Blockchain-Based Resource Trading in Multi-UAV-Assisted Industrial IoT Networks: A Multi-Agent DRL Approach

Abegaz Mohammed Seid, *Member, IEEE*, Hayla Nahom Abishu, Yasin Habtamu Yacob, Tewodros Alemu Ayall, Aiman Erbad, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—With the Industrial Internet of Things (IIoT), mobile devices (MDs) and their demands for low-latency data communication are increasing. Due to the limited resources of MDs, such as energy, computation, storage, and bandwidth, IIoT systems cannot meet MDs' quality of service (QoS) and security requirements. Recently, UAVs have been deployed as aerial base stations in the IIoT network to provide connectivity and share resources with MDs. We consider a resource trading environment where multiple resource providers compete to sell their resources to MDs and maximize their profit by continually adjusting their pricing strategies. Multiple MDs, on the other hand, interact with the environment to make purchasing decisions based on the prices set by resource providers to reduce costs and improve QoS. We propose a novel intelligent resource trading framework that integrates multi-agent deep reinforcement Learning (MADRL), blockchain, and game theory to manage dynamic resource trading environments. A consortium blockchain with a smart contract is deployed to ensure the security and privacy of the resource transactions. We formulated the optimization problem using a Stackelberg game. However, the formulated optimization problem in the multi-agent IIoT environment is complex and dynamic, making it difficult to solve directly. Thus, we transform it into a stochastic game to solve the dynamics of the optimization problem. We propose a dynamic pricing algorithm that combines the Stackelberg game with the MADRL algorithm to solve the formulated stochastic game. The simulation results show that our proposed scheme outperforms others to improve resource trading in UAV-assisted IIoT networks.**

***Index Terms*—Blockchain, DRL, industrial IoT, resource trading, unmanned aerial vehicles.**

Manuscript received 18 February 2022; revised 15 June 2022; accepted 27 July 2022. Date of publication 9 August 2022; date of current version 7 March 2023. This work was made possible by NPRP-Standard (NPRP-S) Thirteen (13th) Cycle grant NPRP13S-0205-200265 from the Qatar National Research Fund. The findings achieved herein are solely the responsibility of the authors. The associate editor coordinating the review of this article and approving it for publication was J. Zhang. *(Corresponding author: Aiman Erbad.)*

Abegaz Mohammed Seid and Aiman Erbad are with the Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar (e-mail: mamsied2002@gmail.com; aerbad@hbku.edu.qa).

Hayla Nahom Abishu and Yasin Habtamu Yacob are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.

Tewodros Alemu Ayall is with the Department of Computer Science, Zhejiang Normal University, Jinhua 321004, Zhejiang, China.

Mohsen Guizani is with the Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE.

Digital Object Identifier 10.1109/TNSM.2022.3197309

I. Introduction

OVER the previous decade, the Internet of Things (IoT) revolution has had a significant impact on manufacturing, energy, agriculture, transportation, and other industrial sectors [1]. The Industrial IoT (IIoT) is an industry-specific variant of the IoT, which provides an impressive potential for businesses via connected machines, sensors, and applications [2], [3]. It is one of the most exciting technologies now reshaping industrial enterprises, prompting them to modernize their processes, system intelligence, and facilities in order to cope with emerging disruptive technologies. The IIoT improves manufacturing efficiency, safety, scalability, production time, and profitability in the industrial sector [4]. In this regard, beyond fifth-generation (B5G) networks are envisioned to provide factory platforms capable of realizing mission-critical use-cases in an industrial setting with low latency, high reliability, and the capacity to accommodate a much larger number of IIoT devices (IIoTD) [5]. It promises to improve factory automation by supporting ultra-reliable low latency communication (uRLLC), enhanced machine type communication (eMTC), and enhanced mobile broadband (eMBB). Thus, current industry systems connect more sophisticated mobile devices (MDs), such as sensitive and precise sensors and location-aware technologies, to develop integrated manufacturing and supply chain monitoring. These MDs run mission-critical and ultra-low latency applications/services like online gaming, augmented/virtual reality, and image/video encoding to provide broader, advanced, and automated controls in industries like aerospace, defense, healthcare, and energy [6], [7], [8]. However, most of MDs connected to the IIoT ecosystem (e.g., sensors, actuators, and so on) have resource constraints such as computation, battery, spectrum, and storage in order to communicate efficiently in the system [9], [10].

Recently, many researchers have attempted to propose solutions for resource trading in IIoT networks by integrating blockchain and B5G enabler technologies such as software-defined industry automation networks (SDIAN) [11], network function virtualization (NFV), and smart contract (SC) [12]. The integration of B5G systems, mobile edge computing (MEC), and network slicing has been introduced to meet MDs' ultra-low latency response and quality of service (QoS) requirements [13], [14]. Blockchain is a technology that uses

1932-4537 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

distributed consensus and cryptographic hashing to keep digital assets secure and transparent [15]. It is being utilized in various industries to guarantee the security and privacy of transactions between untrusted nodes [16], [17]. Besides, the MEC paradigm has been given great attention to solve the computational and storage issues of MDs. The MEC operates as a new model to provide computing resources for MDs. It allows MDs to offload their computation tasks to a nearby edge nodes/MEC servers for processing. The MEC servers then allocate resources, perform the requested computation tasks, and return the results to MDs in a significantly shorter time frame. This could help MDs overcome their computing resource restrictions and improve the user experience [18]. When there is a system failure, or MDs are moved out of coverage, the execution time of each task can be slowed, lowering the QoS. Thus, the B5G network has recently deployed unmanned aerial vehicles (UAVs) as aerial base station (ABS) equipped with MEC to provide computation offloading [19], resource allocation, and enhanced coverage or relaying services to MDs in wireless systems with limited or no physical infrastructure coverage [20], [21], [22], [23], [24]. Besides, UAVs are actively used in mission-critical services such as military, emergency communication, and health care due to their low implementation cost, short-range line-of-sight connection, and capacity to execute jobs like delivery services, disaster relief, and agriculture applications that humans cannot easily perform [25], [26], [27]. These UAVs provide flexible short-distance wireless communication, allowing the collection and dissemination of information at a minimal cost. Moreover, the rapid deployment and relocating capabilities of UAVs also enable them to automatically adapt to dynamic and emerging communication requirements, improving fault tolerance and resilience in IIoT systems.

Furthermore, several studies have proposed solutions for resource trading in IIoT networks to address the resource limitations of MDs. Traditional resource-sharing approaches cannot achieve the desired performance due to the dynamic nature of the IIoT environment and the resource limitations of MDs. Moreover, the IIoT mandates that resource sharing policies be intelligent enough to make intelligent resource access decisions [28], [29]. In this regard, machine learning (ML) is one of the most powerful data-driven approaches for enabling intelligent decision-making by using a massive amount of data from multiple heterogeneous IIoT devices [30]. In recent years, researchers have applied ML approaches to the problem of dynamic resource allocation and sharing, such as single-agent reinforcement learning (RL) and deep reinforcement learning (DRL) [31], [32], [33], [34]. However, in a complex and multi-agent system, a single agent RL/DRL approach does not achieve various optimization problems [35]. Thus, multi-agent reinforcement learning algorithms (MARL) have been proposed for finding optimal decision-making policies for the pricing and resource management problem between cloud providers and miners [35]. Nevertheless, the existing resource trading solutions are still insufficient to address the ultra-reliable, low latency, and differentiated service requirements of MDs connected to the IIoT system, particularly in addressing the spectrum and energy constraints.

We are motivated to address the issues with MDs in the IIoT system mentioned above. In this work, we propose a DRL approach combined with game theory for blockchain-enabled resource trading scheme in which UAVs are deployed as ABS to lease resources to MDs associated to the IIoT network. Because of its ability to handle a wide range of complex decision-making tasks that were previously unreachable for a machine to address real-world problems with human-like intelligence, the DRL is preferred over other ML approaches. UAVs have recently been widely deployed as ABS in the B5G system to assist MDs connected to IIoT networks in computing [36], resource sharing, and expanding access to areas outside of the physical infrastructure coverage [21], [22], [23], [24]. The consortium blockchain is used in combination with SC to ensure the security and privacy of resource transactions. A two-stage Stackelberg game model is adopted and integrated with the DRL method to find the optimal pricing policy for buyers and sellers in a resource trading environment where there are multiple buyers and sellers. In the trading process, the multi-agent deep deterministic policy gradient (MADDPG) is used for intelligent decision-making policy to maximize the benefits for both MDs and UAVs. The UAVs affordably lease spectrum and energy resources to the MDs via wireless communication systems, and the MDs can then efficiently interact in the network to improve the performance of the industrial system. It can significantly improve the performance of the IIoT communication system. The main contributions of this paper are as follows:

- We propose a novel resource trading framework that integrates multi-agent DRL (MADRL) with consortium blockchain and the Stackelberg game. In this framework, UAVs act as ABS to lease resources such as spectrum and energy to the MDs deployed in the IIoT ecosystem. We formulate utility maximization problem as a two stage multi-leader-multi-followers (MLMF) Stackelberg game to allow resource sellers/UAVs and resource buyers/MDs to maximize the aggregate reward and improve resource trading efficiency.
- We model the optimization problem as an extended Markov decision process (MDP)/stochastic game to handle the dynamic resource trading problems of multi-UAV-assisted IIoT networks, in which each UAV and MD acts as a learning agent and each resource trading solution corresponds to a UAV and MD action.
- We adopt a dynamic pricing algorithm that combines the Stackelberg game with MADDPG algorithm, namely, Stackelberg MADDPG (SMADDPG) to solve the formulated stochastic game of multi-UAV-assisted IIoT networks. It allows UAVs to integrate spectrum and energy strategic planning to increase their utilities while meeting the QoS requirements of various MDs.
- Extensive simulations are conducted to demonstrate the efficiency of our proposed framework. The numerical analysis of these simulation results proves that the proposed model is better than the baseline schemes.

The rest of the paper is organized as follows: Section II summarizes related studies. The system framework used in resource trading is described in Section III, and Section IV

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

presents the optimization problem formulation. Further, we present the MADRL approach to solve the optimization problem in Section V. Section VI presents the simulation results and analysis. Finally, in Section VII, we conclude our work.

II. Related Work

Recently, various studies have attempted to propose a solution for the resource constraints of MDs connected to the IoT and IIoT networks. In the IoT era, the MEC is a viable paradigm for resolving MDs' computational and resource allocation difficulties by proximate resources at the edge level. The MEC provides the services required of high data rate and high computation capability [37]. Many previous research works proposed optimization methods for task offloading in MEC systems [38], energy consumption optimization, devices service response latency [39]. Naram *et al.* [40] proposed on designing intelligent agents to provide services in blockchain-enabled systems. The authors propose lightweight online learning SCs that can optimize performance using DRL agents. The authors in [41] proposed a blockchain-based peer-to-peer computation resource trading for edge-cloud computing IoT, where a broker manages the computing resource trading among the sellers and buyers. Yao *et al.* [35] presented a decentralized self-organized trading platform for IIoT devices, focusing on the resource management and pricing problem between the cloud providers and miners. In this work, cloud mining is used to help MDs offload massive computational work to the cloud provider, and a multi-agent RL algorithm is introduced for finding optimal decision-making policies. This work tries to address MDs' computational limitations while also enhancing the IIoT system's efficiency through resource trading. In [42], the authors proposed a futures-based resource trading strategy to address the risk of trading failure and the unfair on-site negotiating process in wireless networks to share resources. Based on historic resource supply and demand facts, the resource requester and resource supplier agree on a mutually advantageous forward contract in advance.

