$$Q_{k+1}(s_t, a_t, \theta_t) = Q_k(s_t, a_t, \theta_t) + \varphi(Y_k^{DQN} - Q_k(s_t, a_t, \theta_t)) \tag{17}$$

Eq. (15) represents the Y when using DQN, where the parameters are frozen to some previous values $\bar{\theta}$. The updated loss function $\mathcal{L}$ considers the saved previous experience concerning Y^{DQN} in replay buffer $\mathcal{D}$ as presented in Eq. (16). Therefore, the Q function is stated in Eq. (17), where k represents the episode length, λ is the discounting factor that ensures that the BM will not rely totally on future rewards, and θ is the set of neural network parameters (weights and biases for l_h hidden layers). Rectified linear unit (ReLU) ∂ activation function is used which is a positive function and equals to $\partial = \max(0, x)$. φ represents the learning rate, which specifies how far the current estimate is adjusted toward the update target.

The online learning from the samples is enabled by applying an intermediate approximate Y of the next state's rewards. The smooth-L1 loss function is defined and optimized using Adam Optimizer. Adam is used instead of the classical Stochastic Gradient Descent (SGD) methods, where the network weights are updated iteratively based on the training data samples. The defined Y depends on the used Q-Learning approach.

Double Deep Q-Network (DDQN). DDQN is similar to DQN, but with modification to the target network [56]. In DQN, the online and target networks use the same maximum operator. Therefore, the estimated actions are overoptimistic. An over-optimistic problem arises mostly in large-scale problems, where the networks are favored to select the best action with the maximum Q-value, leading to a false-positive problem.

However, the target network in DDQN uses the expected Q-values for the next state by considering the action that gives the maximum Q-value of the next state and updates the Q-values of the online network based on the estimated Q-value from the target network. After a particular number of iterations, the target network parameters will be updated based on the online network. In contrast, the online network will be updated based on the optimizer (e.g., Adam Optimizer). Therefore, the overoptimism problem will be reduced, and the learning phase will be steadier and more reliable. Eq. (18) represents the Y_k^{DDQN} and Eq. (19) is the action-value function Q considering Y^{DDQN} in the DDQN.

$$Y_k^{DDQN} = r + \lambda Q(s', \arg\max_{a'}(Q(s', a'; \theta_k)); \bar{\theta}) \tag{18}$$

$$Q_{k+1}(s_t, a_t, \theta_t) = Q_k(s_t, a_t; \theta_t) + \varphi(Y_k^{DDQN} - Q_k(s_t, a_t, \theta_t)) \tag{19}$$

Dueling Double Deep Q-Network (D3QN). The concept of Dueling Double Deep Q-Network (D3QN) was proposed [57]. The trigger behind proposing D3QN is that for many states estimating the value of each action choice is unnecessary as they are approximately similar and may slow down the learning [57]. The proposed dueling neural network model is structured in a way that estimates two streams. The state-value function $\mathcal{V}$ (how good is to be in specific state s_t) and advantage function A (the importance of each action relatively) are the two output streams from the model but the returned Q is the summation of $\mathcal{V}$ and A [57]. Since the output of the neural network is Q function where each action associated with Q-value, then the advantage function concept can be integrated into different algorithms such as DDQN and SARSA, with their improvements (i.e., soft update, exploration policies, replay memories, and so on).

In this paper, the concept of Advantage function is implemented in DDQN, where this approach of using advantage function A causes identifiability problems, which prevents the recovery of both the $\mathcal{V}$ and A. Therefore, as suggested by [57], the value of A is deduced by its mean to increase the stability level of the optimization. Eq. (20) represents the action–state Q function using in D3QN, considering the value and advantage functions.

$$Q(s,a;\theta,\ddot{\alpha},\ddot{\beta}) = \mathcal{V}(s;\theta,\ddot{\beta}) + \left(A(s,a;\theta,\ddot{\alpha}) - \frac{1}{|\mathcal{A}|}\sum_{a'}^{\mathcal{A}} A((s,a';\theta,\ddot{\alpha})) \right) \tag{20}$$

In Eq. (20), $\ddot{\alpha}$ and $\ddot{\beta}$ represent the parameters of the two aggregated streams A and $\mathcal{V}$, respectively. The state-value function $\mathcal{V}$ can be represented as $\mathcal{V}(s) = \mathbb{E}[Q^*(s,a)]$.

It is worth mentioning that overoptimism cannot be considered a problem for different applications since getting excellent performance is still possible. However, reducing it will significantly stabilize the learning process [56].

5.0.2. Proposed approaches at training time

Algorithm 1 represents the detailed steps done by the BM while training. The weights of both the online and target neural networks are initialized along with replay memory $\mathcal{D}$, which stores the experience data tuples. The BM balances exploration and exploitation behavior during each episode by collecting experience tuples through interactions (lines 7–10).

To ensure the quality of exploring the actions in different states by the agent, it selects the action at different states either randomly with probability ϵ or based on ϵ greedy policy considering the highest $Q-$ value seen so far (line (7)). The default starting value of ϵ equals one, and then it starts to decay over time to enable the learning of the environment by the BM. This behavior of taking actions is done to handle the trade-off between the exploration and exploitation behavior, where the best actions (the highest Q-value) represent the exploitation behavior and the random actions represent the exploration behavior.

