Journal of Network and Computer Applications 224 (2024) 103834

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Research paper

# Optimized blockchain-based healthcare framework empowered by mixed multi-agent reinforcement learning

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:*
POMDP
Cooperative-competitive
MARL
Blockchain
Optimization
Resource allocation
Healthcare

ABSTRACT

The world has recently witnessed the devastating impact of a global pandemic, where countless lives were tragically lost due to delayed abnormal disease detection and the lack of seamless coordination among healthcare organizations such as hospitals, medical labs, and pharmacies. Addressing the heterogeneity of healthcare entities and the extensive volume of medical-related data they possess gives rise to significant challenges in establishing trust, security, privacy, and low-latency connections. To this end, this paper introduces the IP-HealthChain framework, a comprehensive healthcare solution built upon blockchain technology to foster trust, decentralization, and secure data exchange while integrating intelligence within its ecosystem. In this work, we propose a blockchain-based optimization model that leverages the available computational resources to enhance each participant's latency, security, and cost. We achieve this by developing an online intelligent Deep Reinforcement Learning decision-making algorithm tailored to address the mixed cooperative-competitive nature inherent in decision-making processes within the shared blockchain network. The optimization model is formulated as a Mixed Decentralized Partially Observable Markov Decision Process and efficiently solved using state-of-the-art Deep Multi-Agent Q-Learning techniques. Through this approach, we enable indirect communication among participants and ensure the processing of low-urgency data, thereby promoting an inclusive and responsive healthcare ecosystem. Extensive simulations are conducted to evaluate the performance of our proposed solution. The results demonstrate significant advantages over traditional heuristics in terms of decision-making speed, resource utilization, latency and cost minimization, security maximization, and adherence to stringent Quality of Service requirements.

## 1. Introduction

The utilization of the Internet of Medical Things (IoMT) in the development of intelligent healthcare systems has opened up vast opportunities for collecting and harnessing large amounts of data, necessitating efficient processing, transmission, and analysis. However, the current centralized healthcare infrastructure poses significant challenges and limitations that affect multiple stakeholders (i.e.,Healthcare related organizations). These challenges encompass a wide range of issues, such as data duplication, fragmented data repositories, limited analysis capabilities, delayed identification of abnormal conditions, protracted communication delays, concerns over data security and privacy, vulnerability to single points of failure, the heterogeneous nature of medical data generated by diverse devices, and monopolistic control over information flow (Al-Marridi et al., 2021a).

Furthermore, the escalating frequency of healthcare data breaches, encompassing personal patient information, hospital data, and the ever-present menace of counterfeit drugs, has further eroded trust and impeded the seamless exchange of information among healthcare organizations. These impediments underscore the imperative to replace the traditional healthcare system with an innovative, secure, efficient, unified, and interconnected framework. This is crucial, especially in times of crisis, where quick decision-making is imperative (Al-Marridi et al., 2021a).

The emerging technologies of blockchain, edge computing, and artificial intelligence (AI) hold immense potential to revolutionize healthcare by addressing critical challenges and improving patient outcomes. Blockchain's decentralized and secure nature can facilitate secure and transparent data sharing, while edge computing enables real-time data processing and analysis. AI, particularly reinforcement learning, provides a robust framework for optimizing decision-making in complex healthcare environments. The synergistic combination of these technologies can transform healthcare delivery, enhancing efficiency, reducing costs, and improving patient care (Al-Marridi et al., 2021a).

* Corresponding author.
*E-mail address:* abeer_almarridi@outlook.com (A.Z. Al-Marridi).

https://doi.org/10.1016/j.jnca.2024.103834
Received 16 July 2023; Received in revised form 21 January 2024; Accepted 23 January 2024
Available online 26 January 2024

Blockchain technology, introduced in 2008 as a secure distributed ledger utilizing peer-to-peer (P2P) networking, has proven its effectiveness in maintaining data consistency and integrity (Nakamoto, 2008). Its decentralized nature fosters interconnectivity among participants, making it a versatile tool across various sectors by providing distributed data storage and processing capabilities. Blockchain can be categorized into three main types: public, private, and consortium blockchains. Public blockchains are fully decentralized, distributed, and open to anyone to participate. Private blockchains are managed by a single organization and permissioned to specific users. Consortium blockchains are a hybrid of public and private blockchains, governed by a group of authorized organizations (Gupta et al., 2021). Vantage Market Research anticipates significant growth in blockchain utilization within the healthcare market, projecting a Compound Annual Growth Rate (CAGR) of 61.3% until 2028 (Research, 2022). The immutability and transparency inherent in distributed ledger technology are identified as primary drivers for blockchain's growth in healthcare (Research, 2022). Notably, the elimination of third-party intermediaries fosters a trustful environment (Hussien et al., 2019).

Blockchain's performance in healthcare systems surpasses other technologies, particularly cloud-based databases. Its security is highlighted by its resilience against hacking, even for well-equipped attackers, in contrast to the vulnerability of cloud-based databases. Moreover, the transparency of blockchain transactions, governed by defined privileges in smart contracts, minimizes the risk of fraud and corruption. The decentralized nature of blockchain renders it resistant to censorship and interference, a significant advantage over the typically centralized cloud-based databases (Kumari et al., 2020). However, blockchain application in healthcare frameworks presents unique challenges, notably in latency and security. Balancing the processing, storage, and secure access to massive healthcare data with acceptable latency is crucial. Furthermore, considering conflicting objectives such as security, latency, and cost becomes paramount when communicating with diverse healthcare organizations with distinct privileges and visions. Addressing these challenges is critical when implementing blockchain-empowered healthcare systems. AI has demonstrated its proficiency in addressing various medical tasks, leveraging its capacity to learn from extensive datasets (Jiang et al., 2017) effectively. Within AI, Reinforcement Learning (RL) stands out for its ability to facilitate intelligent decision-making in non-deterministic environments to maximize the system's reward over time (Sutton and Barto, 2018). Single-agent RL (SARL) represents the default RL setting involving a single agent. The integration of SARL into blockchain exhibits promise by eliminating the need for executing a complex traditional multi-objective optimization problem at every step, thereby significantly impacting average complexity (Al-Marridi et al., 2021a). However, real-world healthcare systems involve multiple participants. Hence, the SARL solution will not be effective. This aligns more efficiently with the decentralized nature of blockchain. This paper introduces a pioneering blockchain-based healthcare framework named IP-HealthChain, employing Mixed Cooperative-Competitive Deep Multi-Agent Reinforcement Learning (DMARL) at the blockchain participant's level. IP-HealthChain's objective is to provide optimal Quality of Service (QoS) by dynamically adjusting blockchain configurations. Blockchain Intelligent Participants (IPs), representing diverse healthcare entities like hospitals, clinics, pharmacies, research labs, insurance companies, and the Ministry of Public Health (MOPH), facilitating the exchange of medical-related data with external systems. IP-HealthChain adeptly leverages the computational resources of the blockchain while optimizing the trade-off between latency, security, and cost. This optimization takes into account the heterogeneous requirements of participating IPs. The paper delves into various design approaches for blockchain participants.[1] Our main contributions can be summarized as follows.

- Propose a novel healthcare framework, IP-HealthChain, a smart blockchain-based healthcare system that maps the transaction characteristics (i.e., security level, urgency level, and queuing time) to blockchain configuration. The framework adeptly leverages extant computational resources, achieving optimal performance at the individual IP level and concurrently accommodating the inherent heterogeneity associated with each specific IP.
- Formulate a Multi-Objective Optimization Problem (MOOP) that describes the whole system and aims at obtaining, for each IP, (i) the optimal data compression decision, (ii) the selected number of transactions, and (iii) the number of verifiers to address the trade-off between three conflicting objectives: latency, security, and cost. Meanwhile, the MOOP considers the optimization of the available computational resources to the maximum.
- The problem is reformulated in terms of Multi-Agent theory, where the IPs in the system will be defined along with their observations, actions, and reward functions. Formulate the Mixed Decentralized Partially Observable Markov Decision Process (Dec-POMDP) of IP-HealthChain and facilitate an implicit communication between the IPs while considering the temporal aspects of the healthcare-based blockchain.
- Propose a distributed DMARL algorithm, namely, Multi-Agent Dueling Double Deep Q-Network (MAD3QN), which handles the heterogeneity of the IPs in the system and looks for the optimal policy that maximizes the rewards considering the cooperative-competitive natural of the IPs. The proposed algorithm utilizes the blockchain computational resources, solves the MOOP in real-time, and prevents the starvation problem of low-level urgency transactions by considering various application-level constraints and reacting to system dynamics efficiently.

The rest of the paper is organized as follows: Section 2 explores related works. Section 3 introduces the IP-HealthChain framework. Section 4 presents the MARL formulation. Section 5 provides an empirical performance comparison with other heuristics. Finally, we provide a concrete application through a case study in Section 6 and conclude in Section 7.

## 2. Related works

Blockchain technology has garnered substantial attention across diverse industries, including healthcare, owing to its inherent promises. In Roehrs et al. (2019), an interoperability model for personal health records was proposed. The study evaluated the structuring of semantic interoperability and integration of different health standards. However, since the system depended on Kafka, which only offered crash fault tolerance, it was not resistant to byzantine failures. The authors in Tanwar et al. (2020b) proposed an access control algorithm to enhance healthcare data accessibility. However, limitations include centralized user key generation and vulnerability to security issues with the RAFT consensus protocol. A cooperative healthcare data-sharing architecture using Ethereum blockchain and edge-cloud computing was introduced in Nguyen et al. (2021). Their scheme includes privacy-aware data offloading, secure data exchange through smart contracts, and efficient user authentication. The evaluation demonstrated low smart contract operational costs and ensured system security, affirming the feasibility of the healthcare applications' approach. However, potential scalability issues and concerns about trustworthiness arise due to the lack of a certifying entity for information authenticity. Another work by Uppal et al. (2021) proposed Careblocks, a blockchain-based framework for secure and efficient healthcare data sharing. However, Careblocks faces challenges in its multi-chain architecture complexity, causing an increase in operational costs and administrative burden, in addition to weakness in handling emergency cases, increasing the possibility of losing patients' lives. In Demirbaga and Aujla (2022), the authors address trust issues raised in Nguyen et al. (2021) by employing a zero-knowledge approach to prevent data interconnections and restrict access to unverified individuals. They developed a flexible computing

[1] The rules, eligibility, authority, and the participant's priority level of performing actions are presented and signed by all the participants in the blockchain smart contract

platform with a monitoring structure for big data statistics and a blockchain-based information storage mechanism. This approach combines blockchain technology and big data platforms to assess, secure, and enable validated access to information from IoT-enabled devices, effectively addressing challenges in big data analysis and ensuring patient confidentiality. Regardless the benefits of zero-knowledge approach, concerns over computational overhead, implementation complexity, and privacy aggregation need to be addressed. In Uppal et al. (2023), the authors proposed HealthDote, a blockchain-based model for continuous health monitoring using the InterPlanetary File System (IPFS). It allows patients to share their health data with healthcare providers securely and efficiently. HealthDote used smart contracts to manage access to health data and ensure that only authorized users can view or modify patient records. IPFS was used to store health data in a distributed and tamper-proof manner. HealthDote addressed the limitations of Careblocks (Uppal et al., 2021) by incorporating an alarm feature for emergency cases and enhancing resistance to censorship and data breaches. However, the usage of IPFS introduced complexity and scalability issues. The authors in Dubey et al. (2023) introduced HECC-ABE, a novel blockchain-based IoT healthcare data storage system that utilizes hybrid cryptography schemes and a hybrid meta-heuristic algorithm for key optimization. While it presents promising features, it may suffer from scalability issues and complexity due to the combination of symmetric and asymmetric encryption.

RL has emerged as a significant subfield within the area of Machine Learning (ML), primarily addressing problems formulated as Markov Decision Processes (MDPs) and their various extensions (Sutton et al., 1992). The fundamental goal of RL revolves around the maximization of cumulative rewards over time through the discovery of an optimal state–action policy. Within the healthcare sector, RL has found applicability in diverse tasks, including the extraction of clinical concepts, generation of medical image reports, and optimization of treatment policies concerning chronic illnesses, as evidenced in Li et al. (2018), Ling et al. (2017), Zhu et al. (2020). In healthcare systems, participants often cooperate to maximize the framework's overall performance while exhibiting competitive tendencies to maximize their benefits. The decision-making phase within healthcare systems is paramount, considering factors such as the current and future performances, other participants (i.e., agents) in the system, and the utilization of the available computational resources.