Moreover, the authors in [43] presented a contract-based cooperative spectrum sharing system to solve the bandwidth limitation of MDs. In addition, a cooperative relaying technique was devised that uses superposition coding at both cellular and device-to-device (D2D) transmitters. This maximizes the profit of cellular links while maximizing transmission chances for D2D lines. The authors of [44] proposed a bandwidth-auction game-based spectrum trading system in which base stations (BSs) can sell or share spectrum with the D2D pair, allowing the D2D pair to operate in orthogonal or non-orthogonal sharing. The D2D pair can buy the appropriate spectrum from a variety of service providers based on the frequency spectrum of each mode. Qiu *et al.* [45] proposes a secure blockchain-based spectrum trading system in which UAVs purchase spectrum bands from service providers/owners of spectrum resources. In [46] presented a novel distributed spectrum trading protocol based on blockchain, which focuses on efficiency, simplicity, safety, and energy-saving in IoT networks. Besides, a blockchain-enabled secure power trading method is proposed for the smart grid using wireless networks in [47]. Lin *et al.* [48] proposed a novel wirelessly powered edge intelligence framework that uses energy harvesting approaches to generate a stable, reliable, and sustainable edge intelligence. Baig *et al.* [49] introduced an IoT-based energy trading system linked to blockchain-based financial transactions. The authors of [50] presented an optimal contract-based electricity trading method that increases profit effectively. Nguyen *et al.* [51] proposed an economic model to jointly optimize revenues of energy service providers and IoT service providers participating in a heterogeneous IoT wireless-powered backscatter communication network.

The above-mentioned resource trading-related studies do not adequately address the challenges of MDs' energy and spectrum resource limitations, and do not take into account the mobility of MDs in the IIoT networks. In this work, we propose an intelligent resource trading framework that integrates blockchain technology, DRL, game theory, and SCs. In the UAV-assisted IIoT system, the ABS can serve MDs moving outside physical infrastructure coverage area and when the physical infrastructure fails due to natural or human-made disasters. Therefore, MDs can buy energy and spectrum resources from UAVs while in motion to continue performing in the IIoT system. This can improve the reliability and consistency of IIoT systems. Despite the benefits of integrating the B5G key enabler technologies, such as blockchain, AI, and game models in IIoT networks, their inappropriate implementation can lead to inefficient resource utilization, performance degradation, and security breaches. These challenges require careful consideration when integrating these technologies. In this regard, we integrate these technologies carefully to improve the performance of the IIoT network. We use consortium blockchain for better security and efficient resource utilization because it is a lightweight platform that allows only authorized nodes to join the system. In addition, we applied a multi-agent DRL algorithm combined with a stochastic game model to handle the dynamics and complexity of resource trading and optimization problems arising due to the heterogeneity and diversified service requirements of devices. Furthermore, edge nodes are in charge of the consensus process to assist devices with limited resources.

III. System Model

We consider UAV-assisted IIoT network which consists of macro BSs (MBS), UAVs, MEC servers, and heterogeneous MDs, as shown in Fig. 1. The MBSs are deployed to provide wireless connectivity for the UAVs and MDs in the physical infrastructure coverage area. In addition, UAVs provide wireless connectivity to devices outside of the physical network coverage and provide access to resources for MDs. The MDs connected to the UAVs can request spectrum, energy, or both resources to meet their needs. The MEC performs resource-intensive and latency-sensitive tasks, including block mining and storage to assist resource-limited nodes. Software-defined networking (SDN) is a network architecture that allows for the creation of programmable resource-trading connectivity services, which can dynamically manage and direct

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

TABLE I
LIST OF NOTATIONS

Notation	Description
$\mathcal{U}$	Set of UAVs
$\mathcal{D}$	Set of MDs
$\mathcal{Z}$	Number of clusters
$\mathcal{C}$	The set of sub-channels
(x_0, y_0, H_0)	The location coordinate of BS
$(x_{i,z}, y_{i,z}, H_{i,z})$	The location coordinate of UAV i
(x_j, y_j)	Location coordinate of MD j
$H_{i,z}$	The antenna height UAV i
H_0	The antenna height of BS
$\alpha_{i,j}$	Euclidean distance between UAV i and MD j
$SINR(i,j)$	The SINR of j^{th} MD associated with the i^{th} UAV
θ	Resource purchased by MDs
β	Unit price of resource

traffic flows for maximum performance gain [52]. The central SDN controller has a global and aggregated view of the trading network. Cloud computing utilizes virtual machines with high-performance computing and storage capabilities to solve complicated computation tasks. It consists of blockchain and DRL applications, which communicate securely with an SDN controller. The certificate authority (CA) is responsible for issuing identity certificates to entities participating in the resource trading system. The CA manages the network infrastructure and ensures its security and privacy.

It maintains the list of registered users with their associated SCs, and the SC is used to enforce compliance with the agreements signed by the entities. Moreover, heterogeneous IIoT devices such as autonomous industrial machines, mobile vehicles, robots, and management systems are connected via ubiquitous connectivity to control and manage industrial machinery in real-time. These devices can exchange their sensed data/knowledge for efficient production.

A. Business Model

We present a business model that consists of MDs, UAV operators (UAVOs), and an infrastructure provider/primary resource provider (PRP) to promote resource trading. These entities have the following business relationship; PRPs own the infrastructure and lease their resources (spectrum and energy) to UAVOs on a service level agreement (SLA) basis. Likewise, depending on MD's demand, the UAVOs sell these resources to MDs at a reasonable price. In addition, the UAVs are equipped with solar energy receivers, allowing them to harvest solar energy and serve as alternative energy sources [53]. Due to the broadcast nature of the wireless environment and multiple untrusted resource sellers and purchasers, it is resulting in privacy breaches, double-spending attacks, and overall system vulnerability in the real-time resource trading process. Therefore, we used a consortium blockchain with SC to establish a decentralized, transparent, and trusted environment for entities who seek to trade resources. In this work, the trading entities are the UAVOs and MDs, where UAVOs act as a resource provider and MDs act as resource requester. The resource provider/seller and resource requester/buyer need to contract SLA to build a harmonious environment for resource trading. In this regard, the buyers and sellers deploy SCs that will self-execute when the parties achieve all contractual conditions, ensuring real-time renegotiation of agreements when network traffic changes rapidly. The SC can easily manage the agreements between UAVOs and MDs involved in the resource trading.

B. Network Model

In this paper, we consider the IIoT network with MDs that need a variety of services and connected to ABS. Multiple UAVs are deployed as ABS to provide resources and network coverage for MDs outside of the physical infrastructure coverage or the terrestrial cellular network is out of service due to natural or human-made disaster, ensuring network stability. A group of UAVs can act as ABS to relay between the central network controller (CNC) and IIoT networks. Let $i \in \mathcal{U} = \{1, 2, \ldots, U\}$ and $j \in \mathcal{D} = \{1, 2, \ldots, D\}$ denotes the set of UAVs and MDs connected to the IIoT network, respectively. The MDs in the network are distributed at random and organized into *Z* clusters. We use *i* UAVs that are cellularly-connected to the core network to assist *Z* clusters of MDs in the IIoT network coverage area. Both UAVs and MDs are equipped with antennas to establish a wireless connection between them [54]. We consider three data transmission links at any given time slot *t*: UAV to MD (U2M), UAV to UAV (U2U) in a single cluster, and UAV to core (U2C). The multi-UAV network and the terrestrial IIoT network are managed by a single CNC. Each cluster in the network coverage can serve a finite number of MDs with the set of $D_z = \{1, \ldots, D_z\}$, where $z \in \mathcal{Z} = \{1, .., Z\}$. The MD (*z*, *j*) denotes the j^{th} MD in cluster *z*. The BS, ABS, and MDs are all assumed to be in 3D space, and their locations coordinate can be expressed as (x_0, y_0, H_0), $(x_{i,z}, y_{i,z}, H_{i,z}), \forall i \in \mathcal{U}$, and $(x_j, y_j), \forall j \in \mathcal{D}$, respectively, where $(x_{i,z}, y_{i,z})$ denotes location of i^{th} ABS in the *z* cluster. The antenna heights of the BS and the UAV are denoted as H_0 and $H_{i,z}$, respectively. The distance between the BS and the i^{th} UAV is then expressed as [55]:

$$\mathrm{L}_{0,z} = \sqrt{\left(x_{i,z} - x_0\right)^2 + \left(y_{i,z} - y_0\right)^2 + \left(H_{i,z} - H_0\right)^2}. \quad (1)$$

In the same way, the distance between the UAV *i* and the MD *j* in its cluster is defined by:

$$\mathrm{L}_{i,j} = \sqrt{\alpha_{i,j}^2 + H_{i,z}^2}, j \in \mathcal{D}_z, \quad (2)$$

where $\alpha_{i,j} = \sqrt{(x_{i,z} - x_d)^2 + (y_{i,z} - y_j)^2}$ is the Euclidean distance between the UAV *i* and the MD *j*. When positioning the ABS, considering interference from the other co-channels is very important. We consider the impact of cross channel interference (CCI) in the ABS placement and MD association problems [56]. The signal-to-interference-plus-noise ratio (SINR) of j^{th} MD associated with the i^{th} ABS is given as:

$$\lambda(i,j) = \frac{SPR(i,j)}{IE_{Agg}(i) + N_0}, \quad (3)$$

where *SPR*(*ij*) denotes the signal power received at the j^{th} MD from the i^{th} ABS, $IE_{Agg}(i)$ is the aggregate interference encountered by the j^{th} MD, and N_0 refers the power spectral density of the Gaussian noise. The signal power received,