The state–action transitions will be stored as tuples in the replay memory $\mathcal{D}$ to be used through the optimization process as experience data (line (12)) to improve the estimation of Q^*. We consider fetching random samples of experiences ρ from $\mathcal{D}$ (line (12)). For each experience tuple i in the subset ρ, the TD-target Y_i is calculated to find the new estimation considering $\bar{\theta}$. This process helps the convergence and stabilization of the learning process discussed previously (experience replay) [57]. TD-target Y_i is calculated based on the used model (DQN, DDQN, or D3QN), as shown in line (13).

Finally, Adam optimizer is used to fit θ to Y^i then applying a soft update to $\bar{\theta}$ after a certain number of steps to reflect the most recent knowledge of the environment (lines 14–15). Stabilizing the learning process can also be achieved through soft updates toward the target network. Eventually, the convergence is achieved where $\theta \sim \bar{\theta}$.

Algorithm 1: Training Process of BM (agent)

```
Input: Environment Simulator
Output: $\theta^*$: The NN parameters for the approximation $Q^*$
1: $\mathcal{D} \leftarrow$ Initialize replay memory with the size $N$.
2: $\theta \leftarrow$ Initialize online network parameters randomly.
3: $\bar{\theta} \leftarrow \theta$ Initialize target network parameters
4: for $episodes = 1 : E$ do
5:     Initialize State
         $s_0 \leftarrow \langle [\check{S}_0, \check{U}_0, \check{a}_0], [R_1 \cdots R_M] \rangle$
6:     for $t = 1 : K$ do
         /** Interaction with the environment**/
7:       Select state update action $a_t$
         $a_t \begin{cases} \text{Random}, & \text{With probability } \epsilon \\ \epsilon\text{-greedy policy}, & \text{otherwise} \end{cases}$
         Find the number of selected transaction $tr$
          and miners $m$ from $a_t$
         Refer to Eq. (7) and (8) respectively
8:       Execute $a_t$ and Observe $s_{t+1}$, and $r_t$
         Refer to Eqs. (9) and (10) for reward calculations
         considering $tr_t$ and $m_t$ from $a_t$
9:       Environment Status
         done $\leftarrow$ Boolean
10:      Insert Tuple of Experience into Replay Memory
         $\mathcal{D} \leftarrow (s_t, a_t, r_t, s_{t+1}, done)$
11:      Update State
         $s_t \leftarrow s_{t+1}$
12:      /**Updating the estimates**/
          Select random minibatch of experience $\rho \subseteq \mathcal{D}$
         $\rho \leftarrow \left\{ (s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)}, done^{(i)}) \right\}_{i=1}^{|\rho|}$
13:      Calculate Q-targets using target network based on the
         chosen approach.
         Refer to Eqs. (15), (18), and (20) for DQN,
         DDQN and D3QN based respectively.
14:      Fit $\theta$ to Target $Y_t$ using Adam Optimizer
15:      Every target steps, update the target network $\bar{\theta}$
         $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$
16:    end for
17: end for
18: return $\theta^* \sim \theta$
```

5.0.3. Proposed approaches in real-time

After running algorithm 1 and reaching convergence, the neural network parameters saved and ready to be used in a real-time scenario for the online decision, where sudden changes might occur to the environment (i.e., increasing the prices of the miners, losing some miners, and so on). The agent is supposed to deal with such changes in the environment toward getting back to the convergence state while considering taking the best long-term actions. Algorithm 2 represents the steps in real-time, where only one neural network pass is executed every time step. At every state, the BM chooses the optimal action that maximizes long-term rewards.

6. Performance evaluation

Healthchain-RL is evaluated through numerical simulations in terms of the following: the reward convergence of the proposed approaches, a comprehensive comparison with different heuristic policies, the adaptation to sudden changes, and the consumed action-time in real-time.

6.1. Experimental setup and design

Based on the definition of the states, the actions, and the environment described earlier, we train our Healthchain-RL on 10^4 training

Algorithm 2: Real-Time BM (agent)

```
Input: The trained NN parameters $\theta^*$
Output: State update
1: $done$ ← Initialize environment status to false.
2: Initialize State $s_t$
3: while !done do
4:     Find $a^*$
       Consider that $a^* \leftarrow tr^*$ and $m^*$
5:     Calculate the corresponding $Q^*(s_t, a_t, \theta)$
6:     Observe $s_{t+1}$, and $r_t$
7:     Environment Status
           done ← Boolean
8:     Insert Tuple of Experience into Replay Memory
           $\mathcal{D} \leftarrow (s_t, a_t, r_t, s_{t+1}, done)$
9:     Update State
           $s_t \leftarrow s_{t+1}$
10: end while
```

Table 2
Optimization parameters.

Parameters	Values
w	0.5 MHz
SNR_d	10 dB
SNR_u	12 dB
q	2
O	0.5 Mb
B	1 Kb
g	1
α, β, γ	0.33

episodes (E), the maximum number of time steps per episode (K) is 10^3, the multiplicative factor (per episode) for epsilon (ϵ) decay is 0.999, the size of the BM queue (Q) and the maximum number of transactions per block (T_r) is 20 transactions, and the maximum number of miners (M) is 10. Four potential reinforcement learning approaches are proposed and compared while building Healthchain-RL, vanilla DQN, DQN with a deeper neural network, DDQN, and D3QN.