RL and blockchain technology have the potential to revolutionize healthcare by providing new and innovative solutions for a wide range of challenges (Tanwar et al., 2020a). The authors in Hathaliya et al. (2019) propose blockchain-based healthcare architecture considering the possibility of deploying different machine learning algorithms to solve application-level problems. However, the architecture is restricted by the range of one organization. In this regard, several resource optimization methods within the healthcare domain have proposed SARL techniques capable of accommodating diverse application-specific and system-level requirements, including time, block offloading, and computational resource allocation, as demonstrated in Li et al. (2020b), Mohammed et al. (2020), He et al. (2020), Lin et al. (2021), Alam et al. (2023). In Tanwar et al. (2020c), a patient-centric approach was proposed using a permissioned blockchain, where the analysis aligned with stakeholder needs, identifying performance optimization strategies. However, sharing patient data entails risks such as data security, personal information leakage, and potential falsification. In Liu and Li (2022), the authors proposed a permissioned blockchain and deep reinforcement learning (DRL)-powered Healthcare Internet of Things (H-IoT) system to address security and energy efficiency issues in the H-IoT. DRL facilitated computing offloading tasks in mobile edge computing (MEC). Additionally, the system adopts an energy harvesting method to improve energy efficiency further. Despite offering promising solutions, the system faced limitations, including the scalability and complexity of the multi-chain architecture. In Abdellatif et al. (2020), the authors highlighted the significance of integrating blockchain technology within healthcare systems, considering the heterogeneous nature of participating organizations. Previously in Al-Marridi et al. (2021a), we optimized the complex trade-off between cost, latency, and security, considering the overall system's future performance and the temporal aspects of the transactions. However, the existence of a centralized manager responsible for coordinating transactions at a specific time, when all other participants send their transactions, deviates from the decentralized nature inherent in blockchain technology and subsequently introduces potential security vulnerabilities. Consequently, preserving the privacy of information exchanged amongst participants becomes a formidable challenge. Moreover, each participant's diverse requirements and qualifications add further complications for SARL solution, prompting the introduction of MARL techniques. MARL methodologies have been harnessed effectively within resource allocation scenarios across various sectors, as evidenced by the contributions in Nguyen et al. (2021), Peng and Shen (2020), Wu et al. (2021), Li et al. (2020a). The integration of MARL techniques holds considerable promise in addressing the scalability limitations of blockchain technology, as highlighted by Arduin and Opatowski (2018). The taxonomy of MARL algorithms has been comprehensively discussed, considering the nature of tasks and agent awareness concerning other agents (Busoniu et al., 2008). MARL techniques have been classified into three primary taxonomies: fully cooperative, fully competitive, or mixed cooperative-competitive settings. In the fully cooperative setting, all the agents cooperate to maximize the long-term reward (i.e., All agents receive the same reward). Conversely, competitive settings necessitate that the cumulative rewards of all agents sum to zero as they compete directly with one another. In the mixed cooperative-competitive settings, the agents must exhibit cooperative and competitive behavior to maximize the long-term reward (Lowe et al., 2017). Thus, MARL is a promising solution to address distributed decision-making at the blockchain participants' level. Nevertheless, the direct integration of SARL techniques in which multiple agents function as independent learners does not guarantee convergence towards optimal solutions, owing to the non-stationarity of the environment from each agent's perspective, a well-known challenge in MARL.

This paper presents a secure, adaptive, and intelligent blockchain-based. The proposed MARL approach ensures preserving the stationary aspect of the environment and enhances the overall system's performance.

## 3. System model

This section introduces the structure of a blockchain-based Healthcare framework (IP-HealthChain), where Fig. 1 illustrates the overall system architecture. IP-HealthChain uses a modified consortium medical Blockchain technology.[2] Moreover, intelligence was introduced to the blockchain participants (i.e., healthcare-related organizations) to manipulate the blockchain configurations wisely and optimize system performance Fig. 1 shows the system comprises local and blockchain networks. At the local network, each blockchain participant has medical-related transactions to be shared at the blockchain network. The source of those transactions depends on the participant's nature. Each participant is responsible for preparing the data locally and annotating them in terms of urgency and security. Given the transmission of extensive real-time medical data at the local network, data compression is crucial to optimize network resources by minimizing transmission energy and cost, all while ensuring minimal distortion during data reconstruction (Al-Marridi et al., 2018; Al-Marridi et al., 2021b; AlMarridi et al., 2020).

[2] The participants and verifiers in a consortium blockchain are known and trusted, unlike the participants in a public blockchain

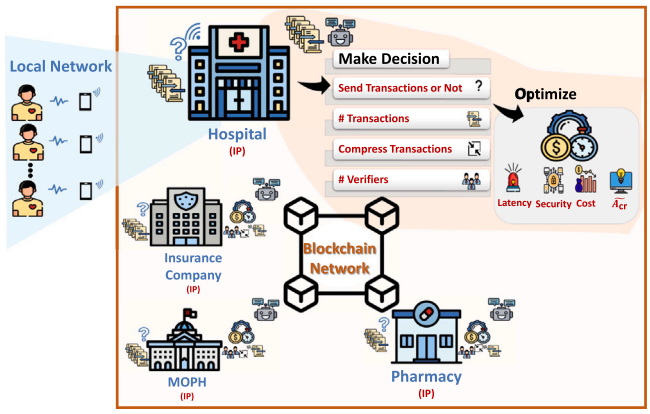



**Fig. 1.** Proposed IP-HealthChain framework.

This paper focuses on the blockchain network, which is highlighted with orange in Fig. 1. We have introduced intelligence at the blockchain participants' level to manage effective blockchain resource allocation while optimizing the trade-off between security, latency, and cost per IP. The optimization considers the transaction's constraints and characteristics: security, urgency, and queuing time. The received transactions will be queued in an urgency-based queue on the IP side. The new transactions are issued with zero queuing time. Our proposed system prevents starvation problems in the queue by deploying the *zombie prevention* algorithm explained in Section 4.3. Moreover, implicit communication between the blockchain participants is maintained by a novel reward formulation, named, *bipartite-reward*. Bipartite-reward is a scalar quantity consisting of two main compounds. The first compound, the global reward, relates to utilizing the available computational resources; all IPs receive it equally. The second compound, local reward, corresponds to the MOOP, considering the IP's local observation and optimizing the three conflicting objectives. The formulated bipartite-reward considers the mixed-taxonomy of our framework while eliminating direct IPs' communication; unlike the work in Weng et al. (2021), Rashid et al. (2018), Son et al. (2019) which considered messaging and parameter sharing approaches. Through the formulation of the reward function, the IP can indirectly perceive the system situation, which is considered an implicit nature of communication, allowing a streamlined process and eliminating additional communication costs. Additionally, this work considers compressing transactions with an urgent flag to minimize the latency. In the following subsections, a detailed explanation of IP-HealthChain components.

### 3.1. Blockchain network

The existing implementation of linear-based blockchain technology permits the simultaneous creation of multiple blocks by various participants, yet it lacks the capability to establish a valid ledger. This situation introduces a forking phenomenon in the ledger, leaving no means to ascertain the accuracy of each block. Consequently, this scenario induces disorder and uncertainty, rendering the blockchain impractical. Contrastingly, the envisioned consortium blockchain technology within IP-HealthChain represents an evolved iteration of consortium blockchain using a linear-based ledger, integrating the modified Delegated Proof of Stake (DPoS) consensus scheme, where verifiers are selected based on their computational resources. This innovative framework accommodates the concurrent generation of multiple blocks while preserving essential blockchain attributes, including immutability, transparency, data integrity, trust, and the facilitation of decentralized healthcare data sharing. Every generated block is contingent upon an optimal number of verifiers and transactions, aligning with participants' application-level requirements. This encompasses considerations such as the urgency and security levels of each transaction and the queuing time of transactions. The blockchain configuration is determined by the IP, thereby influencing system performance. This influence is exerted by effectively managing the trade-off between three conflicting objectives, all while wisely allocating blockchain resources.

**Table 1**
Impact of IP Decisions on Latency, Security, and Cost.

| Sub-Decision | Latency | Security | Cost |
|---|---|---|---|
| #Transactions (Increase) | Increase | – | Decrease |
| #Verifiers (Increase) | Increase | Increase | Increase |
| Compression Choice | Decrease | – | – |

### 3.2. Intelligent blockchain participants (IP)

In IP-HealthChain, the IPs must make the optimal decision to enhance the system's performance. As shown in Fig. 1, the action taken by IP represents four embedded actions to generate a block. The composed actions can be summarized as follows: First, whether to send any transactions. Hence, the queuing time of the remaining transactions in the queue needs to be incremented. If IP decides not to send any transaction, no other sub-decisions are needed. Second, the number of transactions to be sent. Third, whether to compress the selected transactions or not (i.e., based on the urgency level of the transaction). Finally, decide on the needed number of verifiers. The dispatch of transaction choice directly impacts maximizing the utilization of the computational resources, effective allocation, optimizing the transactions queuing time, and enhancing the overall system performance. Table 1 represents the impact of the decisions on the three conflicting objectives: latency, security, and cost. Compression is directly proportional to the latency, hence, the latency decreases on deciding the compression Nevertheless, the frequency of changing those configurations might add further expenses to the system. Therefore, all the participants decide on a specified rate of change in the smart contract.

### 3.3. Multi-objective optimization problem formulation

In the context of IP-HealthChain, each IP dynamically adjusts the blockchain configuration to optimize the delicate balance among three conflicting objectives: latency ($L$), security ($S$), and cost ($C$), all while adhering to the requirements of the transactions. The optimization formulation is designed to maximize security while minimizing latency and cost concurrently. It is crucial to recognize that scaling the number of transactions exhibits a proportional impact on latency, inversely affecting cost objectives. In this work, the verification time of the transactions is considered part of the latency objective. We are considering the availability of computational resources to speed up the verification process. Moreover, the transaction size is another critical parameter that is considered along with the number of transactions; our approach considers compressing transactions based on their urgency level, directly impacting the verification process and storage availability. Conversely, increasing the number of verifiers amplifies all the aforementioned objectives, introducing a nuanced trade-off. The following scenarios provide detailed insight into the definition of objectives and the intricate interplay between urgency and security:

**Emergency Scenarios:** In high-urgency transactions necessitating swift ledger processing, security considerations are adjusted by minimizing the number of verifiers and expediting the verification process. For instance, the prompt detection of an epidemic demands accelerated action.

**High-Security Transactions:** Transactions requiring elevated security entail more verifiers participating in the verification stage. Consequently, both latency and cost are influenced as verification latency and transaction processing costs escalate with an augmented number

of verifiers, influenced by varied computational resources and service prices. Example: Sharing a video of a necessary surgery.

In IP-HealthChain, the conditions of the transactions are mapped into different blockchain configuration modes while considering the conflicting objectives. The mathematical representation of the security $S$ is in Eq. (1), where $\kappa$ is a system coefficient, $q$ is the network scale indicator factor, and its' value is greater than or equal to two (Al-Marridi et al., 2021a). $\upsilon$ is the number of verifiers selected by the IP.

$$S(\upsilon) = \kappa.\upsilon^{q} \tag{1}$$

The latency ($L$) includes four main stages: creating, verifying, broadcasting, and uploading a block (Al-Marridi et al., 2021a). Eq. (2) refers to the total latency ($L$) where $tr$ is the number of transactions allocated in the block, and $\zeta$ is the transactions' sizes. $G$ is the required computation resources for block verification, $x_i$ is the computational resources of verifier $i$, and $M$ is the maximum number of possible verifiers to participate in the verification process. $\psi$ is a predefined parameter described in detail in Kang et al. (2019). The verification feedback size is $\hat{\xi}$ while $r_u$ and $r_d$ are the uplink and downlink transmission rates (Al-Marridi et al., 2021a).

$$L(tr, \upsilon, \zeta) = \frac{tr.\zeta}{r_d} + \max_{i \in 1,\dots,M} \left(\frac{G}{x_i}\right) + \psi.tr.\zeta.\upsilon + \frac{\hat{\xi}}{r_u} \tag{2}$$

The cost ($C$) is represented in Eq. (3), where the computational cost of verifying $tr$ transactions by $\upsilon$ verifiers is the multiplication of available computational resources $x$ of the selected verifiers and the prices of using the resources $p$.