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

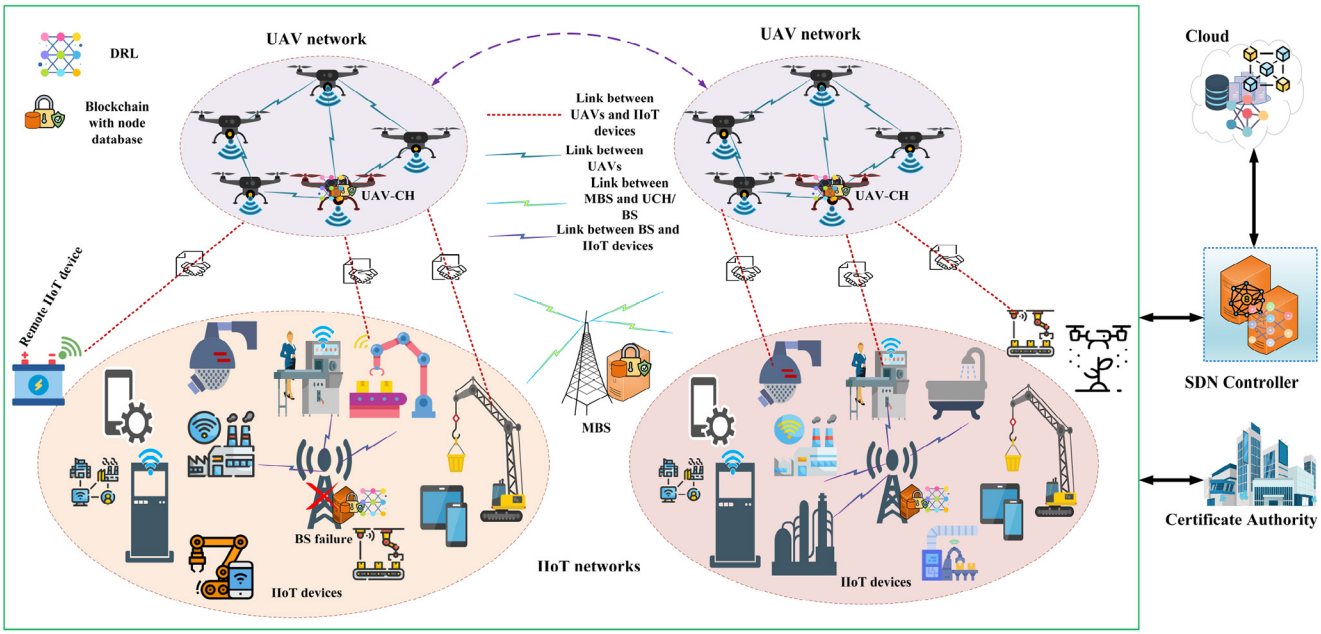


Fig. 1. System architecture.

SPR(*ij*) can be expressed as:

$$SPR(i,j) = p_t^i \Big[P\Big(LoS, \theta_i^j\Big) h_L(i,j) + \Big(1 - P\Big(LoS, \theta_i^j\Big)\Big) h_N(i,j) \Big], \tag{4}$$

where p_t^i is the transmit power of i^{th} ABS and LoS is the line of site. Beside the aggregate CCI, $IE_{Agg}(i)$ is given as:

$$IE_{Agg}(i) = \sum_{l=1, l \neq j}^{U} SPR(i,j), \tag{5}$$

where U is the number of UAVs. The LoS and non-LoS (NLoS) propagation from interfering ABS is included.

C. Communication Model

In this multi-UAV assisted IIoT network scenario, MDs are connected to UAV network through wireless networks. MDs and UAVs use the orthogonal frequency division multiple access for their resource trading activities, in which different MDs occupy different sub-channels for their communications. The U2M communication channel experiences both LoS and NLoS propagation conditions depending on the altitude of the ABSs. Both LoS and NLoS links should be considered to evaluate the realistic performance of the system. The power gain of the U2M channel is determined primarily by the free space path loss model [57], which is given below.

$$\Pi_{i,j}^n = \phi \Big(\alpha_0 / \alpha_{i,j}^n \Big)^2, \tag{6}$$

where ϕ is a constant that varies with antenna physiognomies and frequency, dignified at the reference distance $\alpha_0 = 1\,\mathrm{m}$ and $\alpha_{j,i}^n$ is the square of the Euclidean distance between UAV i and MD j. Let $m \in \mathcal{C} = \{1, 2, \ldots, C\}$ denotes the set of sub-channels available to each UAV during the restoration process. These sub-channels will be subdivided further and assigned to the MDs associated with each UAV. Thus, each UAV i transmits to each MD j at a transmit power of $p_{j,i,m}^n$ per sub-channel. If UAV i is not assigned to sub-channel m, then $p_{j,i,m}^n$ equals zero. As a result, the SINR between UAV i, and MD j per sub-channel m during time block n is as follows:

$$\mathcal{N}_{j,i,m}^n = \frac{p_{j,i,m}^n \Pi_{j,i}^n}{\sum_{j \in D, i \neq j} \sum_{i \in U} p_{i,j,m}^n \Pi_{j,i}^n + \epsilon^2}, \tag{7}$$

where ϵ^2 denotes the power of the Additive White Gaussian Noise at the receiver. Thus, the achievable per sub-channel downlink rate from UAV i to MD j can be calculated as:

$$\gamma_{j,i,m}^n = \log_2 \Big(1 + \mathcal{N}_{j,i,m}^n \Big). \tag{8}$$

D. Blockchain Model

We employ a consortium blockchain integrated with SC, DRL model, and game theory to ensure the security and privacy of the resource trading transactions between the UAVOs and MDs while allowing entities to make intelligent decisions to increase their utilities. A consortium blockchain is a lightweight blockchain that is preferable than the permissionless blockchain for designing a secure and distributed resource trading framework that optimally allocates resources from UAVOs to MDs in the multi-UAV-assisted IIoT system. The consortium blockchain is fast, energy-efficient, and more suitable for resource-constrained devices. SC is used to efficiently manage and ensure that the parties' agreed-upon terms and conditions for satisfying service quality and quantity are met. The operation of the consortium blockchain in our proposed resource trading framework is presented in five phases as in Algorithm 1. The detailed procedures are as follows:

1) System Initialization and Entity Registration: Nodes that wish to engage in trading create an account in the trading system and receive a unique identification that proves they

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

are a legitimate entity to participate as a seller, buyer, or miner/consensus node (lines 1-2). We use an elliptic curve digital signature algorithm [58] and asymmetric cryptography for system initialization in order to guarantee the unforgeability and transparency of data [59]. After registering to the trading system, authorized entities would be issued a certificate by the trusted authority. In the proposed resource trading blockchain, every entity joins the resource trading network with its certificate and obtains public parameters such as public key *(PK)*/private key *(SK)* and wallet address W_{add}. To ensure the security and integrity of message transmission between the senders and receivers, we utilize an asymmetric cryptography scheme, which is expressed as:

$$D_{PK_i}\big(Sig_{SK_i}(H(m))\big) = H(m), \tag{9}$$

where $Sig_{SK_i}()$ is the digital signature of sender i with its private key, $D_{PK_i}()$ is the decryption function with sender i's public key, and $H(m)$ is the hash digest of message m. Once the system is initialized, sellers and buyers interested in trading join the system by presenting their unique identities. The system verifies the participants' identities, sets up the Stackelberg game, and runs SC for trading (lines 4-12).

2) Block Mining: The main process of block mining in IIoT resource trading is adding transaction data to the resource trade blockchain. Edge nodes/consensus nodes are responsible for achieving consensus on resource trading transactions. Upon the initialization of the resource trading system, resource demanding nodes begin sending requests to the nearest edge node. The edge node then collects the requests and uses the collected data to train the DRL model. The resource providers then observe the environment state and audit their supplies, set unit prices, and respond to the edge servers. Based on consumer demand and available supplier resources, the edge server will match demand-supply trade pairings. After that, both consumers and suppliers can complete their transactions.

3) Block Generation: In the consortium blockchain, only some selected nodes can participate in the block generation and verification process. Thus, we use the hybrid of practical byzantine fault tolerant and proof of reputation (PPoR) consensus scheme, which selects miners based on their reputation value. More than three-fourths of nodes with high reputation value would be selected as candidate consensus nodes. And a node with the highest reputation value is taken as the leader of the miners. The leader node collects the transaction from the transaction pool, creates a block, sings it, and broadcasts to the other consensus nodes [60].

4) Consensus Process: After the leader has built and broadcast the block, the miners must audit the newly created block using a consensus mechanism before adding it to the chain. The consensus nodes check the legitimacy of the new block broadcasted by the leader. If it is legitimate, they will start auditing the transactions in the block and broadcast their audit results to other consensus nodes for mutual supervision. They compare their (lines 15-18).

5) Blockchain Update: Finally, the leader collects the consensus nodes' block verification reports. If more than three-quarters of the consensus nodes agree that the block is valid, the leader sings on it, appends the block to the chain, and broadcasts the update to all nodes (lines 19-24).

Algorithm 1 Blockchain-Enabled Resource Trading

```
1: Initialize: Sellers, Buyers
2: Registration and certification of sellers, buyers, consensus nodes
3: for time =1 to T do
4:   for all i, j do
5:     Verify the certificates
6:     if the certificates of sellers, buyers are verified
7:       set-up Stackelberg game as Eqns. (12) and (13)
8:       Run SC for trading and create a block
9:     else
10:      Go back to initialization step
11:    end if
12:  end for
13:  for N consensus nodes do
14:    Leader l broadcast block blk to all consensus nodes
15:    Each node receive the block and check its legality;
16:      if blk is legal then
17:        Audit and send the result to each other
18:        Compare its work with others and sends to the leader
19:        The leader analyze the responses of nodes
20:        if majority of nodes agree then
21:          Append the block to the global chain and broadcast updates to all nodes
22:        else
23:          Discard blk
24:        end if
25:      else
26:        Ignore
27:      end if
28:  end for
29: end for
```

IV. Problem Formulation

In this section, we formulate the utility optimization problem of buyers and sellers as a two-stage MLMF Stackelberg game, a strategic game in which leaders and followers compete for resources, allowing honest nodes to maximize their utility by actively participating in the resource trading [61]. The resource sellers/UAVOs act as leaders, while the buyers/MDs act as followers [62]. The resource seller UAVOs compete with one another to sell their idle resources by continuously observing the environment, setting their resource price, and estimating QoS levels in a non-cooperative manner. The customers/MDs also constantly observe the state of the environment and make a matching decision based on the costs and expected QoS of resource providers. Each leader must determine an appropriate price β in order to maximize utility within the constraints of their resources. Likewise, each follower decides how to maximize their utility (reduce their costs) while achieving a desired QoS. In our scenario, both UAVOs and MDs are the agents who

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

make intelligent decisions to optimize their utilities. The ideal response of each agent is determined using Stackelberg-based MADDPG, a variant of the MADRL algorithm in which each agent interacts with its environment to learn an optimal policy that maximizes its long-term reward without prior knowledge of the actions of others. The Stackelberg leader–follower game is characterized as follows:

1) Players: The UAVOs and MDs are the players/agents of the game, with the UAVOs act as the leader and MDs act as the followers.

2) Strategy: The strategy for each UAVO is to determine the decision strategy Λ and the price of a unit quantity of resource β, where $\beta^e, \beta^s \in \beta = \{\beta_1^e, \beta_1^s, \beta_2^e, \beta_2^s, \ldots, \beta_N^e, \beta_N^s\}$ denote the energy and spectrum prices. The strategy for the MDs is to decide how much resource to buy from each UAVO. Let $\theta^e, \theta^s \in \theta = \{\theta_1^e, \theta_1^s, \theta_2^e, \theta_2^s, \ldots, \theta_N^e, \theta_N^s\}$ denote the energy and spectrum demand of MDs. For all MDs $j \in \mathcal{D}$, we define resource request as $rr_j(t) \in \{-1, 0, 1\}$, where $rr_j(t) = -1$ denotes the MD *j* request spectrum, $rr_j(t) = 0$ denotes the MD *j* request energy, and $rr_j(t) = 1$ denotes the MD *j* request both spectrum and energy resources.