It is worth noting that the number of miners identifies the security level since increasing the number of miners will increase security, latency, and cost. On the other hand, increasing the number of transactions will increase the latency and the overall cost. Therefore, to ensure fairness, the weights given to each of the objectives (α, β, γ) are the same, and their summation equals one. The value of weighting factor and other optimization parameters concerning the security, latency, and cost are reported in Table 2.

6.2. Reward convergence

In the training process of Algorithm 1, for the four Q-Learning approaches, we set the hyperparameters' values according to the parameters reported in Table 3. These values were set based on experiments and observations in order to maximize the performance. Fig. 3 represents the reward while training through 10^4 episodes considering vanilla DQN with the neural network of four layers, DQN with a deeper neural network with ten hidden layers instead of four, DDQN, and finally D3QN. The shown rewards values are smoothed over a window of 50 episodes. The figure proves that the proposed approaches were able to reach convergence empirically. All approaches start acting randomly for the first 1000 episodes as ϵ equals one; therefore, the agent explores the environment through random actions and then improves that random policy toward the optimal version smoothly by exponentially decaying the value of ϵ as mentioned previously.

6.3. Performance comparison

Toward verifying the performance of our proposed Healthchain-RL versus a set of Q-Learning and heuristic approaches explained in details below:

Table 3
System parameters.

Parameters	Values
T_r	20
M	10
E	10^4
K	10^3
λ	0.9
ϵ	1, with 0.999 Decay
h_l	4 or 10
Q-Network	when $h_l = 4 \rightarrow 70, 48, 48, 80, 200$
neurons/layer	when $h_l = 10 \rightarrow 70, 48, 48, 48, 48, 48, 48, 48, 48, 80, 200$
φ	3×10^{-4}
$\vert\rho\vert$	48
$\vert\mathcal{D}\vert$	10^5
τ	10^{-3}
Soft update target steps	4

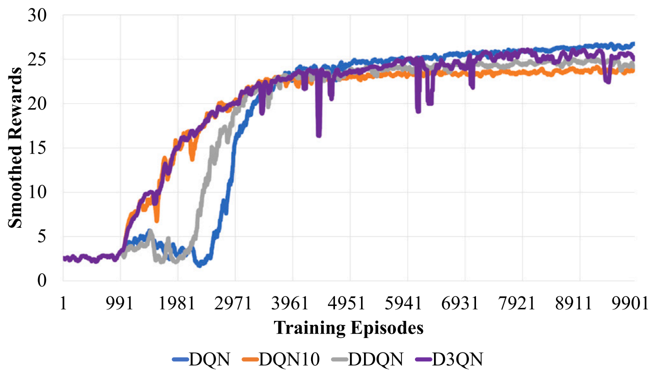

Fig. 3. The Training Process, Reward throughout Episodes for vanilla DQN, DQN with a more in-depth neural network, DDQN, and D3QN.

6.3.0.1. Q-learning policies. As stated earlier, in our implementation of Healthchain-RL, DDQN with the advantage function A concept is used (D3QN). However, we also considered the vanilla DDQN, DQN with four layers, and DQN with ten layers. In [22], the authors mentioned that adding more layers to the model significantly affects the convergence. Nevertheless, it does not add significant value, compared to extra execution time in real-time compared to the four-layer model, as the next sections indicate. Furthermore, two heuristic approaches are considered for comparison, namely (1) The Greedy Approach and (2) Random-Selection Approach (RS).

6.3.0.2. Greedy policy. The Greedy algorithm finds a local-optimal solution at every state; it runs through all actions and selects the best one (best number of transactions and miners) without considering the effect of the decision on the future system reward. This approach was used in [43].

6.3.0.3. Random-selection policy (RS). A baseline approach, which chooses the action of the state uniformly random, precisely like the exploration phase of any DQN approach at the very early episodes. This approach is very lightweight, but it is not as efficient as the techniques proposed.

6.3.0.4. Results and discussion. We have applied a set of experiments to compare the performance of different policies integrated into Healthchain-RL. The assessment methods include running all the policies for 1000 testing episodes. The state changes regularly considering a static environment where no changes occur, and based on the trained model; the action is chosen. The accumulated rewards for all the policies during the testing (real-time) episodes are recorded and plotted, as presented in Fig. 4. The values of Q-Learning approaches are close to each other, unlike Greedy and RS policies. A Zoom-in of the four deep reinforcement approaches shows that the vanilla DQN

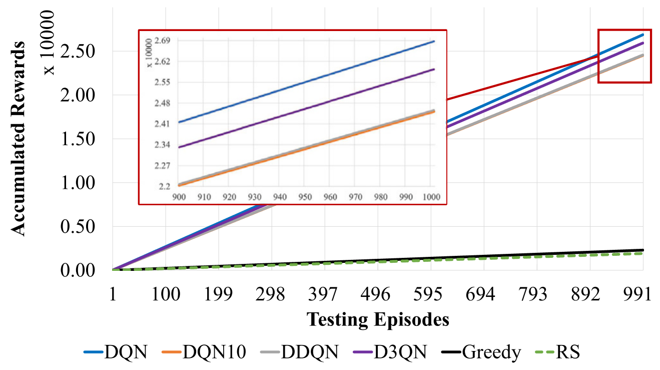

Fig. 4. Accumulated reward during testing (real-time) episodes for all policies.