$$C(tr, \upsilon) = \frac{\sum_i^{\upsilon}(x_i \times p_i)}{tr} \tag{3}$$

Eq. (4) represents the MOOP with $L$, the inverse of $S$ and $C$ minimized simultaneously, subject to a set of dependency constraints. The first and second constraints specify the boundaries of the selected number of transactions $tr$ and the selected number of verifiers $\upsilon$, where their values cannot go beyond $T_r$ and $M$. $T_r$ is the maximum number of transactions allocated in a block, while $M$ is the maximum number of verifiers in the system. The third constraint denotes the transaction queuing time $\tilde{a}$, which should not exceed the threshold $\tilde{a}_{th}$. The threshold $\tilde{a}_{th}$ represents the maximum duration the transaction can be queued in the IP's queue $\mathcal{Q}$. The weighting coefficients $\alpha$, $\beta$, and $\gamma$ represent the relative significance of the three objectives based on the system administrator requirements, where $\alpha + \beta + \gamma = 1$. The objectives' maximum values certify normalized, comparable, and unitless equations. The maximum latency, security, and cost symbolized $l_m$, $s_m$, and $c_m$, respectively. As mentioned earlier, the IP should decide on compressing the transaction ($d$) if needed, considering the urgency level, effecting the size of the transactions ($\zeta$) directly. The transaction size reflects on the latency objective $L$. In this paper, we optimize the MOOP for each IP and utilize the shared computational resources. We will refer to this optimization as CSLR throughout the paper.

$$\begin{aligned} \underset{tr,\upsilon,d}{\text{minimize}}\ & \alpha \cdot \frac{L(tr,\upsilon,\zeta)}{l_m} + \beta \cdot \frac{s_m}{S(\upsilon)} + \gamma \cdot \frac{C(tr,\upsilon)}{c_m} \\ \text{subject to}\ & \\ & 1 \le tr \le T_r, \\ & 1 \le \upsilon \le M, \\ & \tilde{a} \le \tilde{a}_{th}. \end{aligned} \tag{4}$$

## 4. Multi-agent deep reinforcement learning formulation

Real-time system optimization and change adaptability are imperative. However, solving the non-convex problem repeatedly for each variation in system parameters proves inefficient. To address this, we employ a learning-based optimization approach at the IP level using MAD3QN. In our context, especially with multiple IPs and incomplete knowledge of the environment, the decentralized extension of the Partially Observable Markov Decision Process (Dec-POMDP) becomes pivotal. Dec-POMDP finds application in multi-agent cooperative systems, defined as a tuple $\mathcal{G} = (\mathbb{D}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \Omega, \mathcal{R}, \lambda)$, where $\mathbb{D} = \{1 \cdots N\}$ is a finite set of agent indices; $\mathcal{S}$ is the set of finite states space, $\mathcal{A}$ is the set of all possible actions, $\mathcal{O}$ is the set of finite observations, $\mathcal{T}$, and $\Omega$ are the state and observation transition probability functions, respectively. $\mathcal{R}$ is the reward function (i.e., equal for all agents in a cooperative setting), and $\lambda \in [0, 1]$ is the discount factor that ensures that the agent will not rely totally on future rewards (Oliehoek and Amato, 2015). Our new formulation involves multiple heterogeneous IPs that compete and cooperate, necessitating the use of Mixed Dec-POMDP due to varying rewards among IPs compared to the cooperative scenario. IPs learn optimal policies by observing the environment and collaborating with or competing against other IPs to maximize system utility. Competition involves using verifiers to satisfy local needs (urgency, security, and transaction queuing time) while making strategic decisions regarding other IPs. The challenge lies in determining the opportune moment to send transactions, especially when the available computational resources are insufficient. Participants must decide between sending with available resources or abstaining. The subsequent subsections elaborate on the detailed explanation of Mixed Dec-POMDP tuples at each time step.[3]

### 4.1. State and observation spaces

At a given decision step ($t$), $s$ represents the system state as a union of the mapped transactions for all the IPs $N$, along with the shared available computational resources in the IP-HealthChain framework. Each IP $i \in \mathbb{D}$ has access to a partial part of the state, $o \in \mathcal{O}$. $\mathcal{M}$ is the finite set of verifiers indices, $\mathcal{M} = \{1 \cdots M\}$, and $\mathcal{T}_r$ is the finite set of transactions indices, $\mathcal{T}_r = \{1 \cdots T_r\}$. The IP should decide an action based on the observation $o^i$, which is the concatenation of IP's transactions ($\mathcal{Q}$) considering the three transaction characteristics and the computational resources ($\widetilde{A_{cr}}$) as illustrated in Eq. (5).

$$o = (\mathcal{Q}, \widetilde{A_{cr}}) \tag{5}$$

Where $\mathcal{Q}$ and $\widetilde{A_{cr}}$ are 2D vectors and represented as $\mathcal{Q} = \left[(u_1\ se_1\ \tilde{a}_1) \cdots (u_{T_r}\ se_{T_r}\ \tilde{a}_{T_r})\right]$ and $\widetilde{A_{cr}} = \left[(x_1\ p_1\ \iota_1\ \tilde{p}_1\ \tilde{z}_1) \cdots (x_M\ p_M\ \iota_M\ \tilde{p}_M\ \tilde{z}_M)\right]$, respectively. Each component of the observation space is defined as follows:

**IP's urgency-based queue** ($\mathcal{Q}$), where the transactions represented by three characteristics $\forall t' \in \mathcal{T}_r$ as follow:

- $u_{t'}$: is the urgency level for transaction ($t'$), where $u_{t'} \in [0, 1]$. An urgency level of 0 indicates a high urgency requirement where latency needs to be the minimum, while 1 is the contrary.
- $se_{t'}$: is the security level for transaction ($t'$), where $se_{t'} \in [0, 1]$. A security level of 1 indicates a high-security requirement, where security needs to be the maximum; hence, more verifiers must be assigned and vice versa.
- $\tilde{a}_{t'}$: is the queuing time for the transaction ($t'$), where $\tilde{a}_{t'} \in \{0 \cdots \tilde{a}_{\text{th}}\}$. The queuing time for any transaction ($\tilde{a}_{t'}$) cannot exceed a specific application-level threshold ($\tilde{a}_{\text{th}}$).

**The available computational resources** ($\widetilde{A_{cr}}$) consist of five main components, considering $K$ time steps per episode, for all verifiers $\upsilon' \in \mathcal{M}$:

- $x_{\upsilon'}$: is the quantity of available computational resources, where $x_{\upsilon'} \in \mathbb{Z}^*$.
- $p_{\upsilon'}$: is the price of using the resources of verifier $\upsilon'$, where $p_{\upsilon'} \in \mathbb{Z}^*$.

[3] We will refer to $s_t$, $s_{t+1}$, $a_t$, $a_{t+1}$, $o_t$, $o_{t+1}$ and $\theta_t$ as $s$, $s'$, $a$, $a'$, $o$, $o'$ and $\theta$, respectively

- $\iota_{v'}$: is the number of time steps needed until releasing the resources assigned for processing the previously selected transactions, where $\iota_{v'} \in \{0 \cdots K\}$.
- $\tilde{p}_{v'}$: is the maximum processing time for accomplishing a task assigned to a verifier $v'$, where $\tilde{p}_{v'} \in \{1 \cdots K\}$.
- $\tilde{z}_{v'}$, is the number of IPs occupying the resources of the verifier $v'$, where $\tilde{z}_{v'} \in \{1 \cdots N\}$.

### 4.2. Multi-discrete action space

As discussed in Section 3.2, during a given decision step ($t$), each IP takes a multi-level action to generate a block. The primary goal for each agent is to optimize the configurations of the blockchain network within the IP-HealthChain system, ultimately maximizing the CSLR utility function. Formally, an action ($a \in \mathcal{A}$) taken by an $\text{IP}_i$ for $i \in \mathbb{D}$ at $t$ is expressed in Eq. (6). Subsequently, each component of the action is defined.

$$a = (d, tr, v) \tag{6}$$

- $d$: is the decision of whether to send any of the transactions or not. $d \in \{0, 1, 2\}$, where 0 represents being in an idle state, when no block is generated. 1 represents sending the selected transactions on compression mode, while 2 represents sending the transaction without compression, considering the selected transactions' urgency level.
- $tr$: is the number of transactions allocated in a block. $tr \in \mathcal{T}_r$.
- $v$: is the needed number of verifiers for the verification process of a block. $v \in \mathcal{M}$.

### 4.3. Environment design

The Mixed Dec-POMDP environment reflects the current state of the IP-HealthChain system at the time step($t$), encompassing all participating IPs. Each IP has access to a partial view of the state ($o$). Subsequently, each IP takes an appropriate action and communicates it to the environment. The environment, in turn, undergoes a state transition based on the collective actions taken, emitting a reward signal ($r$) for each IP. In the context of IP-HealthChain, IPs operate without complete knowledge of the environment's design. Therefore, the experience replay technique is employed. Each IP maintains its replay buffer filled with experiences of observations, actions, rewards, and transitions. This technique supports learning the optimal policy ($\pi^*$), which maps observations to the most effective actions. Further details are provided in the subsequent subsections.

#### 4.3.1. State and observation transition

In the context of the IP-HealthChain, the transition of the Mixed Dec-POMDP defines the next state $s'$ based on the current observation action pairs $(o, a)$ from all the IPs. The transition of the state elements considers the following:

**Transaction Queuing Time Concept:** Each IP manages a dedicated queue ($\mathcal{Q}$) wherein received transactions are allocated. Queuing time ($\tilde{a}$) is one of the transactions' characteristics that signifies the duration spent within an IP's queue, with its value incrementing at each time step ($t$). The introduction of queuing time is strategically motivated by its ability to streamline the observation transition process. This consideration is precious in accommodating the temporal dynamics of the blockchain and meeting the deployment requisites of RL. Thus, queuing time ensures the maintenance of a tractable state and observation transition, providing IPs with valuable samples for subsequent observations ($o'$) without explicitly modeling observation transition distributions. However, it is noteworthy that all IPs' queues are urgency-based, which raises the potential issue of unallocated transactions, categorically referred to as zombie transactions. The occurrence of a transaction turning into a zombie transaction is contingent on its urgency requirements being notably lower compared to other transactions in the queue. Therefore, considering the IP action, we have introduced a zombie prevention algorithm to check and handle zombie transactions through a multi-level prioritization strategy specifically, when the queuing time of a transaction surpasses a predefined application-level threshold ($\tilde{a}_{\text{th}}$), the algorithm triggers a manipulation of the priority level, consequently influencing the processing order of transactions. The algorithm's passed flag maps the decision and the available computational resource ($x$). This flag is true when the transaction satisfies the specified requirements and false otherwise.

**Computational Resources Allocation and Management:** The allocation and management of computational resources within the IP-HealthChain framework involve a shared pool of blockchain verifiers' computational capabilities, consistently ordered in a descending manner. In response to an IP's decision regarding the required number of verifiers, the computational resources of the selected verifiers are evenly distributed among the IPs that have chosen them. Consequently, the cost associated with these resources is equally shared among the involved IPs, highlighting a scenario where each verifier's computational resources are concurrently accessible to multiple blockchain participants. Each computational resource is associated with an approximated processing time to handle the assigned task. This processing time dynamically adjusts based on the number of IPs utilizing the

**Algorithm 1:** Handling Zombie Transactions

```
Data: Current Q_IP, IP action (d, tr, v) and passed
Result: Updated Q_IP
Initialization: Z ⟵ Initialize Zombie set
for each transaction i in Q_IP do
    if ã_i ≥ ã_th then
        j = index of IP transaction i
        Z.insert(j)
if d ≠ 0 then
    if passed = False then
        /* |Z| is the number of Zombie transactions*/
        if |Z| ≠ 0 then
            Drop Zombie transactions in Z from Q_IP
            Accept receiving |Z| new transactions
        else
            Increment Transactions' ã in Q_IP
    else
        if |Q_IP| = tr then
            Accept receiving |Q_IP| new transactions
        else
            if |Z| ≠ 0 then
                Manage |Z| considering tr
                Drop selected transactions from Q_IP
            else
                Drop tr transactions from Q_IP
            Increment Transactions' ã in Q_IP
            Accept receiving tr new transactions
else
    */Nothing will be sent*/
    if |Z| ≠ 0 then
        Drop Zombie transaction from Q_IP
        Accept receiving |Z| new transactions
    else
        Increment Transactions queuing time in Q_IP
return Q_IP
```

**Table 2**
Notations Used in the Reward Formulation.

| Symbol | Description |
|---|---|
| $l_p$ | Latency penalty |
| $s_p$ | Security penalty |
| $u_{\text{th}}$ | Urgency threshold |
| $\tilde{a}_{\text{th}}$ | Queuing time threshold |
| $d_p$ | Penalty on wrong sending decision |
| $\mathcal{E}_u$ | Accepted error interval for latency |
| $\mathcal{E}_s$ | Accepted error interval for security |
| $U_p$ | Insufficient resource utilization penalty |

resources of a particular verifier. The release time ($t_v$) for the resources of a given verifier ($v$) spans from the moment of allocation until it reaches $\tilde{p}_v$. Accordingly, the IP's actions must conscientiously consider the availability of resources within the IP-HealthChain, aiming to ensure maximal utilization while aligning with the stipulated security requirements.