3) Reward: For each UAVO $i \in \mathcal{U}$ and MD $j \in \mathcal{D}$, the payoff functions are shown in equations (10) and (11), respectively. The reward of each UAVO is given as:

$$R_i = \Lambda \sum_{j=1}^{D} \left(\beta_j^e \theta_j^e + \beta_j^s \theta_j^s\right) - \sum_{j=1}^{D} C\left(\theta_j^e + \theta_j^s\right), \quad (10)$$

where $\beta_j^e, \beta_j^s, \theta_j^e$, and θ_j^s denote energy price, spectrum price, energy purchased, and spectrum purchased by MD *j*, respectively. The overhead costs incurred by UAVOs during maintenance, electricity, hardware loss, and operation are denoted by *C*. On the other hand, the MDs wish to maximize their utility by minimizing the cost of resources. The reward of each MD is calculated as:

$$R_j = \omega_j \log_2\left(1 + \frac{\theta_j^e}{\mu_j^e} + \frac{\theta_j^s}{\mu_j^s}\right) - \left(\beta_j^e \theta_j^e + \beta_j^s \theta_j^s\right), \quad (11)$$

where $\omega_j \log_2(1 + \frac{\theta_j^e}{\mu_j^e} + \frac{\theta_j^s}{\mu_j^s})$ is the obtainment gain from the purchased energy and spectrum resources, ω_j is a positive coefficient that is used to convert the obtainment revenues into monetary reward. μ_j^e and μ_j^s are energy and spectrum demand of MD *j*, respectively. To attain optimal utility for both leaders and followers, the optimization problems are formulated as the MLMF Stackelberg game below:

Stage 1: Leaders' pricing

$$\begin{aligned} \max_{\beta} \quad & R_i(\beta, \theta, \Lambda) \\ \text{s.t.} \quad & \beta \geq 0, \\ & \sum_{j=1}^{D} \theta_j^e \leq \chi, \quad \sum_{j=1}^{D} \theta_j^s \leq \chi, \end{aligned} \quad (12)$$

where χ is the total available idle resource of the UAVO.

Stage 2: Followers' purchasing

$$\begin{aligned} & \max_{\theta} R_j(\theta, \mu, \Lambda) \\ \text{s.t.} \quad & \theta^e, \theta^s, R_j \geq 0. \end{aligned} \quad (13)$$

This game aims to find the Nash equilibrium (NE) point(s) where both the leader and the followers benefit the most. Equations (12) and (13) form the Stackelberg game. We define the NE as follows:

Definition 1: Let β^*, Λ^*, and θ^* are optimal price, decision strategy of UAVOs, and resource demand of MDs, respectively. The point $(\beta^*, \Lambda^*, \theta^*)$ is the NE if it satisfies the following conditions.

$$R_i(\beta^*, \Lambda^*, \theta^*) \geq R_i(\beta, \Lambda^*, \theta^*). \quad (14)$$

$$R_j(\beta^*, \Lambda^*, \theta^*) \geq R_j(\beta^*, \Lambda^*, \theta). \quad (15)$$

Since the resource trading in a multi-agent IIoT environment is complex, finding the optimal decision policy is challenging. Therefore, we transform the optimization problem into the stochastic game problem and can be solved by the MADRL techniques.

A. Nash Equilibrium Analysis

To obtain and demonstrate the existence and uniqueness of NE in the game, we must calculate the best response strategy for each trader and derive the derivatives of equations (10) and (11) with respect to β and θ, respectively.

$$\frac{\partial R_i}{\partial \beta} = \frac{\partial}{\partial \beta}\left(\Lambda \sum_{j=1}^{D}\left(\beta_j^e \theta_j^e + \beta_j^s \theta_j^s\right) - \sum_{j=1}^{D} C\left(\theta_j^e + \theta_j^s\right)\right). \quad (16)$$

$$\frac{\partial R_j}{\partial \theta} = \frac{\partial}{\partial \theta}\left(\omega_j \log_2\left(1 + \frac{\theta_j^e}{\mu_j^e} + \frac{\theta_j^s}{\mu_j^s}\right) - \left(\beta_j^e \theta_j^e + \beta_j^s \theta_j^s\right)\right). \quad (17)$$

The utility functions of UAVOs and MDs are strictly concave in terms of β and θ, respectively, because the second-order derivatives functions in equations (16) and (17) are less than zero. The first and second-order derivative of equations (16) and (17) are presented in Appendix 1.

V. MADRL for Resource Trading in IIoT Network

DRL is a promising approach to handle complex optimization problems in the modern wireless networks for finding optimal policies that enable agents to make appropriate decisions [63]. Without prior knowledge of the system model, the agents may easily assess and pick the optimal action among alternative actions. It was initially designed for a single-agent environment; however, it has recently been used in a more complex setting where multiple agents compete or coordinate with each other in a dynamic environment. The MADRL method focuses on optimizing the reward of all trading system agents while taking other agents' behaviors into account, whereas the single-agent approach can learn a policy mapping from state to action by interacting with the underlying environment in order to maximize the cumulative reward. This paper considers a resource trading environment where multiple resource providers compete to sell their resources to MDs and maximize their profit by continually adjusting their pricing strategies. Multiple MDs, on the other hand, interact with the environment to make purchasing decisions based on the prices set by resource providers to reduce costs and improve QoS.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

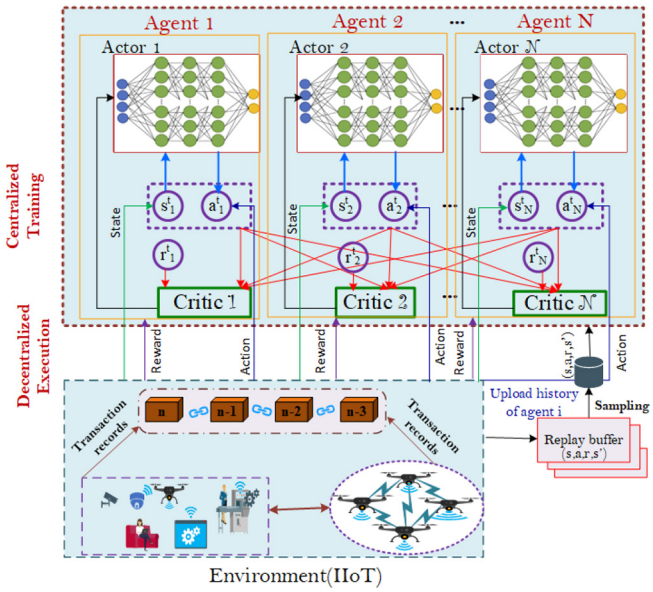


Fig. 2. MADRL Framework for resource trading in IIoT.

The optimization problems defined in equations (12) and (13) are complex and continuous action space problems that need to be formulated as a stochastic game model to handle the complexity and dynamics in the system.

A. Stackelberg Game-Based MADRL

In this subsection, the complex optimization problems formulated in the MLMF Stackelberg game are modeled as a stochastic Markov game to handle the agents' dynamic interactions with the environment that change in response to player behavior. The formulated stochastic optimization issues are then solved using dynamic programming and MADRL techniques. The MADRL is used to build a joint strategy for resource trading. The stochastic Markov game extends MDPs to the multi-agent case and repeated games to multiple state cases. In the MADRL approach, the rewards and transitions in the environment are determined by the actions of all agents in the system, as shown in Fig. 2. Therefore, the agents must learn the environment in a joint action space *A*. However, in the traditional MDP, finding the best solution is challenging because each agent has different profit-maximizing objectives. This can significantly increase the learning complexity of the agents in the resource trading system. These issues can be addressed by the formulated stochastic Markov game. The extended MDP/stochastic game can be defined as the following tuples (S, M, A, P, R, γ) to handle the multiple agents interaction in the environment, where $S = \{1, 2, 3, \ldots, S\}$ represents a finite set of states for agent *m*, $\forall m \in \mathcal{M} = \{1, 2, \ldots, M\}$ is the set of agents/players. The finite set of the joint actions denoted by *A* and A_m is the action set of agent *m*. Further, *P* defines a state transition probability, *R* describes a reward function, and γ is the discount factor $\gamma \in [0, 1]$. The extended MDP involves multiple agents that continually observe the current state s_t of the controlled system and then take action a_t among the available actions allowed in that state. Each agent acts in the environment in accordance with a specific policy $\pi(s, a)$, which represents the probabilities that govern the agent's decision to perform an action in response to the current state s_t of the environment. The agents' goal is to maximize their long-term reward *r* by iteratively changing their policy in response to the reward R_t they receive from the environment after taking action a_t. The agent will then transfer to a new state s_{t+1} and receive a reward r_t, all within a time slot *t*. Hence, these functions can be expressed as:

$$\begin{aligned} P(s, s', a) &= \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a], \\ R(s, s', a) &= \mathbb{E}[R_{t+1} | S_t = s, A_t = a], \\ \pi(s, a) &= \mathbb{P}[A_t = a | S_t = s], \end{aligned} \tag{18}$$

where S_t represents an agent's state at a learning step *t* of an episode, A_t depicts the action the agent takes at learning step *x*, and R_{t+1} denotes the reward received by the agent corresponding to the state-action pair. Therefore, the long-term reward of agent *m* is given as:

$$r_m^t = \sum_{\rho=0}^{+\infty} \gamma^\rho R_m(t + \rho + 1). \tag{19}$$

The values of γ here reflect the effect of future rewards on optimal decisions: if γ is close to 0, the decision emphasizes the short-term gain; if γ is close to 1, the decision emphasizes the long-term gain in which the future benefits are given more weight, and the decisions are said to be farsighted. Each agent *m* in the formulated stochastic game aims to maximize its expected payoff over time. Therefore, for agents in a joint strategy π, each agent *m* has a strategy π_m, and the optimization objective in equation (19) can be reformulated as follows:

$$r_m^t = E\left\{ \sum_{\rho=0}^{+\infty} \gamma^\rho r_m^{t+\rho+1} | s^t = s, \pi \right\}, \tag{20}$$

where $r_m^{t+\rho+1}$ is the immediate reward received by agent *m* at time $t + \rho + 1$, and $E\{.\}$ represents the expectation operations in which the expectation is taken over the probabilistic state transitions under strategy π from state *s*.