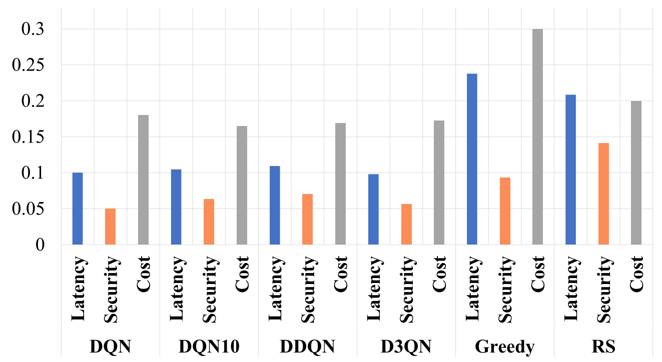

Fig. 5. Zoom-in into the Averaged accumulated reward during testing (real-time) episodes for all the policies per conflicting objective: security, latency and cost.

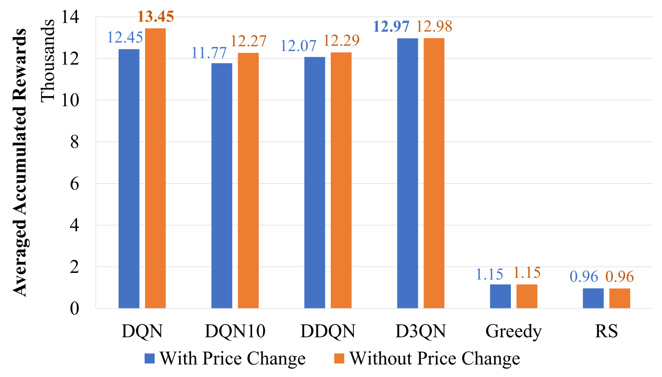

Fig. 6. Averaged accumulated reward during testing (real-time) episodes for all policies with and without changes in the prices of the miners.

could reach the highest reward among other approaches due to the implementation of the online and target networks, where it reaches 26900, followed by D3QN which its primary concern is maximizing the reward and stabilizing performance. The deeper DQN and DDQN have close performance and a lower accumulative reward than other Q-learning approaches.

A zoom-in into the reward considering the three conflicting objectives, latency, security, and cost is presented in Fig. 5. All objectives cannot go beyond their equally assigned weighting factors. Hence, the summation of the three conflicting objectives for any approach should be less than or equal to one. The reward function works toward maximization, while in the proposed multi-objective optimization framework, a trade-off exists. The cost and latency need to be minimized while maximizing the security. Therefore, the reward works toward approaching the optimization of the three conflicting objectives while considering application-level constraints. In Fig. 5, the DQL algorithm and its variation were able to find the policy that addresses and maximizes the long-term trade-off defined by the reward function in Eq. (10), while the assigned penalties make sure the constraints are met. DQL approaches maximize the security while considering the required security level and the selected transactions' urgency requirements. Hence, the latency is the minimum compared to the Greedy and RS approaches. The reputation of the selected miners has a direct effect on the reward function, as introduced previously. The high latency level achieved by the RS approach is due to the uncertainty of choosing an action, while the greedy approach cannot manage the long-term trade-off between security and latency, as it takes the best action considering only the current state. The cost of the Greedy and RS approaches is considered high compared with their performance. The cost objective defined in Eq. (6) is directly proportional to the miners' cost; hence increasing the security causes an increase in the cost objective. Meanwhile, the number of selected transactions affects the latency objective directly. Accordingly, the policy should consider those facts and optimize the reward toward maximizing the security while minimizing latency and cost as much as possible.

Fig. 6 represents the average accumulated rewards for all the approaches where it shows how close the performance of all Q-learning approaches. However, the RS approach's performance is the worst amongst all, and the Greedy approach follows it. Vanilla DQN has the best performance in a static environment, as it reached an average accumulated reward of 13450.

6.4. Convergence of the proposed solutions under drastic changes

Introducing a sudden change to the framework is essential to compare the policies' reactivity to the new system change and illustrate the proposed approaches' adaptability. We decide to consider the miners' prices as the changing parameter since it directly affects all Healthchain-RL conflicting objectives: latency, security, and cost. In this experiment, the number of testing episodes is 1000, and we introduce three drastic changes to the prices. The first change occurred between episodes 100 and 150, where only a slight increase arises for all the miners. Then, in the middle of running 400–500 episodes), another change began where the prices return to their original values. In the last 600–800 episodes, a very high expansion in the price for all the miners happened. Fig. 7 represents the reward per episode and shows how fast the policies can get back to the convergence state. D3QN has the fastest stable and smooth convergence and maintains the highest reward compared to other policies that dropped severely when a change occurs. This behavior of D3QN came from the synchronization of the two networks (online and target network) and the advantage function's implementation, which ensures stabilization during the learning process.

Greedy and Random-Selection (RS) perform poorly and almost constant, as they both do not consider future events. Whereas other Q-Learning approaches can coverage, and their rewards are close to each other. Both vanilla DQN and DQN with ten layers have notable falls when a change occurs in the environment; due to their high sensitivity to the variance, as both the online and target networks use the same maximum operator, unlike D3QN and DDQN [57].