#### 4.3.2. Bipartite-reward formulation

The reward function has been designed to reflect the CSLR, where each IP has reward ($r$) at time step $t$ for the current observation ($o$) and action ($a$). The reward function returns a scalar out of two compounds, namely, the global and local rewards. Therefore, it was called bipartite-reward. The global reward ($r_g$) indicates the reward obtained by respecting the available computational resources in the system and considering the need of all the IPs, and the local reward ($r_l$) obtained by the MOOP decision by each participant independently from the others. The reward values are bounded between *zero* and a *hundred* and are affected by determining application requirements. The bipartite-reward reflects the IPs' cooperate-competitive behavior and ensures a stationary MARL environment. Detailed descriptions of the symbolic representations for application-level penalties are provided in Table 2. These representations directly impact the reward formulation, mitigating the sparse reward problem and fostering applicability to real-case scenarios.

**Global Reward** ($r_g$): When the IP decides to send transactions to the blockchain, these transactions necessitate processing and verification by a specific number of selected verifiers. At the time step ($t$), the IP is expected to leverage the total available computational resources in the system ($X$), ensuring idle status when no resources are available, where $X = \sum_{i=1}^{M} x_i$. The global reward function at ($t$) expressed in Eq. (10), is influenced by three conditions. The first condition ($c_1$) encompasses two cases detailed in Eq. (7). In the first case, the IP decides to send transactions without available resources ($X$). The second case, the IP opts not to send transactions despite available resources ($X$). In the latter scenario, a penalty ($U_p$) is applied to the reward. The second condition($c_2$) is outlined in Eq. (8) arises when the IP chooses to remain idle as $X$ equals *zero*. If this condition holds, $r_g$ takes on the maximum reward value in IP-HealthChain. However, when the IP decides to stay idle while there are resources in the system, the third condition ($c_3$) in Eq. (9) will hold. Considering the urgency level of the IP transactions in $\mathcal{Q}_{\text{IP}}$, if the average urgency level of the transactions ($\bar{u}_*$) is below $u_{\text{th}}$, where $\bar{u}_* = \frac{\sum_{i=1}^{T_r} u_i}{T_r}$, the reward will be affected by two types of penalty: $U_p$ and $d_p$. Otherwise, the second case of the first condition will affect the global reward $r_g$. All the IPs in the framework receive the same $r_g$ at ($t$), ensuring the framework's stability and stationary aspects.

$$c_1 : \begin{cases} d_t \neq 0 \text{ and } X = 0 \\ d_t = 0 \text{ and } X \neq 0 \end{cases} \tag{7}$$

$$c_2 : d_t = 0 \text{ and } X = 0 \tag{8}$$

$$c_3 : d_t = 0 \text{ and } X \neq 0 \text{ and } \bar{u}_* \leq u_{\text{th}} \tag{9}$$

$$r_g = \begin{cases} 100 - (100 \times U_p), & \text{When (7) holds} \\ 100, & \text{When (8) holds} \\ 100 - (100 \times (U_p + l_p)), & \text{When (9) holds} \end{cases} \tag{10}$$

**Local Reward** ($r_l$): Each IP needs to decide based on its transaction's characteristics: the urgency and security levels. The decision includes (1) the number of transactions to be sent, (2) the number of verifiers to verify these transactions, and (3) the compression option. The $r_l$ describes the high-level objective of minimizing total cost and latency while meeting the security requirements. Eq. (15) represents the converted maximized function, which reflects the MOOP in Eq. (4). The three conflicting objectives are calculated using Eq. (1), (2), and (3) with weighting factors $\alpha$, $\beta$, and $\gamma$ to indicate the importance of each objective. In IP-HealthChain, the $r_l$ is restricted by application-level conditions, which reflects on the reward. If the selection of the compression option, transactions, and verifiers does not align with the requested urgency and security levels, penalties will be applied accordingly. The conditions are listed below and followed with the final local reward $r_l$ in Eq. (15).

- Checking the transactions queuing time, where violating the threshold causes a zero reward.

$$c_4 : \forall i \in \mathcal{Q} \quad \exists \quad \tilde{a}_i > \tilde{a}_{\text{th}} \tag{11}$$

- Checking the violation of the security level requirements considering the accepted error interval for security $\mathcal{E}_s$. The average security of the selected transactions $tr$; $\bar{s} = \frac{\sum_{i=1}^{tr} se^{(i)}}{tr}$ and the provided level $S_v$.

$$c_5 : S_v > (\bar{s}(1 + \mathcal{E}_s)) \text{ or } S_v < (\bar{s}(1 - \mathcal{E}_s)) \tag{12}$$

- Checking the violation of the urgency level requirements considering the accepted error interval for security $\mathcal{E}_u$. The average security of the selected transactions $tr$; $\bar{u} = \frac{\sum_{i=1}^{tr} u^{(i)}}{tr}$ and the provided level $L^v_{tr,\zeta}$.

$$c_6 : L^v_{tr,\zeta} > (\bar{u}(1 + \mathcal{E}_u)) \text{ or } L^v_{tr,\zeta} < (\bar{u}(1 - \mathcal{E}_u)) \tag{13}$$

- Checking compression decision, where transactions should satisfy urgency threshold $u_{\text{th}}$to be eligible for compression. Decreasing the transaction size is directly proportional to the latency, as stated in Eq. (2) and the white paper of Hyperledger Fabric blockchain (Scherer, 2017).

$$c_7 : \bar{u} > u_{\text{th}} \tag{14}$$

In Eq. (15), the reward is defined according to all permutations of all possible conditions. If condition (12) holds, a penalty of $s_p$ will be applied to $r_l$. If condition (13) holds, then $l_p$ reflects at $r_l$. In the case of breaking the requirements of urgency and security, then $s_p$ and $l_p$ are applied to $r_l$. A penalty of $d_p$ will be applied to $r_l$ if condition (14) holds.

$$r_l = \alpha \cdot (1 - \frac{L(tr, v, \zeta)}{l_m}) + \beta \cdot \frac{S(v)}{s_m} + \gamma \cdot (1 - \frac{C(tr, v)}{c_m}) \tag{15}$$

$$r_l = \begin{cases} 0, & \text{When (11) holds} \\ r_l - (r_l \times (s_p \times (\beta))), & \text{When (12) holds} \\ r_l - (r_l \times (l_p \times (\alpha))), & \text{When (13) holds} \\ r_l - (r_l \times (l_p + s_p) \times (\alpha + \beta)), & \text{When (12), (13) hold} \\ r_l - (r_l \times (s_p \times \beta \times d_p)), & \text{When (12), (14) hold} \\ r_l - (r_l \times (l_p \times \alpha \times d_p)), & \text{When (13), (14) hold} \\ r_l - (r_l \times (l_p + s_p) \times (\alpha + \beta) \times d_p), & \text{When (12), (13), (14) hold} \end{cases}$$

Finally, Eq. (16) indicates the bipartite-reward ($r$), where a priority-based weighting factor ($\delta$) is considered to represent the degree of

the cooperative-competitive behavior, and it is an application-level variable. When the value of $\delta$ is *zero*, the IPs behave fully competitively. Otherwise, if $\delta$ is *one*, the IPs are fully cooperative.

$$r = \delta \cdot r_g + (1 - \delta) \cdot r_l \tag{16}$$

### 4.4. Multi-agent dueling double deep Q-network (MAD3QN) solution

The Q-learning algorithm, a well-established table-based RL method, is widely employed for determining the optimal action-value function, denoted as the Q-function. Adhering to the Bellman Optimality equation introduced in (Sutton and Barto, 2018), the optimal Q-function is expressed recursively as $Q^*(s,a) = \mathbb{E}_{s'}\left[r + \lambda \max_{a'} Q^*(s',a')|s,a\right]$. Deep Q-learning (DQL) represents a model-free approach that departs from traditional Q-learning by substituting Q-tables with neural networks parameterized by $\theta$ to approximate the action-value function, as articulated by Mnih in Mnih et al. (2013). DQL employs two networks: the online network and the target network, both utilizing the same maximum operator. The online network's parameters are continually updated, while the target network's parameters ($\bar{\theta}$) update after a specified number of time steps. It is noteworthy that DQL may encounter the false-positive problem, leading to overly optimistic estimates of actions due to the maximum operators used in both networks. To mitigate this issue, the Dueling Double Deep Q-Network (D3QN) emerges as a refined version of DQL (Wang et al., 2016). D3QN addresses the false-positive problem by evaluating the action using the target network while selecting the action based on the online network and enhances the state–action mapping by eliminating unnecessary estimations; hence, it accelerates the learning. Eq. (17) represents the TD-Target:

$$Y = r + \lambda Q(s', \arg\max_{a'}(Q(s',a';\theta)); \bar{\theta}) \tag{17}$$

This modification contributes to a more stable and accurate learning process. The structure of D3QN explicitly separates the representation of state values $V$ and state-dependent action advantages $A$ via two separate streams. $V$ represents how good it is to be in a specific state when following the policy $\pi$, while $A$ represents the relative advantage of each action in the Action space $\mathcal{A}$ when observing the observation $o$. Eq. (18) presents the returned $Q$ function as a summation of $V$ and $A$ where the value of the advantage function is deduced by its mean to raise the stability level of the optimization as recommended by Wang et al. (2016). In Eq. (18), $\ddot{\alpha}$ and $\ddot{\beta}$ represent separate sets of the parameters to predict the two aggregated streams $A$ and $V$, successively.

$$Q(o,a;\theta,\ddot{\alpha},\ddot{\beta}) = V(o;\theta,\ddot{\beta}) + \left(A(o,a;\theta,\ddot{\alpha}) - \frac{1}{|\mathcal{A}|}\sum_{a'}^{\mathcal{A}} A((o,a';\theta,\ddot{\alpha}))\right) \tag{18}$$

In the IP-HealthChain context, IPs exhibit heterogeneity with competitive and cooperative behaviors. To address the optimization problem in IP-HealthChain with $N$ IPs, we employ Mixed-MAD3QN, formulated as a Mixed Dec-POMDP. Experience replay and soft updates of neural networks are utilized to stabilize training, minimize the expected Temporal Difference error (TD-error) loss function, and enhance weight convergence (Mnih et al., 2015). Fig. 2 illustrates the training phase of the IPs. In the proposed D3QN, leveraging the advantage function concept, two sets of 1D convolutional layers serve as hidden layers in the network. These layers are then flattened and concatenated to obtain the $Q$ function from the aggregated streams $A$ and $V$. ReLU activation function is used. The RMSProp Optimizer is employed (Ruder, 2016). The overarching objective is to determine the strategy ($\pi^*$) that maximizes long-term reward expectations $\mathcal{V}$ for all IPs in the IP-HealthChain framework, considering the set of action strategies.

The overall expected reward for $\text{IP}_i$ for $i \in \mathbb{D}$ given a particular observation is $\mathcal{V}^i\left(\mathcal{O}^i,\mathcal{A}^i\right) = \mathbb{E}\left[\sum_{t=1}^{T} \lambda^{t-1} R_t^i\left(\mathcal{O}_t^i,\mathcal{A}_t^i\right)\right]$, where $T$ represents the set of all decision time slots. During D3QN's learning process, the action selection balances exploitation and exploration. The IP explores new actions while reinforcing known good ones. This paper employs $\epsilon$-greedy exploration. With probability $\epsilon$, the IP takes a random action. Otherwise, it takes the action with the highest observed Q-value as Eq. demonstrates, balancing exploitation and exploration. The choice of $\epsilon$ is essential and can significantly impact the agent's performance. A too-high value of $\epsilon$ can lead to excessive exploration and prevent the agent from learning optimal policies, while a too-low value of $\epsilon$ can limit the agent's ability to discover new and potentially better actions. A constant decay rate may not effectively capture the dynamic nature of exploration and exploitation in complex reinforcement learning tasks. Due to this, a modified $\epsilon$-decay approach is proposed that dynamically adjusts the probability of exploration based on the number of exploration time steps ($t_e$). This adaptive approach allows the agent to explore more actively during the initial stages of learning when it is crucial to discover new and potentially better actions. It gradually reduces exploration as it gains more experience and becomes more confident in its policy. The value of $\epsilon$ is initially assigned to its maximum; then it will linearly anneal from $\epsilon_{\max}$ to $\epsilon_{\min}$ over the exploration phase, thereafter it will be fixed at $\epsilon_{\min}$ as Eq. (20) shows.

$$a = \begin{cases} \text{Random}, & \text{With probability } \epsilon \\ \epsilon\text{-greedy policy}, & \text{otherwise} \end{cases} \tag{19}$$

$$\epsilon = \epsilon_{\min} + ((\epsilon_{\max} - \epsilon_{\min}) \times (t_e - t)/t_e) \tag{20}$$

Algorithm 2 represents the detailed steps while training our proposed MAD3QN for multiple episodes $E$. For all the IPs, initialize the weights of both the online and target neural networks along with replay memory $D$ and $\epsilon$. The IP balances exploitation and exploration during each episode by collecting experience tuples through interactions. Based on the action made by the IP, a bipartite-reward will be received based on Eq. (16) with the next observation considering the deployment of zombie prevention algorithm. Afterward, the transition of the observation-action will be stored in the experience replay memory $D$ as a tuple to improve the estimation of $Q^*$ during the optimization process. At each episode, a mini-batch of experiences $\rho$ are randomly sampled from $D$ to update $\bar{\theta}$ by calculating the TD-target $Y_i$ defined in Eq. (17). Repeatedly sampling stored experiences improves the sampling efficiency and breaks the correlation in subsequent updates. Finally, RMSProp optimizer (Ruder, 2016) is used to fit $\theta$ to $Y^i$ then updating $\bar{\theta}$ to reflect the recent knowledge of the environment after $t_\tau$ steps. Eventually, the convergence is achieved where $\theta \sim \bar{\theta}$.