The MDP model of the resource trading in the IIoT system can be shown as follows. Let $\beta_t \in A_i$ and $\mu_t \in A_j$ denote the unit price set by UAVO and the resource demand action of MD, respectively, where A_i indicates the action space of the UAVO/resource provider and A_j represents the action space of MD. The state $s \in S$ is a tuple of environment features relevant to the problem at hand, and it defines the agent's relationship with its environment. At every time step *t*, the agent observes the state of its environment $s_t \in S$ and then takes action $a_t \in A$ in accordance with a policy π. The policy $\pi(s, a)$ serves as the basis for individuals to determine their probability of taking action on the current state *s*. The policy function must satisfy $\sum_{a \in A} \pi(s, a) = 1$. The environment of the agent transitions from the current state s_t to the next state s_{t+1} once the agent takes action a_t, and the agent then receives a reward r_{t+1} which reveals the benefit of doing action a_t at state s_t. This strategy forms an experience *e* at time $t + 1$, which describes an interaction of agents with the environment and is expressed as $e_{t+1} = (s_t, a_t, r_{t+1}, s_{t+1})$ [64].

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

State space: All agents in the system observe the state of the environment to gain experience and make the optimal decision. The state space is given as: $s_{m,t} = \{\beta_{m,t-1}, \mu_{m,t}, \chi_{m,t}, \psi_{m,t}\} \in \mathcal{S}$, where $\beta_{m,t-1} = \{\beta^e_{m,t-1}, \beta^s_{m,t-1}\}$ denoting prior unit pricing of energy and spectrum supplied by other agents, while $\mu_{m,t} = \{\mu^e_{m,t}, \mu^s_{m,t}\}$ denotes the energy and spectrum resource demand of MDs. Moreover, $\chi_{m,t} = \{\chi^e_{m,t}, \chi^s_{m,t}\}$ denotes the aggregated available energy and spectrum resource, and $\psi_{m,t} \in [-1, 0, 1]$ denotes the type of resource requested by the MDs. Here, $\psi_{m,t} = -1$ indicates that MD requests energy, $\psi_{m,t} = 0$ implies that MD requests spectrum, and $\psi_{m,t} = 1$ means that the MD wants both energy and spectrum.

Action space: The agent acts in accordance with a policy π, which is a mapping from the state space S to action space A, expressed as $\pi : s_t \in \mathcal{S} \rightarrow a_t \in \mathcal{A}$. The action can be from UAVOs (leaders) and MDs (followers). The action space contains a vector of transmitted energy $\theta^e = [\theta^e_1, \theta^e_2, \ldots, \theta^e_N]$, a vector of allocated spectrum $\theta^s = [\theta^s_1, \theta^s_2, \ldots, \theta^s_N]$, and a vector of decision strategy $\Lambda = [\Lambda^{e,s}_1, \Lambda^{e,s}_2, \ldots, \Lambda^{e,s}_N]$. The action space is given as $a(t) = [\Lambda^{e,s}_1, \theta^e_1, \theta^s_1, \Lambda^{e,s}_2, \theta^e_2, \theta^s_2, \ldots, \Lambda^{e,s}_N, \theta^e_N, \theta^s_N]$. The policy π can be determined using the space-action function called Q-function $Q(s_t, a_t)$, which can be approximated by deep learning [65].

Reward function: The agents consider to maximize the long term reward $r_m(t)$ by selecting optimal decision from the available actions at the time slot *t*. The reward of the followers is $r_j(t) = R_j$ (11) and the reward of UAVOs is given by $r_i(t) = R_i$ (10). The system reward is given by:

$$r(t) = \sum_{j=1}^{N} \sum_{i=1}^{U} \big(r_j(t) + r_i(t)\big). \tag{21}$$

The players in the formulated stochastic game have individual expected rewards based on the joint strategy rather than the players' individual strategies. In this non-cooperative game, players learn their best strategies through repeated interactions with the stochastic environment, and NE points are obtained where the optimal expected reward can be realized.

Definition 2: NE is a collection of strategies, one for each player in the environment, so that each individual strategy is the best response to the others, where $\pi^* = \{\pi^*_1, \ldots, \pi^*_M\}$. For each agent *m*, the strategy π^*_m is expressed as:

$$r_m(\pi^*_m, \pi_{-m}) \geq r_m\big(\pi'_m, \pi_{-m}\big), \forall \pi'_m, \tag{22}$$

where $\pi'_m \in [0, 1]$ represents all possible strategies taken by agent *m*.

B. Stackelberg-Based MADDPG

For our resource trading optimization problem, we adopt the MADDPG, which extends a DDPG-based actors-critics algorithm to handle multi-agent scenarios in a non-stationary environment and performs better in a multi-agent environment. It contains *M* agents with a set of deterministic policies for all agents. Agents can learn a policy function and a value function simultaneously using actor-critic approaches. The actor-critic has two parts: a policy model called actor and a value function named critic. The policy function acts as an actor, whereas the value function acts as a critic [66]. The output of the actor network is action value whereas the output of the critic network is Q-value. The Q-value represents an approximation of the value of the action selected by the actor network.

1) Actor Network: The actor network is a function that maps the resource trading environment state s_t to an action a_t in order to find the best price policy. Agents choose their actions based on the parameters and the trade state in each decision time slot *t*, which is expressed as:

$$a^t_m = \pi_m\big(S^t_m | \Phi^\pi_m\big). \tag{23}$$

2) Critic Network: The critic network is primarily used to evaluate the value of the actions that have been chosen by the agents. All states $(s_1, s_2, \ldots, s_M)$ and agents' actions $(a_1, a_2, \ldots, a_M)$ are inputs for critic network. The critic network calculates the temporal difference (TD)-errors, ϵ as:

$$\epsilon_m = \frac{1}{\Gamma} \sum_{\iota} \Big(y^\iota_m - Q_m\Big(s^\iota_1, \ldots, s^\iota_M, a^\iota_m | \Phi^Q_m\Big)\Big), \tag{24}$$

where $y^\iota_m = r^t_m + \gamma Q'_m(s^{\iota+1}_1, \ldots, s^{\iota+1}_M, \pi_m(s^{\iota+1}_m | \Phi^{Q'}_m))$, γ is the discount factor of the long term cumulative reward with the range of [0, 1], Γ denotes the size of the minibatch, and Φ^Q_m denotes the parameters of the critic network. The TD-errors can be taken as the evaluation results and use them to update the critic network. The actor network is updated by the policy gradient as:

$$\nabla \Phi^\pi_m J = \nabla \Phi^\pi_m \log \pi_m(s_m, a_m) \epsilon_m. \tag{25}$$

Let π is the set of all agents' deterministic policies, which is denoted as $\pi = \{\pi_1, \pi_2, \ldots, \pi_M\}$ and parameterized by $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_M\}$. The gradient of the expected reward for agent *m*, $J_{\Phi_m} = \mathbb{E}[R_m]$ expressed as:

$$\nabla_{\Phi_m} J_{\Phi_m} = \mathbb{E}_{s,a\sim\mathcal{B}}[\nabla_{\Phi_m} \log \pi_m(o_m) \nabla_{a_m} Q^\pi_m \times (\kappa, a_1, \ldots, a_M) | a_m = \pi_m(o_m)], \tag{26}$$

where $Q^\pi_m(\kappa, a_1, \ldots, a_M)$ is a centralized action-value function that takes all agents' actions, $a_1, \ldots, a_M$, and state information κ as input and outputs the Q-value for agent *m*. The observations of all agents κ is expressed as $\kappa = \{o_1, \ldots, o_M\}$. The experience replay buffer is denoted by $\mathcal{B}$, which is made up of tuples $(\kappa, \kappa', a_1, \ldots, a_M, r_1, \ldots, r_M)$, where κ' is the next state from κ after taking action $a_1, \ldots, a_M$. Furthermore, the centralized action-value function Q^π_m is updated by minimizing the loss function, which is denoted as:

$$\mathcal{X}_{\Phi_m} = \mathbb{E}_{\kappa,a,r,\kappa'}\Big[\big(Q^\pi_m(\kappa, a_1, \ldots, a_M) - y\big)^2\Big], \quad y = r_m + \gamma Q^{\pi'}_m\big(\kappa', a'_1, \ldots, a'_M\big) | a'_m = \pi'_m(o_m), \tag{27}$$

where $\pi' = \{\pi_{\Phi'_1}, \ldots, \pi_{\Phi'_M}\}$ is the set of target policies with delayed parameters Φ'_m.

1) Stackelberg-Based Actor-Critic Update: Let $F_j \subseteq M$ represent the set of followers, and $L_i \subseteq M$ represents the set of leaders for agent *m*. The actions of the leaders, $a_i = \pi_i(o_i), \forall i \in L(m)$, are computed using private observation o_i

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

and are unlikely to be affected by agent m's action because the leaders have already committed to their actions. The set of leader actions is denoted by $a_L = \{a_{i_1}, \ldots, a_{i_Z}\}$, where $Z = |L(m)| < M$. The follower responses, $a_j = \pi_j(o_j, a_m)$, $\forall j \in F(m)$, are computed using personal observations o_j and agent m's action a_m. Each follower $j \in F(m)$ may have other leaders actions a_L, where a_m in a_L are part of the input to π_j, but these actions are omitted to minimize notation and considered to be part of o_j. The set of follower actions $a_F = \{a_{j_1}, \ldots, a_{j_G}\}$, where $G \in |F(m)| < M$.

The objective of Stackelberg-based learning is to maximize the gradient of the expected reward J_{Φ_m}, $\max_{\Phi_m} J_{\Phi_m}$, where

$$\begin{aligned} J_{\Phi_m} &= \mathbb{E}_{s\sim\sigma^\pi}[R_m] \\ &= \mathbb{E}_{\kappa^0\sim\sigma}\begin{bmatrix} Q^\pi_{S,m}\big(\kappa^0, a^0_L, a^0_m, a^0_F\big)| \\ a^0_m = \pi_m\big(o^0_m\big), \\ a^0_L = \pi_i\big(o^0_i\big)\forall i \in L(m), \\ a^0_F = \pi_j\Big(o^0_j, a^0_m\Big)\forall j \in F(m) \end{bmatrix}. \end{aligned} \tag{28}$$

Here σ^π represents the discounted state observation distribution of deterministic policy π_m, which is parametrized by Φ_m and the follower responses a^0_F at time slot 0 are confined with a^0_m. The updated centralized action-value function $Q^\pi_{S,m}(\kappa, a_L, a_m, a_F)$ in (28) can be re-structured as:

$$\begin{aligned} Q^\pi_{S,m}(\kappa, a_L, a_m, a_F) &= r_m(\kappa, a_L, a_m, a_F) \\ &+ \gamma\mathbb{E}_{\kappa'}\begin{bmatrix} Q^\pi_{S,m}\big(\kappa', a'_L, a'_m, a'_F\big)| \\ a'_m = \pi_m\big(o'_m\big), \\ a'_L = \pi_i\big(o'_i\big)\forall i \in L(m), \\ a'_F = \pi_j\Big(o'_j, a'_m\Big)\forall j \in F(m) \end{bmatrix}, \end{aligned} \tag{29}$$

where the Q-function allows both leaders and followers to rely on the current global state κ and actions a to determine their conditions. The Q-value produced from this function represents the current reward r_i, as well as the discounted future return starting from state κ', and discounted by γ.