Fig. 8 represents the accumulated rewards during the testing episodes, where D3QN has the highest accumulated reward followed by DQN even though it faced high drops when a drastic change occurred. The difference between DDQN and DQN with ten layers in terms of the accumulated reward is very minimal. However, DDQN has better performance than the deeper DQN. The Random-Selection (RS) policy is the worst as the actions are selected randomly without considering the application-level requirements and the future estimated state. Meanwhile, the Greedy policy ensures getting instantaneous best action reflecting the current state. Therefore, the Greedy approach performs

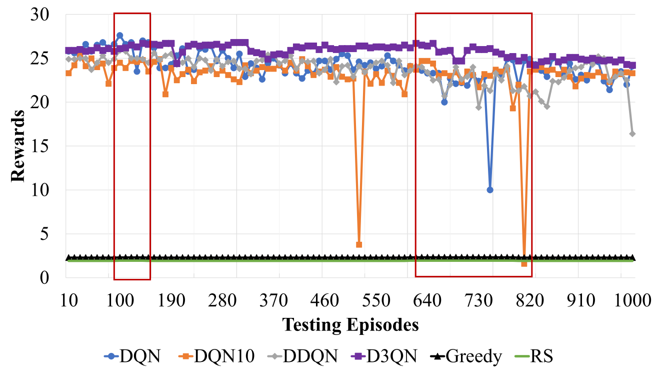

Fig. 7. Convergence comparison — Reward during testing episodes for all policies while changing the prices of the miner at three different episodes.

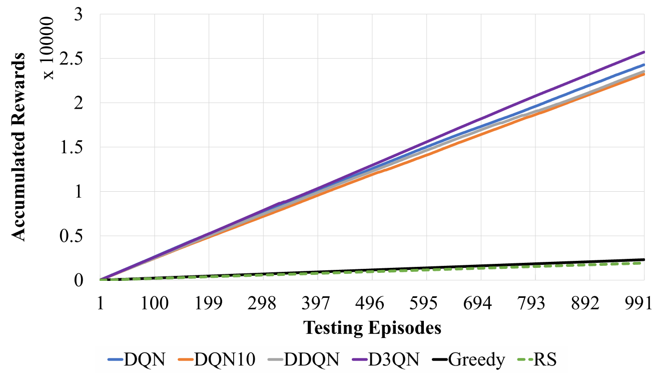

Fig. 8. Accumulated reward during testing episodes for all policies while changing the prices of the miner at three different episodes.

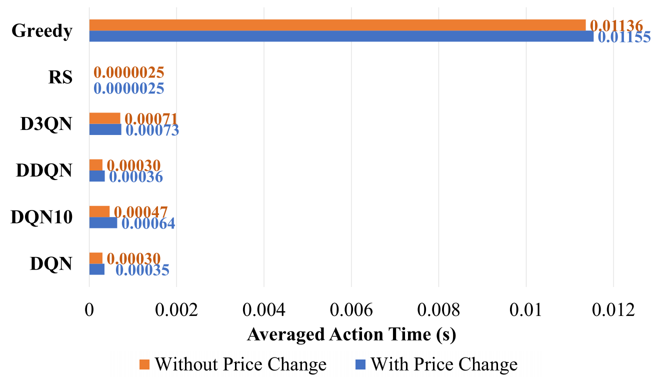

Fig. 9. Action Time while Testing in Seconds (s) for all policies with and without changes in the prices of the miners.

slightly better than the RS, but it cannot compete with Q-learning approaches.

The comparison of the Q-Network approaches is conducted under the same conditions as Table 3 states. It worth mentioning that fine-tuning these conditions for a specific approach might lead to earlier convergence and stability. Each approach has characteristics and limitations, and only one of the proposed approaches can be used based on the system administrator's preference.

6.5. Action-time

In healthcare applications, decision-making time is considered an essential factor, particularly for emergency scenarios. Fig. 9 represents the averaged time needed by all the techniques to decide on an action for a particular state during the 1000 testing episodes, with and without introducing a sudden change to the system.

The Greedy approach consumes the highest time as it performs a single pass and searches through all possible actions to find the best action considering only the instantaneous reward. The Greedy approach may be deemed unreliable in healthcare systems as the time required to make online decisions is critical. In contrast, the Random-Selection (RS) approach runs extremely fast. It randomly chooses an action without considering any of the rewards, subsequent state transmission, entity-level requirements, or the number of attempts needed to effectively solve the environment, leading to low accumulative reward over time. Therefore, the RS approach is not a desirable solution in healthcare systems since taking the right action considering the future and simultaneous reward is significant.

The performance of our proposed trained approaches is close to each other, mainly when no sudden changes occur in the environment. However, when introducing a sudden change in the miners' prices, the proposed approaches need to accommodate the sudden change while considering the accumulated future reward. As presented in Fig. 9, D3QN takes more time than others due to the extra computation of the advantage function explained before, which ensures a relatively stable convergence during drastic system changes. The vanilla DQN can be the fastest approach in both the static system and when introducing drastic changes. However, a wrong action has a higher probability in DQN, as discussed previously in Section 6.4. Therefore, assessing the severity of the situation depends on the system's characteristics.

Based on the results stated, DQN, DDQN, and D3QN can be used in Healthchain-RL, considering their characteristics, limitations, and overall system requirement.

The following tips can help the system administrator while selecting the approach to be deployed in Healthchain-RL, considering that all discussed approaches in this paper follow the same environmental parameters and setup:

- D3QN is a preferable approach when having an unstable environment, where changes occur frequently. However, the time needed to make an action using D3QN is slightly higher than other approaches, as indicated in Fig. 9.
- DQN should be the best choice when the time has a higher priority, as the time to select an action compared to other approaches is the shortest, as shown in Fig. 9. However, a sudden change in the system reflects significantly in the reward, where an incorrect action is taken. Such a wrong action may have severe implications in healthcare systems.