### 4.5. Complexity analysis of the MA3DQN algorithm

This subsection provides a comprehensive analysis of the time complexity of the MAD3QN algorithm using big-O notation. Traditional MARL techniques introduce overhead due to agents coordination mechanisms, such as synchronizing experience replay buffers and maintaining a centralized critic. Our MAD3QN algorithm, featuring a bipartite-reward function, eliminates agent communication and coordination complexities. The algorithm incorporates neural networks with 1D convolutional and fully connected layers, forming two streams ($A$ and $V$). The overall time complexity ($C_a$) of the neural network architecture is given by: $C_a = O\left(l_1 \cdot (H \cdot W \cdot n) + l_2 \cdot n\right)$. Here, $l_1$ and $l_2$ represent the numbers of convolutional and dense layers, $H$ and $W$ are the input layer's height and width, and $n$ is the number of neurons. Layers such as input, reshape, flatten, concatenate, and lambda are considered to have constant complexity $O(1)$. The complexity associated with soft updates and experience replay, denoted as $C_p$, considers hyperparameters such as the soft update factor ($\tau$), batch size ($|\rho|$), and experience replay buffer size ($|D|$): $C_p = O\left(\frac{|D|}{|\rho|} + \tau\right)$. Notably, the complexity decreases with larger batch sizes, indicating more efficient processing, while it increases with a decreasing soft update factor, signifying higher computational demand for faster updates. The MAD3QN

**Algorithm 2:** Training Process of MAD3QN Algorithm for Blockchain Optimization

**Input:** System's parameters
**Output:** $\theta^*$: The NN parameters for the approximation $Q^*$
**Initialization:** $\forall$ IP$_i$ where $i \in \mathbb{D}$
$\theta \leftarrow$ Initialize online network parameters randomly
$\bar{\theta} \leftarrow \theta$ Initialize target network parameters
$\mathcal{D} \leftarrow$ Initialize replay memory with the size $|D|$
$\epsilon \leftarrow$ Initialize $\epsilon$ to $\epsilon_{\max}$
**for** *episode* $= 1$ *to* $E$ **do**
  Initialize System State $s_0$
  **for** $t = 1$ *to* $K$ **do**
    $\forall$ IP$_i$ where $i \in \mathbb{D}$, Select observation update action $a^i$
    $\forall$ IP$_i$ where $i \in \mathbb{D}$, Execute actions $a^i$ and observe reward $r^i$ (16) and new observation $o'^i$
    Environment Status done $\leftarrow$ Boolean
    $\forall$ IP$_i$ where $i \in \mathbb{D}$ Insert Tuple of Experience into Replay Memory $\mathcal{D}^i \leftarrow (o^i, a^i, r^i, o'^i, done)$
    Apply $\epsilon$ decay following (20)
    Update State $s_t \leftarrow s_{t+1}$
    /*Updating the estimates*/
    **for** $i \in \mathbb{D}$ **do**
      Select random $\rho^i \subseteq \mathcal{D}^i$ $\rho^i \leftarrow \left\{(o^{(j)}, a^{(j)}, r^{(j)}, o'^{(j)}, done^{(j)})\right\}_{j=1}^{|\rho|}$
      Calculate Q-targets using (17): $Y^{ij} \leftarrow \left\{y^{(j)}\right\}_{j=1}^{|\rho|^i}$
      Fit $Q^{ij}(s, a; \theta, \ddot{\alpha}, \ddot{\beta})$ to Targets $Y^{ij}$ using RMSProp Optimizer
      **if** $t = t_\tau$ **then**
        update the target network $\bar{\theta}$ $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$
**return** $\theta^* \sim \theta$

algorithm's training complexity ($C_T$) for $N$ IPs over $E$ episodes, each with $T$ time steps, is given by: $C_T = O(N \cdot E \cdot T)$. Thus, the overall MAD3QN algorithm complexity can be expressed as the summation of $C_a$, $C_T$, and $C_p$ and can be simplified under one big-O notation as $O\left(N \cdot E \cdot T \cdot \left(\frac{|D|}{|\rho|} + \tau\right) \cdot \left(l_1 \cdot H \cdot W \cdot n + l_2 \cdot n\right)\right)$. This expression succinctly captures the computational requirements, highlighting their linear growth with the number of agents, training episodes, and time steps per episode. The complexity is further influenced by the trade-offs between experience replay size and batch size, the soft update parameter, and the neural network architecture.

## 5. Performance analysis and simulation results

This section delves into the experimental setup, assesses the convergence of D3QN IPs in IP-HealthChain, and benchmarks our proposed approach against existing heuristic policies. The evaluation methodology encompasses running all the policies for a hundred testing episodes and scrutinizing the following parameters: accumulated rewards, computational resources consumption, MOOP optimization objectives, and action-time expenditure for each observation.

### 5.1. Experimental setup and design

To thoroughly evaluate the performance of IP-HealthChain under varying conditions, we conducted simulations utilizing the parameters outlined in Table 3. We focused on small and medium-scale setups,

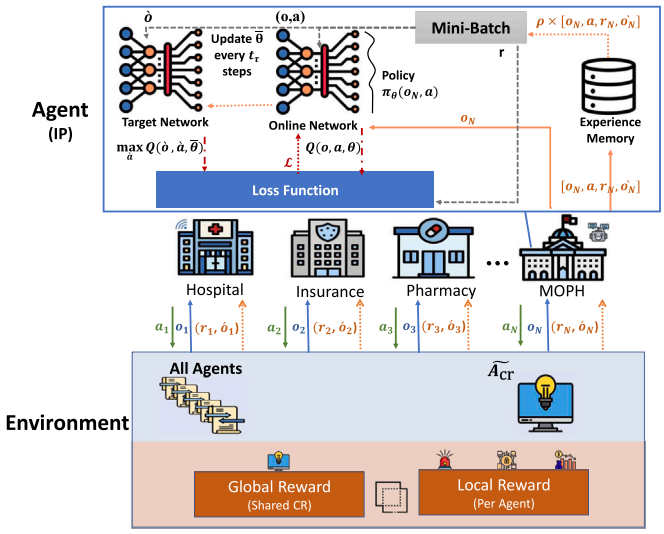



**Fig. 2.** Mixed MAD3QN training architecture in IP-HealthChain framework.

**Table 3**
Optimization parameters.

| Parameters | Values |
|---|---|
| $w$ | 0.5 MHz |
| $SNR_d$ | 10 dB |
| $SNR_u$ | 12 dB |
| $\hat{\xi}$ | 0.5 Mb |
| $\zeta$ | $2 \times 10^5$ |
| $q$ | 2 |
| $k$ | 1 |

encompassing two and four IPs. In this section, we considered a homogeneous group of IPs representing entities with similar requirements (e.g., hospitals). Initially, we employed equal weighting factors for each objective in IP-HealthChain ($\alpha, \beta$, and $\gamma$), ensuring that each objective received equal consideration. This approach allowed for a balanced system performance assessment across multiple objectives. To capture the intricate reward structure of IP-HealthChain, we introduced a weighting factor ($\delta$). This factor played a crucial role in reflecting the relative importance of each reward component and the agents' mixed cooperative and competitive behavior. This refinement enabled a more nuanced evaluation of the system's ability to optimize conflicting objectives while considering the strategic interactions between the IPs.

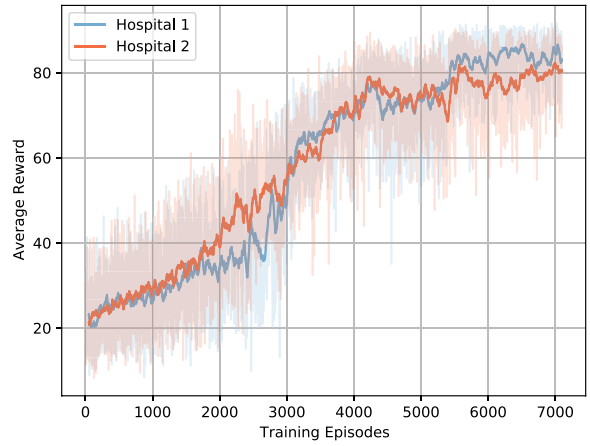
### 5.2. Training analysis for learning-based IPs

The training process follows algorithm 2, with hyper-parameters values reported in Table 4. These values were set based on experimental observations and analysis to maximize the performance. Fig. 3 represents the reward while training for two and four homogeneous IPs. The shown rewards are smoothed over a window of 50 episodes. Fig. 3 proves that the proposed approach could reach convergence empirically. All IPs explore the environment by taking random actions and improve their random policy toward learning an optimal version by decaying the value of $\epsilon$. The convergence happens after roughly 5500 episodes at the small-scale framework with an average reward unit of $\sim$ 80. Meanwhile, at the medium-scale framework, the convergence occurs after almost 4000 episodes with an average reward unit of 70. The proposed D3QN IPs are adaptable to application-level changes as they preserve their convergence and high performance. In Fig. 4, the utilization of the available computational resources during the training of homogeneous IPs is presented on medium and small-scales during the training process. Both cases learn to use the maximum available computational resources while maintaining other constraints and maximizing the ultimate reward ($r$), as presented in Fig. 3.

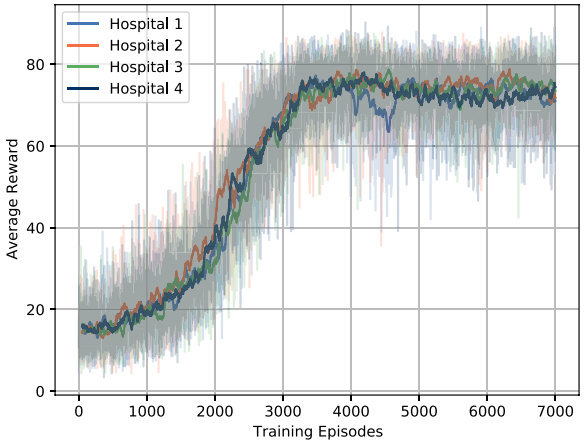
**Table 4**
System parameters.

| Parameters | Values[a] |
|---|---|
| Maximum number of transactions ($T_r$) | SS: 10, MS: 20 |
| Maximum number of verifiers ($M$) | SS: 5, MS: 10 |
| Learning episodes ($E$) | $7 \times 10^3$ |
| Episode length ($T$) | 20 |
| Number of IPs ($N$) | SS: 2, MS: 4 |
| Discount factor ($\lambda$) | 0.99 |
| Maximum exploration rate ($\epsilon_{\max}$) | 1 |
| Exploration steps ($t_\epsilon$) | 59 500 |
| Q-Network learning rate ($\varphi$) | $3 \times 10^{-4}$ |
| Q-Network neurons ($n$) | SS: 128, MS: 512 |
| Activation function ($\partial$) | ReLU |
| Optimizer | RMSProp |
| Batch size ($\lvert\rho\rvert$) | 128 |
| Replay buffer size ($\lvert D\rvert$) | $50 \times 10^3$ |
| Soft update factor ($\tau$) | $10^{-3}$ |
| Soft update target steps ($t_\tau$) | 4 |

[a] SS refers to small-scale framework. MS refers to medium-scale framework.




(a) Two IPs




(b) Four IPs

**Fig. 3.** IPs rewards over the training episodes.

### 5.3. Performance comparison

Three non-AI approaches for decision making are considered for comparison to verify the performance of our proposed Mixed-MAD3QN in IP-HealthChain.