In an IIoT environment with multiple hierarchies, an agent m can have multiple leaders with actions a_L and multiple followers with actions a_F. Each action has an effect on the gradient $\nabla_{\Phi_m} J(\Phi_m)$ used to update the deterministic policy π_m of agent m. We consider two agent systems to estimate the impacts of each agent, with A_1, A_2 representing leaders and followers actions, respectively. The Stackelberg objective for the A_1 (leaders) is as follows:

$$J(\Phi_1) = \mathbb{E}_{\kappa\sim\sigma}\begin{bmatrix} Q^\pi_{S,1}(\kappa, a_1, a_2)| \\ a_1 = \pi_1(o_1), \\ a_2 = \pi_2(o_2, a_1) \end{bmatrix}. \tag{30}$$

In addition, the gradient $\nabla_{\Phi_1} J(\Phi_1)$ for A_1, considering the impacts of A_2's response is given as:

$$\nabla_{\Phi_1} J(\Phi_1) = \mathbb{E}_{\kappa,a\sim\mathcal{B}}\begin{bmatrix} \nabla_{\Phi_1}\pi_1(o_1)\nabla_{a_1} Q^\pi_{S,1}(\kappa, a_1, a_2) \\ +\nabla_{\Phi_1}\pi_1(o_1)\nabla_{a_2} Q^\pi_{S,1}(\kappa, a_1, a_2) \\ \nabla_{a_1}\pi_2(o_2, a_1)|a_1 = \pi_1(o_1), \\ a_2 = \pi_2(o_2, a_1) \end{bmatrix}, \tag{31}$$

where $\nabla_{\Phi_1}\pi_1(o_1)\nabla_{a_2} Q^\pi_{S,1}(\kappa, a_1, a_2)$ denotes the impact of follower A_2 to the gradient $\nabla_{\Phi_1} J(\Phi_1)$, which is a result of the chain rule, and $\nabla_{\Phi_1}\pi_1(o_1)$ denotes the updates to Φ_1 that modify the output policy π_1 in order to maximize the expected reward. Additionally, since the follower reaction depends on the leaders' actions, modifying the output policy π_1 impacts the follower response. Furthermore, $\nabla_{a_1}\pi_2(o_2, a_1)$ denotes the impact of changes in a_1 on the follower response, and $\nabla_{a_1} Q^\pi_{S,1}(\kappa, a_1, a_2)$ shows how changes to the follower response affect the modified action-value $Q^\pi_{S,1}$.

The Q-function updates similarly to the MADDPG update, but with the additional constraint of leaders committing to their actions before followers, and followers observe and respond to this commitment. The Q-function is updated to accommodate these constraints by minimizing the TD-error.

$$\begin{aligned} \mathcal{X}(v_1) &= \mathbb{E}_{\kappa,a,r,\kappa'\sim\mathcal{B}}\Big[\Big(Q^\pi_{S,1}(\kappa, a_1, \ldots, a_M) - y\Big)^2\Big], \\ y &= r_m + \gamma Q^{\pi'}_{S,m}\big(\kappa', a'_L, a'_m, a'_F\big) \\ a'_m &= \pi'_m(o_m), \\ a'_L &= \pi'_i(o_i), \forall i \in L(m) \\ a'_F &= \pi'_j(o_j, a'_m), \forall j \in F(m), \end{aligned} \tag{32}$$

where $\pi' = \{\pi_{\Phi'_1}, \ldots, \pi_{\Phi'_M}\}$ is the set of target policies with delayed parameters Φ'_m and $Q^{\pi'}_S = \{Q^{\pi'}_{S,1}, \ldots, Q^{\pi'}_{S,M}\}$ is the set of target critics with delayed parameters v'. Lastly, the target network of agent m can update by soft update as:

$$\begin{aligned} \Phi^{\pi'}_m &\leftarrow \Phi^\pi_m + (1-\tau)\Phi^{\pi\,\prime}_m \\ \Phi^Q_{m'} &\leftarrow \Phi^Q_m + (1-\tau)\Phi^{Q\prime}_m. \end{aligned} \tag{33}$$

Algorithm 2 provides a full description of the learning process in the SMADDPG. First, initialize the available resources of sellers and buyers, the sellers' resources (i.e., bandwidth and energy), the actor and critic network parameters with random weight Φ^π_m, and the replay memory buffer $\mathcal{B}_m$ (line 1). Each agent observes the state of the environment and performs the action, receives the reward, and creates a new state (lines 3-7). In the training phase (lines 9-15), we calculate the reward and store the experience in the replay memory buffer; we use policy training, which involves mini-batch sampling from a replay memory buffer. An actor and critic networks are then updated based on a randomly selected sample.

2) Computational Complexity Analysis: We investigate the computational complexity of our proposed SMADDRL algorithm using Big *O* notation. Its computational complexity is of order *O*(*M.T.W*), where *M* is the total number of agents, *T* is the number of episodes and *W* is the learning steps. Therefore, increasing the heterogeneity and number of agents has no significant effect on each agent's computational complexity.

VI. Performance Evaluation

This section presents the performance evaluation of our proposed DRL-based resource trading scheme (SMADDPG) with the benchmark algorithms (DDPG [67], DQN [34], and MADDPG [68]) in terms of various metrics such as transaction processing delay, profit, cost, and convergence rate.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2 SMADDPG Algorithm for Resource Trading

Initialize: The actor network $\pi_m(s_m|\Phi_m^\pi)$ with weights Φ_m^π; the critic network $Q_m(s_1,\ldots,s_M,a_m|\Phi_m^Q)$ with weights Φ_m^Q;
Initialize: Actor and critic target networks with weights $\Phi^{\pi'} \leftarrow \Phi^\pi$ and $\Phi^{Q'} \leftarrow \Phi^Q$
Initialize: The replay buffer $\mathcal{B}_m$
for each *episode* $= 1$ to V **do**
  Configure simulation environment
  Receive initial state S_0
  for $t = 1$ to T **do**
    Each agent observes the trading state s_m^t
    Execute **Algorithm 1**
    For agent m, selects action $a_m = \pi_m(s_m|\Phi_m^\pi)$
    Execute a_m^t
    Observe next state s_m^{t+1}
    for each agent $m = 1$ to M **do**
      Calculate the reward r_m^t for m
      Store $(s_m^t, a_m^t, r_m^t, s_m^{t+1})$ in $\mathcal{B}_m$
      Sample a random mini-batch of Γ transactions $(s_m^t, a_m^t, r_m^t, s_m^{t+1})$ in $\mathcal{B}_m$
      Set $y_m^t = g_m^t + \gamma Q'_m(s^t,\ldots,s_m^t,a_m^t|\Phi_m^Q)$
      Update the critic network Q_m by minimizing the loss as in (27)
      Each agent updates the actor network as in (25)
    end for
    Update target network parameters of each agent m by (33)
  end for
end for

TABLE II
SIMULATION PARAMETERS

Parameters	Values
Number of MBS	1
Number of BSs	2
ABS coverage radius	800m
Distance between ABSs	125m
Number of clusters	2
Number of channels	10
Number of MDs	70
Number of UAVOs	10
Max. transmission power of UAVOs	125dBm
Computation capacity of UAVOs	15GHz/sec
Max. transmission power of MDs	10dBm
Noise power	-90dBm
Learning rate of actor, critic networks	$1e^{-4}, 3e^{-4}$
System bandwidth	20 Mhz
Number of RBs	100
Discount factor	0.95
Soft update factor	$1e^{-3}$

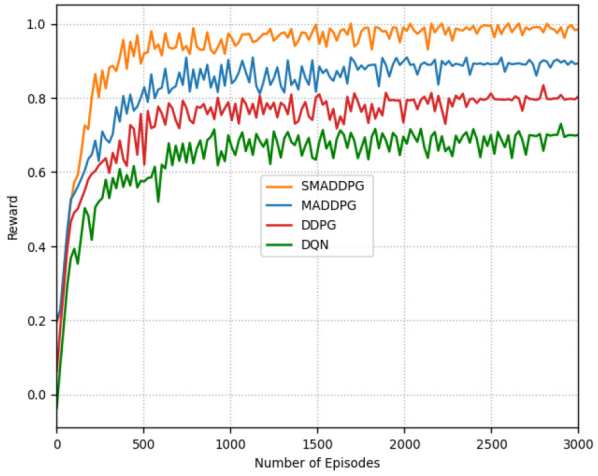


Fig. 3. System reward ($\delta = 0.5$).

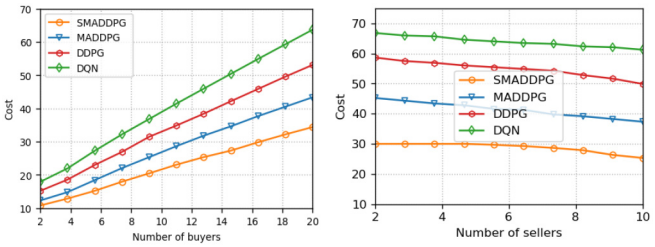
A. Simulation Environment

In this paper, we set up a multi-UAV-assisted IIoT network simulation environment based on the configuration and parameters utilized in [68], [69]. We consider a group of multi-UAV connected in D2D link called UAV cluster deployed in a small cell area with radius $r_u = 800\,\mathrm{m}$. The MDs are randomly and uniformly distributed in small cells, where UAVs fly at a fixed altitude of 100m. The sub-channel bandwidth is set to 80KHz, and the QoS threshold ($SINR_{min}$) is 3.5dB. The simulations are run in a Python 3.6 environment on a machine with a Core i7 processor running at 2.4GHz and 16GB memory. In our simulation, we set the maximum number of training episodes as 3000 and the maximum episode step as 25 for both our algorithm and baseline algorithms. The path-loss exponent is set to -90 dBm and the sub-channel bandwidth is 80 kHz. In this proposed multi-agent strategy, we use a fully connected neural network (NN) with critic-network and actor-network. In both the actor and critic NNs, we deploy two hidden layers for each agent, with the first hidden layer set to 256 and the second hidden layer set to 128. We use Rectified Linear Unit (ReLU) as an activation function for the hidden layers, and the sigmoid function is employed at the output layer. We used a replay buffer size of 10^7 and a mini-batch size of 256. We set the probability of agent action selection $\delta \in (0,1)$. We use Adam optimizer function, which measures the degree to which newly acquired information overrides old information, resulting in a faster learning rate. The learning rate for an actor is set to 0.0001 and for a critic is 0.001. The discount factor is set as 0.95, which determines the importance of future rewards.