7. Conclusion

In conclusion, we propose Healthchain-RL, which ensures secure, adaptive, flexible communication between several heterogeneous entities to exchange medical data and patient information. A multi-objective optimization that addresses the trade-off between security, latency, and cost of the overall systems is resolved by introducing an intelligent Blockchain Manager (BM). DQN, DDQN, and D3QN are reinforcement decision-making algorithms used to implement the BM based on the healthcare application requirements and the platform's stability. Only one approach can be selected at a time to implement the intelligent Blockchain manager. The BM holds the role of the agent in the proposed RL approaches, as it decides the number of transactions per block and the needed number of miners for verification, considering the following: the multi-objective optimization, the reputation of the miners, and the entities requirements (i.e., security and urgency levels of each submitted transaction). DQN approach is a practical approach, especially when the system has minimal changes, while D3QN manages frequent fluctuations, and it converges smoothly in real-time, as the experimental results prove. The simulation results show that reinforcement learning approaches outperform the Greedy and the Random-Selection (RS) approaches. The proposed models reach a sub-optimal solution considering the Blockchain's temporal aspects and future behavior, unlike heuristic approaches taking an immediate decision triggering a drop in the accumulative reward. Besides, the Greedy approach suffers from computational complexity compared to DQN, DDQN, and D3QN, which deems this approach impractical, especially in real-time and critical scenarios.

CRediT authorship contribution statement

Abeer Z. Al-Marridi: Conceptualization, Methodology, Investigation, Software, Validation, Resources, Writing - original draft, Funding acquisition. **Amr Mohamed:** Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration and Funding acquisition. **Aiman Erbad:** Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration and Fund acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work of Abeer Z. Al-Marridi is supported by GSRA grant GSRA5-1-0326-18026 and the work of Amr Mohamed was partially covered by NPRP grant NPRP12S-0305-190231 from the Qatar National Research Fund (a member of Qatar Foundation). The statement therein are the sole responsibility of the authors

References

[1] Chronic diseases in America, 2019, https://www.cdc.gov/chronicdisease/resources/infographic/chronic-diseases.htm.

[2] A. Al-Marridi, A. Mohamed, A. Erbad, Chapter 1 - AI-based techniques on edge devices to optimize energy efficiency in m-Health applications, in: A. Mohamed (Ed.), Energy Efficiency of Medical Devices and Healthcare Applications, Academic Press, 2020, pp. 1–23, https://www.sciencedirect.com/science/article/pii/B9780128190456000017.

[3] T. Alladi, V. Chamola, R.M. Parizi, K.-K.R. Choo, Blockchain applications for industry 4.0 and industrial IoT: A review, IEEE Access 7 (2019) 176935–176951.

[4] L.-e. Wang, Y. Bai, Q. Jiang, V.C. Leung, W. Cai, X. Li, Beh-raft-chain: A behavior-based fast blockchain protocol for complex networks, IEEE Trans. Netw. Sci. Eng. (2020).

[5] Z. Xiong, Y. Zhang, N.C. Luong, D. Niyato, P. Wang, N. Guizani, The best of both worlds: A general architecture for data management in blockchain-enabled Internet-of-Things, IEEE Netw. 34 (1) (2020) 166–173.

[6] D. Zhao, Application and development trend of blockchain in the financial field, in: International Conference on Application of Intelligent Systems in Multi-Modal Information Analytics, Springer, 2020, pp. 558–564.

[7] A. Kamišalić, M. Turkanović, S. Mrdović, M. Heričko, A preliminary review of blockchain-based solutions in higher education, Commun. Comput. Inf. Sci. 1011 (2019) 114–124.

[8] L. Bach, B. Mihaljevic, M. Zagar, Comparative analysis of blockchain consensus algorithms, in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, IEEE, 2018, pp. 1545–1550.

[9] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the Internet of Things, IEEE Access 4 (2016) 2292–2303.

[10] ReportLinker, Global blockchain technology market in the healthcare industry, 2018–2022, 2019, https://www.globenewswire.com/news-release/2019/10/17/1931178/0/en/Global-Blockchain-Technology-Market-in-the-Healthcare-Industry-2018-2022.html.

[11] A. Hasselgren, K. Kralevska, D. Gligoroski, S.A. Pedersen, A. Faxvaag, Blockchain in healthcare and health sciences—A scoping review, Intl. J. Med. Inform. 134 (2020) 104040.

[12] M. Liu, Y. Teng, F.R. Yu, V.C.M. Leung, M. Song, Deep reinforcement learning based performance optimization in blockchain-enabled internet of vehicle, in: ICC 2019 - 2019 IEEE International Conference on Communications, ICC, 2019, pp. 1–6.

[13] M. Liu, F.R. Yu, Y. Teng, V.C. Leung, M. Song, Performance optimization for blockchain-enabled industrial Internet of Things (IIoT) systems: A deep reinforcement learning approach, IEEE Trans. Ind. Inf. 15 (6) (2019) 3559–3570.

[14] S. Tanwar, K. Parekh, R. Evans, Blockchain-based electronic healthcare record system for healthcare 4.0 applications, J. Inform. Secur. Appl. 50 (2020) 102407.

[15] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, Y. Wang, Artificial intelligence in healthcare: past, present and future, Stroke Vascular Neurol. 2 (4) (2017) 230–243.