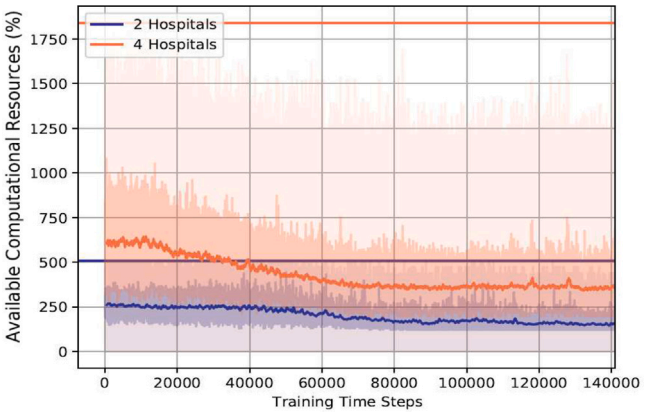



**Fig. 4.** The usage of the available computational resources in small and medium IP-HealthChain Networks, smoothed over different window sizes (original, 10, and 500 steps, respectively) during the training process. The solid straight lines represent the maximum computational resources in the system ($X$), where the blue is for the small-scale, and the orange is for the medium-scale.

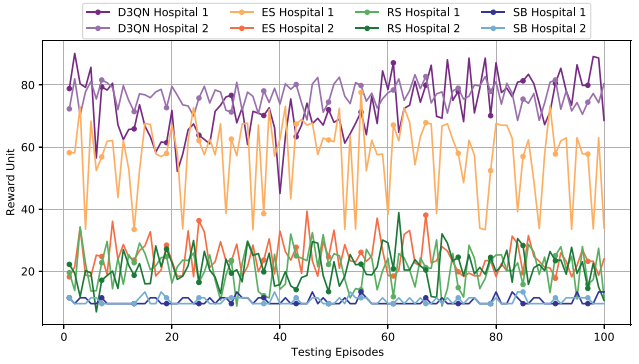
#### 5.3.1. Proposed policies for IPs

- Exhaustive Search Policy (ES): finds a local-optimal solution at a given observation per agent; by searching all possible actions sequentially and selecting the best current action. This approach was used in Abdellatif et al. (2020) and used as a baseline in Al-Marridi et al. (2021a).
- Random-Selection Policy (RS): chooses the action uniformly random without considering the environment's state, precisely like the exploration phase of any Q-Learning approach. This approach is very lightweight, with several drawbacks due to its blindness attitude.
- Static-Based Policy (SB): decides the same action for all possible observations without considering the environment's state.

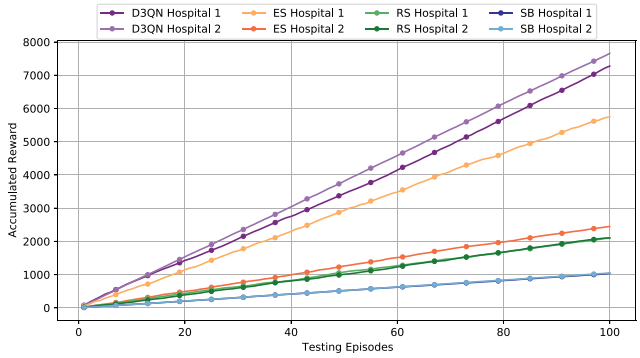
#### 5.3.2. Reward analysis

Fig. 5 showcases the accumulated rewards across various policies during testing episodes in a real-time scenario, specifically focusing on a small-scale context. The reward unit denotes the average reward per episode, considering ($K$) time steps per episode, and the accumulated reward represents the summation of these reward units across all episodes. Fig. 5(a) details the reward unit during the testing episodes. Optimal performance was observed when employing the D3QN approach, with both IPs achieving a maximum reward value of 80. D3QN demonstrated stable behavior with minimal drops compared to alternative non-AI approaches. This behavior can be attributed to D3QN's long-term planning capabilities, emphasizing the expectation of the summation of future rewards. D3QN IPs achieved the highest accumulated reward, as depicted in Fig. 5(b). The intelligent D3QN IPs, by stabilizing performance, managed to maximize future rewards. Contrastingly, the ES approach struggled to achieve comparable performance for both IPs, exhibiting a $\sim 58\%$ difference. The RS approach outperformed the SB approach, considering its exploration of various actions compared to the fixed action employed by SB across all observations. The linear behavior of accumulated rewards indicates the convergence of learning-based IPs during the training phase. The average reward during the testing episodes for D3QN, ES, RS, and SB IPs stood at 75, 41, 21, and 10.5, respectively.

In Fig. 6, deploying ES IPs proved unfeasible due to the complexity introduced by the increased number of IPs, resulting in a multi-dimensional discrete action space requiring substantial experimental and computational resources. Moving to a medium-scale framework, D3QN IPs again excelled, achieving the highest performance, while RS and SB IPs ensured equitable performance at a lower level. D3QN IPs learn to maximize the global and the local rewards by considering the best configuration and the availability of computational resources.




(a)




(b)

**Fig. 5.** Performance comparison in terms of rewards unit and accumulated reward during testing (real-time) episodes in small-scale framework.

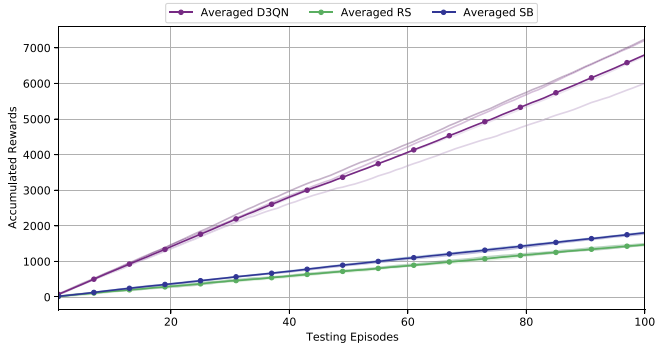
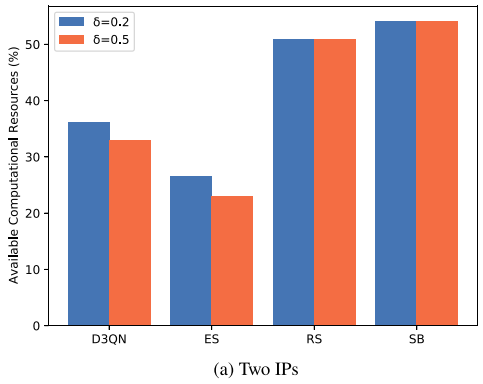
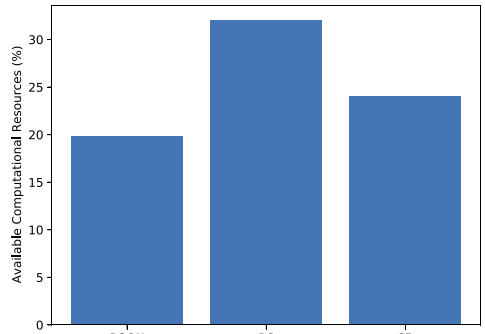



**Fig. 6.** Performance comparison in terms of accumulated reward during testing (real-time) episodes in medium-scale framework.




(a) Two IPs




(b) Four IPs

**Fig. 7.** Averaged available computational resources during testing (real-time) episodes for all policies.

### 5.3.3. *Effective utilization of resources: Global reward analysis*

Fig. 7 represents the utilization of computational resources while testing for small and medium scales. Our main aim is to maximize the utilization of resources considering the application-level requirements. In Fig. 7(a), the effect of changing the weighting factor assigned for the global reward reflects the utilization of the computational resources in the system. At the default setting where weighting factor ($\delta$) equals 0.2, D3QN IPs could effectively coordinate between them to maximize the usage of all available resources, unlike RS, and SB policies, whose resource utilization was minimal. Meanwhile, ES maintained an average utilization of resources but an overall low reward. In the case of increasing the weighting factor ($\delta$) to 0.5, reflecting the need to minimize the percentage of unused resources in the system, the behavior of RS and SB did not change as they neglected the system constraints. However, D3QN outperformed ES with effective utilization of 5.3%. In the medium-scale framework, as shown in Fig. 7(b), the quantity of computational resources increases as the number of IPs increases. D3QN IPs maintain high performance at medium-scale and outperform RS and SB policies. It is worth mentioning that increasing the number of IPs requires providing new verifiers for the verification process.

### 5.3.4. *MOOP analysis: Local reward analysis*

The local reward ensures optimizing the MOOP task based on the selected transactions' characteristics for each IP. Fig. 8 represents a zoom-in into the local reward in a small-scale framework to analyze the optimization of the three conflicting objectives while changing the weighting factors $\alpha$ and $\beta$ for latency and security. In contrast, the cost weighting factor $\gamma$ is fixed.[4] Following Table 5 in assigning $\alpha$ and $\beta$, D3QN IPs have the best performance among all other strategies for all combinations of different weighting factors. In Fig. 8, assigning the weighting factor to *zero* reflects the need to ignore that objective while optimizing; hence, the reward assigned for that objective is *zero*. On the other hand, *high* means that a big portion of the reward should come from maximizing that objective considering the fixed weighting factor $\gamma$. For latency and cost, the aim is to minimize them; hence the represented values in Fig. 8 are the minimum values resulting from maximizing the minimum (i.e., $L$ not $(1 - L)$) presented in Eq. (15). On the other hand, security should be maximized. Therefore, as shown in Fig. 8(d), D3QN IPs could reach the maximum possible security, although the assigned weighting factor is *low* unlike ES, RS, and SB policies. The performance of D3QN IPs surpasses that of other IPs by a significant margin in terms of security across various blockchain modes. In terms of latency, when compared to the D3QN IPs with the second-highest reward, namely ES IPs, D3QN IPs effectively optimize the framework while prioritizing latency requirements. In the Fair latency configuration, D3QN IPs achieve a latency reduction of 24.14%. In

[4] The weighting factors are application-level parameters decided by the IP

**Table 5**
Blockchain Modes Based on Latency and Security.

| Configuration | Latency ($\alpha$) | Security ($\beta$) |
|---|---|---|
| Fair | Equal | Equal |
| Security oriented | 0 | High |
| Delay sensitive | High | 0 |
| Latency driven | Medium | Low |
| Security driven | Low | Medium |

urgent cases and high latency configurations, the reduction in latency reaches as high as 46.6%. On average, across low and medium scales, D3QN IPs achieve a latency reduction of 29.24%. In contrast, ES IPs consistently provide the same reward irrespective of the framework's requirements. Fig. 9 shows the medium-scale framework with "Fair" setting for latency, security, and cost. The D3QN IPs could meet the framework's objectives; the latency and cost were minimized as they did not reach the local reward's maximum possible value for each objective (33%) while maximizing the security. RS and SB approaches ignore the urgency and security aspects of the agent's transactions. Hence, as shown previously, they reached high latency, cost, and minimum-security levels, reflected in the accumulated reward.

#### 5.3.5. Action time analysis

Fig. 10 provides insight into the average decision-making time of IPs in a small-scale framework under various policies during testing episodes. D3QN IPs are the most efficient, making optimal decisions in a reasonable time. On the contrary, ES IPs, burdened by high computational complexity, necessitate a complete exploration of all possible actions, consuming more than two minutes in a small-scale framework. This computational demand grows exponentially with increasing IPs, particularly in multi-discrete action space, rendering ES impractical for medium-scale frameworks and unreliable in healthcare systems. In contrast, the RS approach is incredibly fast but sacrifices consideration for rewards, subsequent state transmission, application-level requirements, or the number of attempts needed for effective environment resolution, resulting in low cumulative rewards over time. Similarly, the SB approach adopts a static action choice irrespective of the observation. Given the critical importance of making decisions considering simultaneous and future rewards, both RS and SB policies are deemed undesirable in healthcare systems. This deficiency becomes evident in the performance comparison section, highlighting that these three approaches overlook the future impact of decisions on overall system performance.

## 6. Case study

This section presents a case study considering heterogeneous application-level participants, while adapting our proposed approach.

### 6.1. Blockchain participants

In order to validate the effectiveness of the proposed MAD3QN, four types of IPs are introduced, each with different application-level requirements concerning latency and security. Table 6 refers to four accepted error levels when optimizing for latency and security with percentage ranges between 15% to 50%: low, normal, medium, and high. The levels consider the required urgency and security characteristics of the participant's selected transactions. *low* means the error percentage required to be very minimal compared to the achieved, while *high* refers to a high error level compared to *low*, which reflects a lower significance of the objective to the participant.