B. Convergence Analysis

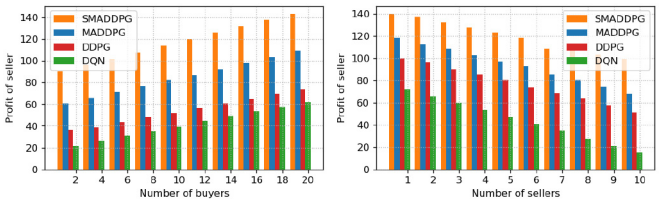
We first evaluate the convergence of our algorithm with respect to various contexts. Fig. 3 compares the convergence on the average system rewards of SMADDPG to baseline schemes in terms of the agent's action selection probability and training step. The performance of all algorithms is initially unsatisfactory due to random action selection during the exploration phase, but after gaining some experience by interacting with the wireless environment, SMADDPG has a higher learning performance than the other algorithms. The agents in the SMADDPG algorithm learn and update their policies more quickly, early converge than the baseline algorithms and after about 400 episodes, they begin to converge. The proposed SMADDPG and MADDPG algorithms have more cooperative and distributive sharing policies and experiences than DDPG and DQN algorithms. Generally, this figure shows that when

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.



(a) Impact of increasing buyers (b) Impact of increasing sellers

Fig. 4. The effect of increasing buyers and sellers with on cost of buyers.



(a) Impact of increasing buyers (b) Impact of increasing UAVOs

Fig. 5. The effect of increasing buyers and sellers number on the profit of UAVOs.

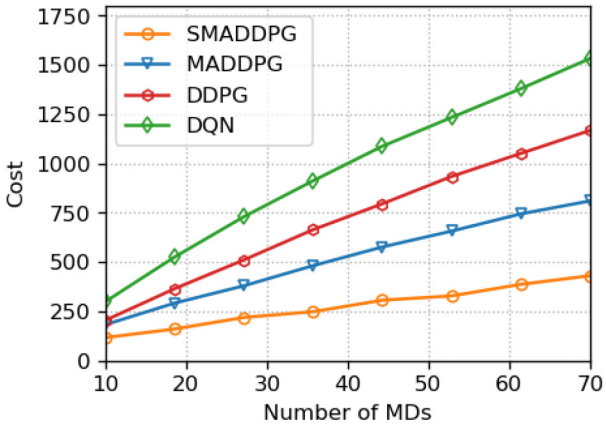


Fig. 6. System cost with respect to number of MDs.

the reward value is high, the leaders and followers achieve higher utilities with optimal prices and minimum costs.

C. Performance Analysis

We compare the performance of the proposed algorithm with the benchmark algorithms in terms of buyer cost as the number of buyers increases. As shown in Fig. 4(a), the costs of buyers increase as the number of buyers increase in the system. The increasing number of buyers indicates the number of MDs that require more resources to achieve their QoS and perform given tasks within the time frame. Due to this, the sellers/UAVOs also change the resource prices dynamically depending on the number of buyers. Then, the overall costs from the resource seller's perspective can increase as the number of buyers increases parallelly. The proposed algorithm increases the cost more slowly than benchmark algorithms. The results show that SMADDPG scored 16.38%, 17.81%, and 19.31% lower than MADDPG, DDPG, and DQN, respectively. Therefore, the proposed SMADDPG algorithm achieved better performance than benchmark algorithms, i.e., it has optimal prices and minimum computation costs. Furthermore, we assessed the cost of buyers in relation to the increase in the number of resource providers. The simulation results in Fig. 4(b) show that the cost of buyers decreases slightly as the number of resource providers increases because resource providers compete with each other to sell more resources by setting an optimal price. When the number of resource providers/sellers in the system increases, the proposed SMADDPG algorithm reduces buyers' costs by 21.66%, 25.15%, and 28.13% compared to MADDPG, DDPG, and DQN, respectively.

We also compare the utility of resource providers as the number of sellers and buyers increases. According to the results shown in Fig. 5(a), the utility/revenue of resource providers increases as the number of buyers increases. The utility of resource provider increase as the number of buyers increase in SMADDPG by 28.62%, 46.36%, and 92.41% compared to MADDPG, DDPG, and DQN, respectively. Therefore, the proposed SMADDPG algorithm obtained better profits when the number of resource buyers increased than other benchmark algorithms.

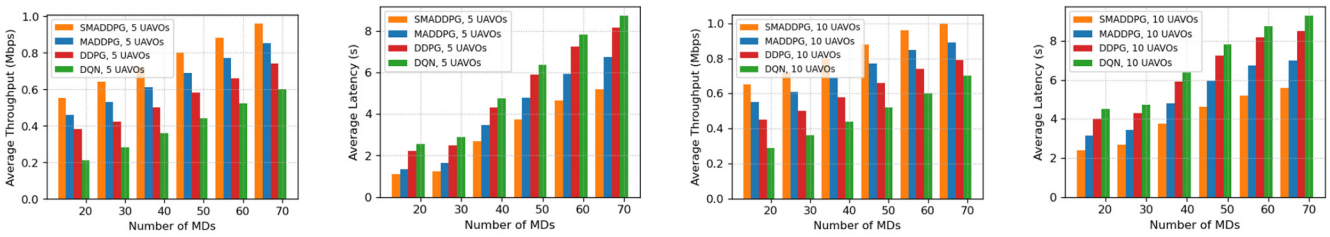
On the other hand, Fig. 5(b) shows that the profit of the sellers decrease when the number of sellers increases. It shows that the sellers compete with each other to provide resources at optimal prices. The sellers' prices decline slowly when the number of sellers increases. Due to this, sellers' profit decreases as the number of sellers increases. Hence, the proposed SMADDPG algorithm outperforms MADDPG, DDPG, and DQN in terms of profit, with 19.32%, 45.35%, and 147.36%, respectively.

In our simulation, the overall system cost is analyzed by considering the number of MDs requesting energy and spectrum resources, consensus nodes, and resource sellers in the proposed scenario. As shown in Fig. 6, the overall cost of the system rises as the number of resources requesting MDs increases, and other blockchain entities increases. Compared with MADDPG, DDPG, and DQN, the SMADDPG reduces system cost by 53.09%, 67.34%, and 75.71%, respectively.

D. Network Performance Analysis

To show the coupling of blockchain and the IIoT network, we investigate network performance in terms of average blockchain transaction throughput and latency with varying UAVOs (sellers) and MDs (connected devices, such as buyers and other devices interacting each other via UAV networks). As illustrated in Fig. 7(a), the average throughput of the network is evaluated with 5 UAVOs serving as resource sellers and increasing the number of MDs connected to the systems from 20-70. The figure shows that the average throughput increases with the increasing number of MDs connected to the network. This is because when the number of the MD increase their service demand also increases. Compared to other schemes, SMADDPG has the highest throughput, with 15.13%, 32.43%, and 61.5% for MADDPG, DDPG, and DQN, respectively. The average transaction processing latency also

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.



(a) Throughput (UAVO=5). (b) Latency (UAVO=5). (c) Throughput (UAVO=10). (d) Latency (VAVO=10).

Fig. 7. Blockchain performance analysis in terms of UAVOs and MDs.

TABLE III
PERFORMANCE COMPARISON OF THE PROPOSED SCHEME WITH STATE-OF-THE-ART APPROACHES

Performance Metrics	**DQN**	**DDPG**	**MADDPG**	**Our proposed SMADDPG**
Latency (sec)	6.46	5.96	4.81	3.74
Throughput (Mbps)	0.46	0.6	0.71	0.82
Average system cost	1185.9	882.2	614.2	288.10
Utility optimization	43.84	67.36	91.15	117.12

increases as the number of MDs in the network increases. This is because when the number of transaction request increases, the consensus time required also increase. However, according to Fig. 7(b), the average transaction processing latency of SMADDPG is 24.85%, 47.89%, and 56.01% lower than that of MADDPG, DDPG, and DQN, respectively. This is because our proposed algorithm allows both UAVOs and MDs to reach at optimal strategy in shorter time than other approaches. Moreover, we analyze the throughput and latency of the blockchain to measure the performance of the proposed scheme with increased number of UAVOs to evaluate the impact of the increasing number of sellers in the blockchain performance. As shown in Fig. 7(c), the throughput slightly increases with the increasing number of the sellers (UAVOs) and MDs (buyers). This is because the increase of both sellers and buyers can allow demand and supply matching. However, SMADDPG achieves a better performance than the benchmark schemes and makes it easier for buyers and sellers to reach optimal policies. SMADDPG allows buyers and sellers to reach NE faster, significantly increasing the system's throughput. From the simulation results, we observed that SMADDPG achieves a throughput of 13.87%, 29.55%, and 53.03%, higher than MADDPG, DDPG, and DQN, respectively. Similarly, the latency increases with an increasing number of UAVOs and MDs, as shown in Fig. 7(d). But the SMADDPG scheme reduced the latency by 22.5%, 44.38%, and 52.27% over the MADDPG, DDPG, and DQN schemes, respectively.

We compared the performance of our proposed scheme with other state-of-the-art schemes in Table III. We assess the performance of the proposed scheme based on the increasing number of UAVs, MDs, and consensus nodes in the blockchain-enabled IIoT network. The SMADDPG has a lower latency as it uses the actors-critics algorithm to handle multi-agent settings and dynamic environments where the agents' actions vary over time. With an increasing number of MDs and agents in the system, transactions/resource requests also increase. Further, the designed dynamic pricing algorithm promotes efficient trading among agents, increasing overall system performance, including throughput, latency, etc. This reduces system costs and improves the utility of traders (both UAVs and MDs).

VII. CONCLUSION

In this paper, we proposed a novel resource trading framework that integrates MADRL with blockchain and game theory to achieve secure and efficient resource sharing between various types of MDs in the UAV-assisted IIoT networks. A consortium blockchain with SC is deployed to ensure the security and privacy of the resource trading system. Furthermore, the optimization problems are modeled using the MLMF Stackelberg game. We converted the formulated optimization problems into a stochastic game and solved them using the proposed SMADDPG algorithm to deal with the complexity and dynamics of the IIoT network. According to the simulation results, our proposed scheme outperforms others in terms of improving the efficiency of resource trading in UAV-assisted IIoT networks. In the future, we will investigate slicing and virtualization for intelligent and efficient resource trading in UAV-assisted IIoT networks.