[16] Y. Li, X. Liang, Z. Hu, E.P. Xing, Hybrid retrieval-generation reinforced agent for medical image report generation, in: Advances in Neural Information Processing Systems, 2018, pp. 1530–1540.

[17] Y. Ling, S.A. Hasan, V. Datla, A. Qadir, K. Lee, J. Liu, O. Farri, Diagnostic inferencing via improving clinical concept extraction with deep reinforcement learning: A preliminary study, in: Machine Learning for Healthcare Conference, 2017, pp. 271–285.

[18] S.M. Shortreed, E. Laber, D.J. Lizotte, T.S. Stroup, J. Pineau, S.A. Murphy, Informing sequential clinical decision-making through reinforcement learning: an empirical study, Mach. Learn. 84 (1–2) (2011) 109–136.

[19] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT press, 2018.

[20] H. Fan, L. Zhu, C. Yao, J. Guo, X. Lu, Deep reinforcement learning for energy efficiency optimization in wireless networks, in: 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis, ICCCBDA, 2019, pp. 465–471.

[21] H. Lee, J. Kim, J. Lee, Resource allocation in wireless networks with deep reinforcement learning: A circumstance-independent approach, IEEE Syst. J. 14 (2) (2020) 2589–2592.

[22] D. Zhang, F.R. Yu, R. Yang, Blockchain-based distributed software-defined vehicular networks: A dueling deep Q-learning approach, IEEE Trans. Cogn. Commun. Netw. 5 (4) (2019) 1086–1100.

[23] Y. Dai, D. Xu, K. Zhang, S. Maharjan, Y. Zhang, Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks, IEEE Trans. Veh. Technol. 69 (4) (2020) 4312–4324.

[24] C.H. Liu, Q. Lin, S. Wen, Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning, IEEE Trans. Ind. Inf. 15 (6) (2019) 3516–3526.

[25] Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He, Y. Zhang, Blockchain and deep reinforcement learning empowered intelligent 5G beyond, IEEE Netw. 33 (3) (2019) 10–17.

[26] N. Mhaisen, N. Fetais, A. Erbad, A. Mohamed, M. Guizani, To chain or not to chain: A reinforcement learning approach for blockchain-enabled IoT monitoring applications, Future Gener. Comput. Syst. 111 (2020) 39–51.

[27] T.T. Anh, N.C. Luong, Z. Xiong, D. Niyato, D.I. Kim, Joint time scheduling and transaction fee selection in blockchain-based RF-powered backscatter cognitive radio network, 2020, ArXiv preprint arXiv:2001.03336.

[28] M. Mettler, Blockchain technology in healthcare: The revolution starts here, in: 2016 IEEE 18th International Conference on E-Health Networking, Applications and Services, Healthcom, 2016, pp. 1–3.

[29] Q. Xia, E.B. Sifah, K.O. Asamoah, J. Gao, X. Du, M. Guizani, MeDShare: Trust-less medical data sharing among cloud service providers via blockchain, IEEE Access 5 (2017) 14757–14767.

[30] L. Chen, W.-K. Lee, C.-C. Chang, K.-K.R. Choo, N. Zhang, Blockchain based searchable encryption for electronic health record sharing, Future Gener. Comput. Syst. 95 (2019) 420–429.

[31] S. Wang, J. Wang, X. Wang, T. Qiu, Y. Yuan, L. Ouyang, Y. Guo, F. Wang, Blockchain-powered parallel healthcare systems based on the ACP approach, IEEE Trans. Comput. Soc. Syst. 5 (4) (2018) 942–950.

[32] A. Zhang, X. Lin, Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain, J. Med. Syst. 42 (8) (2018) 140.

[33] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, J. He, Blochie: a blockchain-based platform for healthcare information exchange, in: 2018 Ieee International Conference on Smart Computing, Smartcomp, IEEE, 2018, pp. 49–56.

[34] B. Shen, J. Guo, Y. Yang, Medchain: efficient healthcare data sharing via blockchain, Appl. Sci. 9 (6) (2019) 1207.

[35] A. Roehrs, C.A. da Costa, R. da Rosa Righi, OmniPHR: A distributed architecture model to integrate personal health records, J. Biomed. Inform. 71 (2017) 70–81.

[36] A. Al Omar, M.Z.A. Bhuiyan, A. Basu, S. Kiyomoto, M.S. Rahman, Privacy-friendly platform for healthcare data in cloud based on blockchain environment, Future Gener. Comput. Syst. 95 (2019) 511–521.

[37] X. Liang, J. Zhao, S. Shetty, J. Liu, D. Li, Integrating blockchain for data sharing and collaboration in mobile healthcare applications, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC, IEEE, 2017, pp. 1–5.

[38] M. Samaniego, R. Deters, Hosting virtual iot resources on edge-hosts with blockchain, in: 2016 IEEE International Conference on Computer and Information Technology, CIT, IEEE, 2016, pp. 116–119.

[39] K. Yaeger, M. Martini, J. Rasouli, A. Costa, Emerging blockchain technology solutions for modern healthcare infrastructure, J. Sci. Innov. Med. 2 (1) (2019).

[40] R.S. Sutton, A.G. Barto, R.J. Williams, Reinforcement learning is direct adaptive optimal control, IEEE Control Syst. Mag. 12 (2) (1992) 19–22.