### 6.2. Reward convergence analysis

Fig. 11 depicts the training reward for the four heterogeneous IPs, where all converged following the exact behavior of the homogeneous

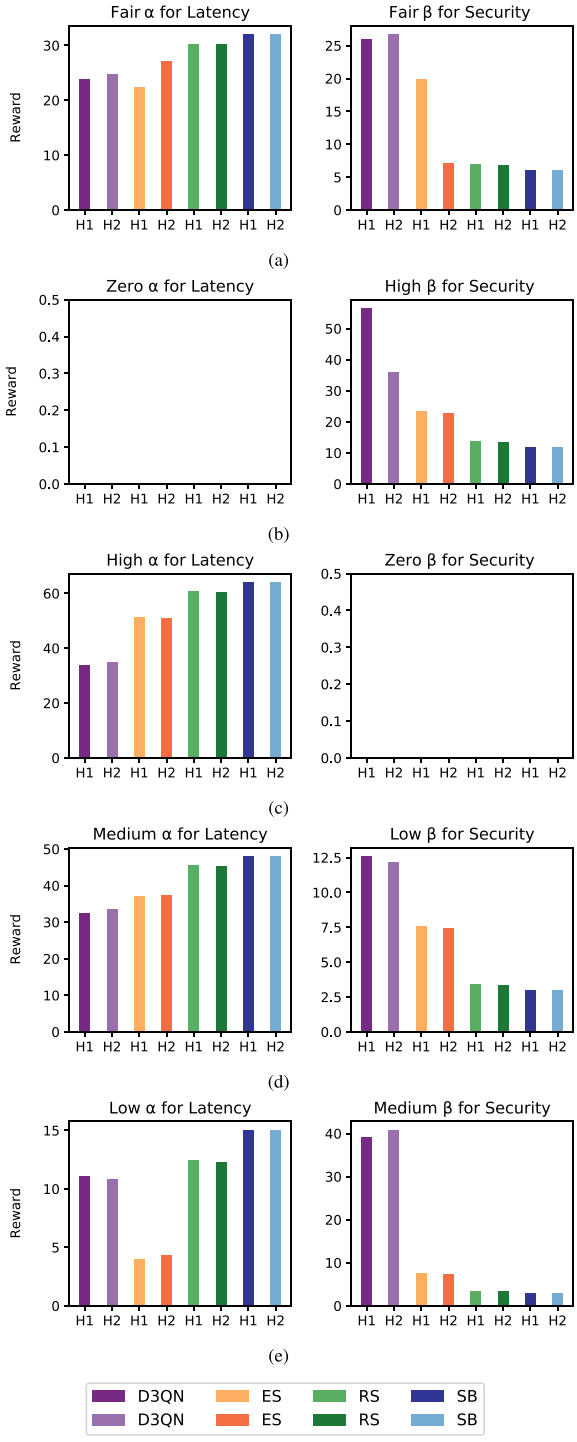



**Fig. 8.** Local reward analysis for two homogeneous IPs during testing episodes for all the policies, considering different weighting factors for security and latency. The $x$-axis represents the IPs, and the $y$-axis represents the averaged accumulated reward. The subtitles reflect the objectives considering the assigned weighting factors.

platform. Moreover, the heterogeneous platform follows the same pattern as the homogeneous for utilizing the available computational resources in the system. The reward values are smoothed over a 100 episode window. Fig. 12 illustrates the averaged accumulated rewards during the testing phase for each approach, where RS and SB did not consider the heterogeneity between the participants, and all were treated the same with low rewards. However, deploying D3QN gives

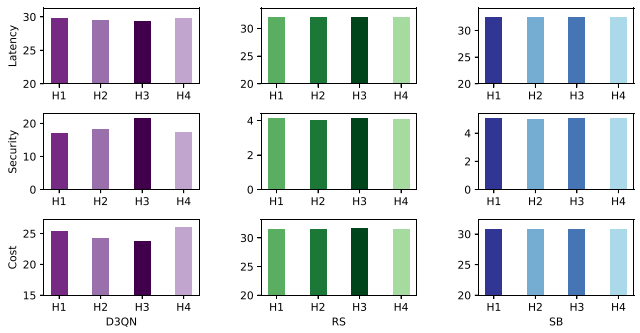



**Fig. 9.** Local reward analysis for four homogeneous IPs during testing (real-time) episodes for all the policies per conflicting objective: latency, security, and cost.

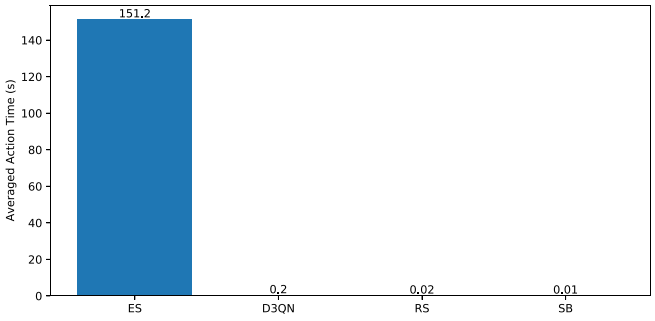



**Fig. 10.** Action time at testing for all policies in small-scale homogeneous framework.

**Table 6**
IP Latency and Security Constraints.

| Agent (IP) | Latency Accepted Error | Security Accepted Error |
|---|---|---|
| Hospital | Low | Normal |
| Insurance Company | Normal | Medium |
| Pharmacy | Normal | High |
| MOPH | High | Low |

| Low | Normal | Medium | High |
|---|---|---|---|
| ± 15% | ± 30% | ± 40% | ± 50% |

a higher reward than expected and considers the overall blockchain network requirements and the participant's needs. Referring to Table 6, the hospital and pharmacy required a combination of normal and extreme cases (either *low* or *high*); hence, they were able to reach higher rewards compared to others. The MOPH comes in third place as it required two extremes (i.e., *high* latency with *low* security), and finally, the moderated participant, the insurance company.

### 6.3. Accepted-error analysis

Each participant in IP-HealthChain has its requirements concerning latency and security optimization objectives. Fig. 13 reflects those requirements during the training and testing phases of the IPs. The proposed framework aims to maximize security while minimizing latency and cost objectives. Therefore, as shown in Fig. 13 during training, the starred security and latency represent the reward following Eqs. (1) and (2) without taking into account the constraints on the accepted error level. In contrast, the non-starred ones consider IPs' constraints by applying the proposed reward function in Eq. (15). In the exploration phase of the training, the reward was penalized highly for the hospital considering latency and for the MOPH considering security as their accepted error levels are minimal. However, after reaching convergence,

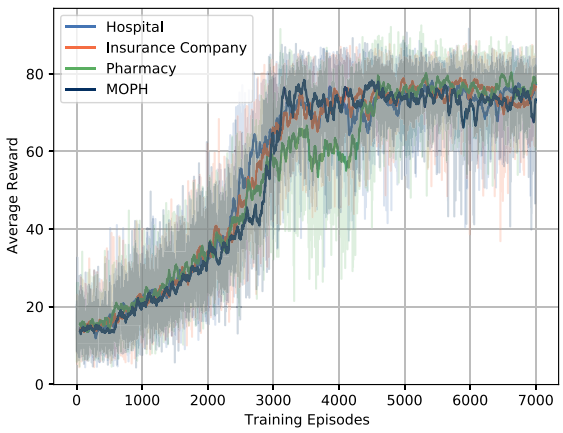



**Fig. 11.** Rewards of IPs over the training episodes.

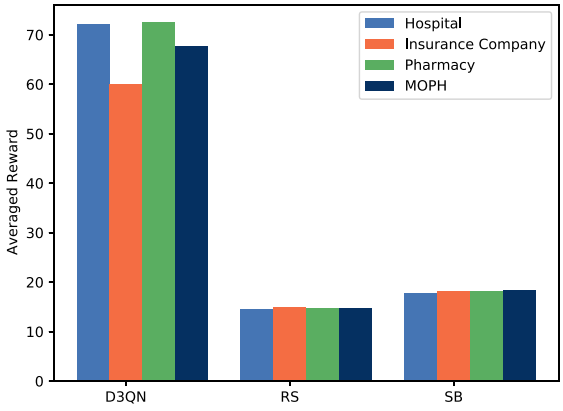



**Fig. 12.** Averaged reward during testing episodes in heterogeneous platform.

the IPs learn to maximize the reward considering their requirements, where both starred, and non-starred objectives achieve high values relatively close to each other. In the testing phase, the D3QN IPs follow the last version of the trained model as no drastic changes occur in their requirements; otherwise, a slight drop may occur. However, they get back to convergence quickly due to the timely synchronization between the online and target network, in addition to implementing the advantage function concept, which ensures stabilization during the learning process.

### 6.4. Action time analysis

Action time is one of the essential factors in healthcare systems, which is elusive in a heterogeneous platform. Each IP has different requirements considering the delivered healthcare service, which can directly reflect the opportunity to save human lives. In IP-HealthChain, it is necessary to consider these differences while deciding on the action since it seeks to be a mixed cooperative-competitive framework between the IPs. In healthcare systems, the temporal aspect of actions is paramount, especially within the intricate landscape of heterogeneity. The unique requirements of each Intelligent Participant (IP) are intricately linked to their healthcare services, with direct implications for potential life-saving opportunities. Within the dynamic framework of IP-HealthChain, the decision-making process regarding actions must meticulously account for these divergent needs, aligning with the system's overarching objective of fostering a mixed cooperative-competitive environment among the IPs. The illustrative Fig. 14 distinctly portrays that the time required to execute an action in a heterogeneous platform is marginally higher than in a homogeneous

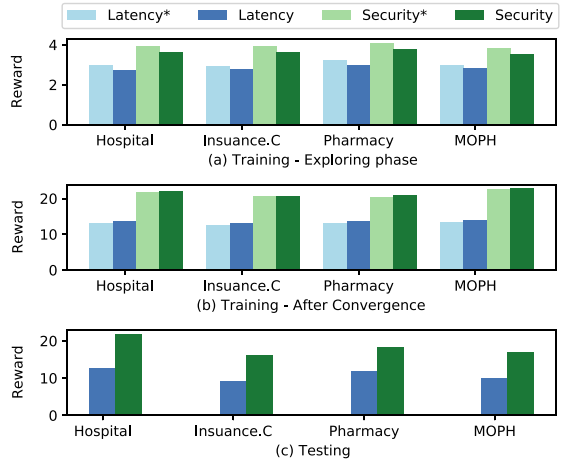



**Fig. 13.** The optimization of latency and security objectives during training and testing phases, where the $y$-axis represents the averaged reward.

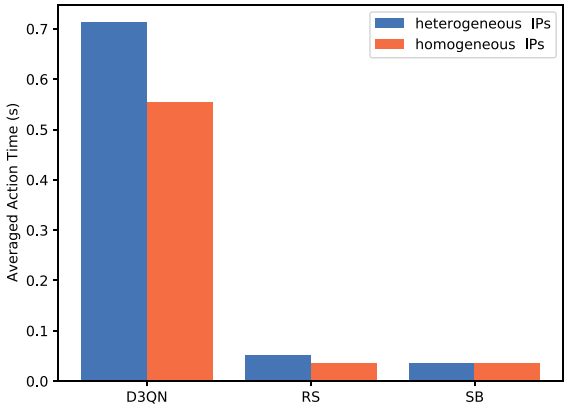



**Fig. 14.** Action time while testing in seconds in medium-scale heterogeneous and homogeneous platforms.

setting. However, this incremental difference in action time is accompanied by a notable elevation in performance concerning optimization when juxtaposed with alternative policies. This nuanced observation underscores the delicate balance between the temporal efficiency of actions and the broader optimization goals within healthcare systems' dynamic and varied context.

## 7. Conclusion

This paper introduces the novel IP-HealthChain, an intelligent healthcare system enabling secure and efficient data sharing among heterogeneous healthcare organizations. Our proposed Mixed cooperative-competitive Multi-Agent Dueling Double Deep Q-Network (MAD3QN), empowered by a bipartite-reward function, has demonstrated remarkable advancements in utilizing computational resources, optimizing latency, security, and cost, and maximizing long-term rewards. Our approach has effectively harnessed available resources by considering participants' varying requirements, such as urgency and security. Furthermore, we have introduced a zombie prevention algorithm to manage the starvation problem caused by low-urgency transactions. The presented results validate the superior performance of our MAD3QN approach compared to existing state-of-the-art techniques and highlight the tangible benefits achieved. For instance, our system achieved a 46.6% reduction in latency in emergency cases, resulting in faster decision-making with minimal time. Additionally, we observed a significant cost reduction and long-term reward maximization while maximizing the security at different modes with a substantial margin of difference. While this research provides a robust foundation for the IP-HealthChain system, we acknowledge the need for further evaluation in more extensive healthcare networks. Future research can explore these challenges and focus on extending the application of our concepts, algorithms, and methodologies to other domains within the realm of intelligent healthcare systems. Ultimately, the advancements presented in this paper contribute to the ongoing development of intelligent healthcare systems and promise to improve data sharing, resource utilization, and decision-making in healthcare organizations.

## CRediT authorship contribution statement

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

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Abeer Ziad Al-Marridi reports financial support was provided by Qatar National Research Fund. Amr Mohamed reports financial support and article publishing charges were provided by Qatar National Research Fund. Abeer Al-Marridi reports a relationship with Qatar National Research Fund that includes: funding grants.

## Data availability

No data was used for the research described in the article.

## Acknowledgments

The work of Abeer Z. Al-Marridi is supported by GSRA, Qatar grant GSRA5-1-0326-18026 and the work of Amr Mohamed was partially covered by NPRP, Qatar grant NPRP13S-0205-200265 from the Qatar National Research Fund (a member of Qatar Foundation).This work was jointly supported by Qatar University and the University of Guelph grant IRCC-2023-171. The statement therein are the sole responsibility of the authors.

## References

Abdellatif, A.A., Al-Marridi, A.Z., Mohamed, A., Erbad, A., Chiasserini, C.F., Refaey, A., 2020. ssHealth: Toward secure, blockchain-enabled healthcare systems. IEEE Netw. 1–8.

Al-Marridi, A.Z., Mohamed, A., Erbad, A., 2018. Convolutional autoencoder approach for EEG compression and reconstruction in m-health systems. In: 2018 14th International Wireless Communications Mobile Computing Conference. IWCMC, pp. 370–375.

Al-Marridi, A.Z., Mohamed, A., Erbad, A., 2021a. Reinforcement learning approaches for efficient and secure blockchain-powered smart health systems. Comput. Netw. 197, 108279. http://dx.doi.org/10.1016/j.comnet.2021.108279, URL https://www.sciencedirect.com/science/article/pii/S1389128621003005.

Al-Marridi, A.Z., Mohamed, A., Erbad, A., Guizani, M., 2021b. CAE adaptive compression, transmission energy and cost optimization for m-health systems. In: 2021 IEEE 22nd International Conference on High Performance Switching and Routing. HPSR, IEEE, pp. 1–6.

Alam, T., Ullah, A., Benaida, M., 2023. Deep reinforcement learning approach for computation offloading in blockchain-enabled communications systems. J. Ambient Intell. Humaniz. Comput. 14 (8), 9959–9972.

AlMarridi, A., Kharbach, S., Yaacoub, E., Mohamed, A., 2020. Optimizing energy-distortion trade-off for vital signs delivery in mobile health applications. In: 2020 IEEE 3rd 5G World Forum. 5GWF, pp. 7–12. http://dx.doi.org/10.1109/5GWF49715.2020.9221315.

Arduin, H., Opatowski, L., 2018. SimFI: a transmission agent-based model of two interacting pathogens. In: International Conference on Practical Applications of Agents and Multi-Agent Systems. Springer, pp. 72–83.

Busoniu, L., Babuska, R., De Schutter, B., 2008. A comprehensive survey of multiagent reinforcement learning. IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.) 38 (2), 156–172. http://dx.doi.org/10.1109/TSMCC.2007.913919.

Demirbaga, U., Aujla, G.S., 2022. MapChain: A blockchain-based verifiable healthcare service management in IoT-based big data ecosystem. IEEE Trans. Netw. Serv. Manag. 19 (4), 3896–3907.

Dubey, A.K., Ramanjaneyulu, N., Saraswat, M., Brammya, G., Govindasamy, C., Ninu Preetha, N., 2023. HECC-ABE: A novel blockchain-based IoT healthcare data storage using hybrid cryptography schemes with key optimization by hybrid meta-heuristic algorithm. Trans. Emerg. Telecommun. Technol. 34 (10), e4839.

Gupta, R., Kumari, A., Tanwar, S., 2021. Fusion of blockchain and artificial intelligence for secure drone networking underlying 5G communications. Trans. Emerg. Telecommun. Technol. 32 (1), e4176.

Hathaliya, J., Sharma, P., Tanwar, S., Gupta, R., 2019. Blockchain-based remote patient monitoring in healthcare 4.0. In: 2019 IEEE 9th International Conference on Advanced Computing. IACC, IEEE, pp. 87–91.

He, Y., Wang, Y., Qiu, C., Lin, Q., Li, J., Ming, Z., 2020. Blockchain-based edge computing resource allocation in IoT: a deep reinforcement learning approach. IEEE Internet Things J. 8 (4), 2226–2237.

Hussien, H.M., Yasin, S.M., Udzir, S., Zaidan, A.A., Zaidan, B.B., 2019. A systematic review for enabling of develop a blockchain technology in healthcare application: taxonomy, substantially analysis, motivations, challenges, recommendations and future direction. J. Med. Syst. 43 (10), 1–35.

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

Kang, J., Xiong, Z., Niyato, D., Ye, D., Kim, D.I., Zhao, J., 2019. Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory. IEEE Trans. Veh. Technol. 68 (3), 2906–2920.

Kumari, A., Gupta, R., Tanwar, S., Kumar, N., 2020. A taxonomy of blockchain-enabled softwarization for secure UAV network. Comput. Commun. 161, 304–323.

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

Li, Z., Xu, M., Nie, J., Kang, J., Chen, W., Xie, S., 2020a. NOMA-enabled cooperative computation offloading for blockchain-empowered Internet of Things: A learning approach. IEEE Internet Things J. 8 (4), 2364–2378.

Li, M., Yu, F.R., Si, P., Wu, W., Zhang, Y., 2020b. Resource optimization for delay-tolerant data in blockchain-enabled IoT with edge computing: A deep reinforcement learning approach. IEEE Internet Things J. 7 (10), 9399–9412.

Lin, P., Song, Q., Yu, F.R., Wang, D., Guo, L., 2021. Task offloading for wireless VR-enabled medical treatment with blockchain security using collective reinforcement learning. IEEE Internet Things J. 8 (21), 15749–15761.

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

Liu, L., Li, Z., 2022. Permissioned blockchain and deep reinforcement learning enabled security and energy efficient healthcare internet of things. IEEE Access 10, 53640–53651. http://dx.doi.org/10.1109/ACCESS.2022.3176444.

Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I., 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in neural information processing systems. Vol. 30.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

Mohammed, A., Nahom, H., Tewodros, A., Habtamu, Y., Hayelom, G., 2020. Deep reinforcement learning for computation offloading and resource allocation in blockchain-based multi-UAV-enabled mobile edge computing. In: 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing. ICCWAMTIP, IEEE, pp. 295–299.

Nakamoto, S., 2008. Bitcoin: A peer-to-peer electronic cash system. Decentralized Bus. Rev. 21260.

Nguyen, D., Ding, M., Pathirana, P., Seneviratne, A., Li, J., Poor, V., 2021. Cooperative task offloading and block mining in blockchain-based edge computing with multi-agent deep reinforcement learning. IEEE Trans. Mob. Comput..

Oliehoek, F.A., Amato, C., 2015. A Concise Introduction to Decentralized POMDPs. Springer.

Peng, H., Shen, X., 2020. Multi-agent reinforcement learning based resource management in MEC-and UAV-assisted vehicular networks. IEEE J. Sel. Areas Commun. 39 (1), 131–141.

Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., Whiteson, S., 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 4295–4304.

Research, V.M., 2022. Blockchain in healthcare market to reach $1189.8 million by 2028. GlobeNewswire News Room. Vantage Market Research. URL https://www.globenewswire.com/news-release/2022/01/25/2372143/0/en/Blockchain-in-Healthcare-Market-to-Reach-1189-8-Million-by-2028-Increasing-Cases-of-Healthcare-Data-Breaches-to-Drive-the-Market-Growth-Exclusive-Report-by-Vantage-Market-Research.html.

Roehrs, A., da Costa, C.A., da Rosa Righi, R., da Silva, V.F., Goldim, J.R., Schmidt, D.C., 2019. Analyzing the performance of a blockchain-based personal health record implementation. J. Biomed. Inf. 92, 103140.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Scherer, M., 2017. Performance and scalability of blockchain networks and smart contracts.

Son, K., Kim, D., Kang, W.J., Hostallero, D.E., Yi, Y., 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 5887–5896.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.

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

Tanwar, S., Bhatia, Q., Patel, P., Kumari, A., Singh, P.K., Hong, W.-C., 2020a. Machine learning adoption in blockchain-based smart applications: The challenges, and a way forward. IEEE Access 8, 474–488. http://dx.doi.org/10.1109/ACCESS.2019.2961372.

Tanwar, S., Parekh, K., Evans, R., 2020b. Blockchain-based electronic healthcare record system for healthcare 4.0 applications. J. Inf. Secur. Appl. 50, 102407.

Tanwar, S., Parekh, K., Evans, R., 2020c. Blockchain-based electronic healthcare record system for healthcare 4.0 applications. J. Inf. Secur. Appl. 50, 102407. http://dx.doi.org/10.1016/j.jisa.2019.102407, URL https://www.sciencedirect.com/science/article/pii/S2214212619306155.

Uppal, S., Kansekar, B., Meher, P., Mini, S., Tosh, D., 2021. CareBlocks: A blockchain-based health information sharing framework for medical IoT. In: 2021 8th International Conference on Signal Processing and Integrated Networks. SPIN, IEEE, pp. 928–933.

Uppal, S., Kansekar, B., Mini, S., Tosh, D., 2023. HealthDote: A blockchain-based model for continuous health monitoring using interplanetary file system. Healthc. Anal. 3, 100175.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N., 2016. Dueling network architectures for deep reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 1995–2003.

Weng, Y., Chu, H., Shi, Z., 2021. An intelligent offloading system based on multiagent reinforcement learning. Secur. Commun. Netw. 2021.

Wu, X., Li, J., Xiao, M., Ching, P., Poor, H.V., 2021. Multi-agent reinforcement learning for cooperative coded caching via homotopy optimization. IEEE Trans. Wireless Commun. 20 (8), 5258–5272.

Zhu, T., Li, K., Herrero, P., Georgiou, P., 2020. Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation. IEEE J. Biomed. Health Inf. 25 (4), 1223–1232.

**Abeer Z. Al-Marridi** obtained a Ph.D. in computer engineering with distinction from Qatar University (Qatar) in 2023. Dr. Al-Marridi's research interests span various cutting-edge areas, including Deep Learning, Reinforcement Learning, Edge Computing, Blockchain, Resource Optimization for Wireless Sensor Networks, and Smart Health Systems. Her expertise lies in leveraging these technologies to address complex challenges and drive innovation in different domains, accommodating the needs. She holds a master's degree in computing from Qatar University, where her thesis focused on Optimal Resource Allocation Using Deep Learning-Based Adaptive Compression for m-Health Applications. This work exemplifies her commitment to utilizing advanced techniques to optimize resource allocation and enhance the performance of healthcare systems. Her research received funding from the Qatar National Research Fund, and his research outcomes were published in respected international conferences and journals. Dr. Al-Marridi is an active member of Arab Women in Computing (ArabWIC), an organization dedicated to empowering and promoting women in the field of computing. She actively contributes to fostering diversity and inclusivity within the computing community through her participation.

**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. He has over 25 years of experience in IoT, edge computing, pervasive AI, and 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 three international journals, has served as a guest editor in several special issues, and has served as a technical program committee (TPC) and co-chair for many IEEE conferences and workshops.

**Aiman Erbad** is an Associate Professor and ICT Division Head in the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). Prior to this, he was an Associate Professor at the Computer Science and Engineering (CSE) Department and the Director of Research Planning and Development at Qatar University until May 2020. He also served as the Director of Research Support responsible for all grants and contracts (2016–2018) and as the Computer Engineering Program Coordinator (2014–2016). Dr. Erbad obtained a Ph.D. in Computer Science from the University of British Columbia (Canada) in 2012, a Master of Computer Science in embedded systems and robotics from the University of Essex (UK) in 2005, and a B.Sc. in Computer Engineering from the University of Washington, Seattle in 2004. He received the Platinum award from H.H. The Emir Sheikh Tamim bin Hamad Al Thani at the Education Excellence Day 2013 (Ph.D. category). He also received the Best Research Paper Award from 3 conferences. His research received funding from the Qatar National Research Fund, and his research outcomes were published in respected international conferences and journals. He is an editor for KSII Transactions on Internet and Information Systems, an editor for the International Journal of Sensor Networks (IJSNet), and a guest editor for IEEE Network. His research interests span cloud computing, edge intelligence, Internet of Things (IoT), private and secure networks, and multimedia systems. He is a senior member of IEEE and ACM.