[41] J. Yang, S. He, Y. Xu, L. Chen, J. Ren, A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks, Sensors 19 (4) (2019) 970.
[42] T. Wang, S.C. Liew, S. Zhang, When blockchain meets AI: Optimal mining strategy achieved by machine learning, 2019, ArXiv preprint arXiv:1911.12942.
[43] A.A. Abdellatif, A.Z. Al-Marridi, A. Mohamed, A. Erbad, C.F. Chiasserini, A. Refaey, ssHealth: Toward secure, blockchain-enabled healthcare systems, IEEE Netw. (2020) 1–8.
[44] A.Z. Al-Marridi, A. Mohamed, A. Erbad, Convolutional autoencoder approach for EEG compression and reconstruction in m-health systems, in: 2018 14th International Wireless Communications Mobile Computing Conference, IWCMC, 2018, pp. 370–375.
[45] A.A. Abdellatif, L. Samara, A. Mohamed, A. Erbad, C.F. Chiasserini, M. Guizani, M.D. O'Connor, J. Laughton, MEdge-Chain: Leveraging edge computing and blockchain for efficient medical data exchange, IEEE Internet Things J. (2021).
[46] J. Kang, Z. Xiong, D. Niyato, D. Ye, D.I. Kim, J. Zhao, Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory, IEEE Trans. Veh. Technol. 68 (3) (2019) 2906–2920.
[47] C. Huang, Z. Wang, H. Chen, Q. Hu, Q. Zhang, W. Wang, X. Guan, Repchain: A reputation based secure, fast and high incentive blockchain system via sharding, 2019, ArXiv preprint arXiv:1901.05741.
[48] Z. Zhao, Y. Liu, A blockchain based identity management system considering reputation, in: 2019 2nd International Conference on Information Systems and Computer Aided Education, ICISCAE, 2019, pp. 32–36.
[49] L.-A. Hîrţan, C. Dobre, H. González-Vélez, Blockchain-based reputation for intelligent transportation systems, Sensors 20 (3) (2020) 791.
[50] E.K. Wang, Z. Liang, C.-M. Chen, S. Kumari, M.K. Khan, PoRX: A reputation incentive scheme for blockchain consensus of IIoT, Future Gener. Comput. Syst. 102 (2020) 140–151.
[51] J. Bou abdo, R. El Sibai, K. Kambhampaty, J. Demerjian, Permissionless reputation-based consensus algorithm for blockchain, Internet Technol. Lett. (2020) e151.
[52] G. Fortino, F. Messina, D. Rosaci, G.M. Sarné, A reputation capital and blockchain-based model to support group formation processes in the Internet of Things, in: 2019 6th International Conference on Control, Decision and Information Technologies, CoDIT, IEEE, 2019, pp. 888–893.
[53] A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decis. Support Syst. 43 (2) (2007) 618–644, Emerging Issues in Collaborative Commerce, http://www.sciencedirect.com/science/article/pii/S0167923605000849.
[54] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
[55] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, 2013, ArXiv preprint arXiv:1312.5602.
[56] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double q-learning, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[57] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, N. Freitas, Dueling network architectures for deep reinforcement learning, in: M.F. Balcan, K.Q. Weinberger (Eds.), in: Proceedings of Machine Learning Research, vol. 48, PMLR, New York, New York, USA, 2016, pp. 1995–2003, http://proceedings.mlr.press/v48/wangf16.html.

Abeer Z. Al-Marridi is currently a Ph.D. student in the Computer Science and Engineering Department at Qatar University. She is awarded as a Graduate Student Research Award from Qatar National Research Fund (QNRF). Her research interests include Deep Learning, Reinforcement Learning, Edge Computing, Blockchain, Resource Optimization for Wireless Sensor Networks, and Smart Health Systems. Abeer holds a master's degree in computing from Qatar University, where her thesis was about Optimal Resource Allocation Using Deep Learning-Based Adaptive Compression for m-Health Applications. She is a member of Arab Women in Computing (ArabWIC).

Amr Mohamed (S' 00, M' 06, SM' 14) received his M.S. and Ph.D. in electrical and computer engineering from the University of British Columbia, Vancouver, Canada, in 2001, and 2006 respectively. He has worked as an advisory IT specialist in IBM Innovation Centre in Vancouver from 1998 to 2007, taking a leadership role in systems development for vertical industries.

He is currently a professor in the college of engineering at Qatar University and the director of the Cisco Regional Academy. He has over 25 years of experience in wireless networking research and industrial systems development. He holds 3 awards from IBM Canada for his achievements and leadership, and 4 best paper awards from IEEE conferences. His research interests include wireless networking, and edge computing for IoT applications. Dr. Amr Mohamed has authored or co-authored over 200 refereed journal and conference papers, textbooks, and book chapters in reputable international journals, and conferences. He is serving as a technical editor for two international journals and has served as a technical program committee (TPC) co-chair for many IEEE conferences and workshops.

Aiman Erbad is an associate Professor at the 2Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Qatar. Dr. Erbad obtained a Ph.D. in Computer Science from the University of British Columbia (Canada), and a Master of Computer Science in Embedded Systems and Robotics from the University of Essex (UK). Dr. Erbad received the Platinum award from H.H. The Emir Sheikh Tamim bin Hamad Al Thani at the Education Excellence Day 2013 (Ph.D. category). Dr. Erbad research interests span cloud computing, multimedia systems and networking, and his research is published in reputed international conferences and journals.