APPENDIX
THE NE ANALYSIS USING FIRST AND SECOND-ORDER DERIVATIVES

To analyzes the NE, we use a Hessian of R_i and R_j as follows:

$$R_i = \left(\Lambda \sum_{j=1}^{D}\left(\beta_j^e \theta_j^e + \beta_j^s \theta_j^s\right) - \sum_{j=1}^{D} C\left(\theta_j^e + \theta_j^s\right)\right)$$

$$\text{So, } \frac{\partial R_i}{\partial \beta} = \frac{\partial}{\partial \beta}\left(\Lambda \sum_{j=1}^{D}\left(\beta_j^e \theta_j^e + \beta_j^s \theta_j^s\right) - \sum_{j=1}^{D} C\left(\theta_j^e + \theta_j^s\right)\right)$$

$$= \left[\frac{\partial R_i}{\partial \beta_1}, \frac{\partial R_i}{\partial \beta_2}, \ldots, \frac{\partial R_i}{\partial \beta_D}\right]$$

$$\text{Let } \Delta_j = \frac{\partial R_i}{\partial \beta}, \text{where}, j = 1, 2, \ldots, D$$

$$\Rightarrow \Delta_j = \Lambda\left[e\beta_j^{e-1}\theta_j^e + s\beta_j^{s-1}\theta_j^s\right]$$

$$\Rightarrow \frac{\partial R_i}{\partial \beta} = [\Delta_1, \Delta_2, \ldots, \Delta_D]$$

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

Then, R_i is calculated as follows using Hessian matrix

$$R_i = \begin{bmatrix} \frac{\partial^2 R_i}{\partial \beta_1^2} & \cdots & \frac{\partial^2 R_i}{\partial \beta_1 \partial \beta_d} \\ \cdots & \cdots & \cdots \\ \frac{\partial^2 R_i}{\partial \beta_d \beta_1} & \cdots & \frac{\partial^2 R_i}{\partial \beta_d^2} \end{bmatrix}$$

$$\frac{\partial R_i}{\partial \theta} = \begin{bmatrix} \frac{\partial \Delta_1}{\partial \beta_1} & \cdots & \frac{\partial \Delta_d}{\partial \beta_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial \Delta_1}{\partial \beta_d} & \cdots & \frac{\partial \Delta_d}{\partial \beta_d} \end{bmatrix}$$

If $r_j = \frac{\partial^2 R_i}{\partial \beta^2} = \Lambda(e(e-1)\theta_j^e \beta_j^{e-2} + s(s-1)\theta_j^s \beta_j^{s-2}$, where $j = 1, 2, \ldots, D$.

$$H(R_i) = \begin{bmatrix} r_1 & 0 & \ldots & 0 \\ 0 & r_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & r_D \end{bmatrix}$$

Therefore, $r_j < 0$, then $H(R_i)$ is strictly concave.

$$R_j = \left(\omega_j \log_2\left(1 + \frac{\theta_j^e}{\mu_j^e} + \frac{\theta_j^s}{\mu_j^s}\right) - \left(\beta_j^e \theta_j^e + \beta_j^s \theta_j^s\right)\right)$$

$$\frac{\partial R_j}{\partial \theta} = \left[\frac{\partial R_j}{\partial \theta_1}, \ldots, \frac{\partial R_j}{\partial \theta_D}\right]$$

$$\text{Let } \alpha_j = \frac{\partial R_j}{\partial \theta_j}, j = 1, \ldots, D$$

$$\alpha_j = \frac{\partial R_j}{\partial \theta_j} = \omega_j \frac{\partial}{\partial \theta_j}\left(log_2\left(1 + \frac{\theta_j^e}{\mu_j^e} + \frac{\theta_j^s}{\mu_j^s}\right) - C\left(B_j^e \theta_j^e + \beta_j^s \theta_j^s\right)\right)$$

$$= \omega_j \frac{\partial}{\partial \theta_j}\left(\ln\left(1 + \frac{\theta_j^e}{\mu_j^e} + \frac{\theta_j^s}{\mu_j^s}\right) - C\left(\beta_j {}^e 0_j {}^e + \theta_j^s \theta_j^s\right)\right)$$

$$= \frac{\omega_j}{\ln_2}\left(\frac{\frac{e\theta_j^{e-1}}{\mu_j^e} + \frac{s\theta_j^{s-1}}{\mu_j^s}}{\left(1 + \frac{\theta_j^e}{\mu_j^e} + \frac{\theta_j^s}{\mu_j^s}\right)}\right) - C\left(e\beta_j^e \theta_j^{e-1} + s\beta_j^s \theta_j^{s-1}\right)$$

Then,

$$\frac{\partial R_j}{\partial \theta} = \begin{bmatrix} \alpha_1 & 0 & \ldots & 0 \\ 0 & \alpha_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \alpha_D \end{bmatrix}$$

Let $\zeta_j = \frac{\partial^2 R_j}{\partial \theta^2} = \frac{\partial \alpha_j}{\partial \theta_j} = \frac{\omega_j}{\ln_j}\left(\frac{a*b-c*d}{b^2}\right) - q$ where

$$a = e(e-1)\frac{\theta_e^{e-2}}{\mu_j^e} + s(s-1)\frac{\theta_s^{s-2}}{\mu_j^s}$$

$$b = 1 + \frac{\theta_j^e}{\mu_j^e} + \frac{\theta_j^s}{\mu_j^s}$$

$$c = d = \frac{e\theta_j^{e-1}}{\mu_j^e} + \frac{s\theta_j^s}{\mu_j^s}$$

$$q = C\left(e(e-1)\beta_j^e \theta_j^e + s(s-1)\beta_j^s \theta_j^{s-2}\right)$$

$$H(R_j) = \begin{bmatrix} \zeta_1 & 0 & \ldots & 0 \\ 0 & \zeta_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \zeta_D \end{bmatrix} \quad (34)$$

Then $\zeta_j < 0$, $H(R_j)$ also strictly concave function.

REFERENCES

[1] D. O'Halloran and E. Kvochko, "Industrial Internet of Things: Unleashing the potential of connected products and services, collaboration with Accenture," Cologny, Switzerland, World Econ. Forum, White Paper, 2015, p. 34. [Online]. Available: http://reports.weforum.org/industrial-internet-of-things

[2] Z. Shi, X. Xie, H. Lu, H. Yang, M. Kadoch, and M. Cheriet, "Deep-reinforcement-learning-based spectrum resource management for Industrial Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3476–3489, Mar. 2021.

[3] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[4] S. Iqbal, R. M. Noor, A. W. Malik, and A. U. Rahman, "Blockchain-enabled adaptive learning-based resource sharing framework for IIoT environment," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14746–14755, Oct. 2021.

[5] S. Messaoud, A. Bradai, O. B. Ahmed, P. T. A. Quang, M. Atri, and M. S. Hossain, "Deep federated Q-learning-based network slicing for industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5572–5582, Aug. 2021.

[6] Y. Miao, Q. Tong, K.-K. R. Choo, X. Liu, R. H. Deng, and H. Li, "Secure online/offline data sharing framework for cloud-assisted Industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8681–8691, Oct. 2019.

[7] C. Paniagua and J. Delsing, "Industrial frameworks for Internet of Things: A survey," *IEEE Syst. J.*, vol. 15, no. 1, pp. 1149–1159, Mar. 2021.

[8] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of Industrial Internet of Things security: Requirements and fog computing opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2489–2520, 4th Quart., 2020.

[9] W. Sun, J. Liu, Y. Yue, and Y. Jiang, "Social-aware incentive mechanisms for D2D resource sharing in IIoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5517–5526, Aug. 2020.

[10] W. Mao, Z. Zhao, Z. Chang, G. Min, and W. Gao, "Energy efficient Industrial Internet of Things: Overview and open issues," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7225–7237, Nov. 2021.

[11] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined Industrial Internet of Things: A dueling deep Q -learning approach," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4627–4639, Jun. 2019.

[12] J. Wan *et al.*, "Toward dynamic resources management for IoT-based manufacturing," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 52–59, Feb. 2018.

[13] L. Yang, M. Li, Y. Zhang, P. Si, Z. Wang, and R. Yang, "Resource management for energy-efficient and blockchain-enabled industrial IoT: A DRL approach," in *Proc. IEEE 6th Int. Conf. Comput. Commun. (ICCC)*, 2020, pp. 910–915.

[14] B. Yang, X. Cao, X. Li, Q. Zhang, and L. Qian, "Mobile-edge-computing-based hierarchical machine learning tasks distribution for IIoT," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2169–2180, Mar. 2020.

[15] Z. Xiong, Y. Zhang, N. C. Luong, D. Niyato, P. Wang, and N. Guizani, "The best of both worlds: A general architecture for data management in blockchain-enabled Internet-of-Things," *IEEE Netw.*, vol. 34, no. 1, pp. 166–173, Jan./Feb. 2020.

[16] M. B. Mollah *et al.*, "Blockchain for the Internet of Vehicles towards intelligent transportation systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, Mar. 2021.

[17] A. H. Khan *et al.*, "Blockchain and 6G: The future of secure and ubiquitous communication," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 194–201, Feb. 2022.

[18] L. Tang and H. Hu, "Computation offloading and resource allocation for the Internet of Things in energy-constrained MEC-enabled HetNets," *IEEE Access*, vol. 8, pp. 47509–47521, 2020.

[19] Z. Jia, Q. Wu, C. Dong, C. Yuen, and Z. Han, "Hierarchical aerial computing for Internet of Things via cooperation of HAPs and UAVs," *IEEE Internet Things J.*, early access, Feb. 16, 2022, doi: 10.1109/JIOT.2022.3151639

[20] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[21] W. Feng, J. Wang, Y. Chen, X. Wang, N. Ge, and J. Lu, "UAV-aided MIMO communications for 5G Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1731–1740, Apr. 2019.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

[22] H. Ke, H. Wang, W. Sun, and H. Sun, "Adaptive computation offloading policy for multi-access edge computing in heterogeneous wireless networks," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 1, pp. 289–305, Mar. 2022.
[23] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.
[24] Z. Zhao *et al.*, "Predictive UAV base station deployment and service offloading with distributed edge learning," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 3955–3972, Dec. 2021.
[25] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 3rd Quart., 2019.
[26] T. Yuan, C. E. Rothenberg, K. Obraczka, C. Barakat, and T. Turletti, "Harnessing UAVs for fair 5G bandwidth allocation in vehicular communication via deep reinforcement learning," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4063–4074, Dec. 2021.
[27] S. H. Alsamhi *et al.*, "Green Internet of Things using UAVs in B5G networks: A review of applications and strategies," *Ad Hoc Netw.*, vol. 117, Jun. 2021, Art. no. 102505.
[28] W. Zhang *et al.*, "Deep reinforcement learning based resource management for DNN inference in IIoT," in *Proc. IEEE Global Commun. Conf.*, 2020, pp. 1–6.
[29] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4925–4934, Jul. 2021.
[30] H. Zhou, C. She, Y. Deng, M. Dohler, and A. Nallanathan, "Machine learning for massive Industrial Internet of Things," 2021, *arXiv:2103.08308*.
[31] A. Mohammed, H. Nahom, A. Tewodros, Y. Habtamu, and G. Hayelom, "Deep reinforcement learning for computation offloading and resource allocation in blockchain-based multi-UAV-enabled mobile edge computing," in *Proc. 17th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, 2020, pp. 295–299.
[32] P. Yu *et al.*, "Intelligent-driven green resource allocation for Industrial Internet of Things in 5G heterogeneous networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 520–530, Jan. 2022.
[33] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in Industrial Internet of Things: Architecture, advances and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2462–2488, 4th Quart., 2020.
[34] A. M. Seid, G. O. Boateng, S. Anokye, T. Kwantwi, G. Sun, and G. Liu, "Collaborative computation offloading and resource allocation in multi-UAV assisted IoT networks: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12203–12218, Aug. 2021.
[35] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3602–3609, Jun. 2019.
[36] Y. Xu, Z. Liu, C. Huang, and C. Yuen, "Robust resource allocation algorithm for energy-harvesting-based D2D communication underlaying UAV-assisted networks," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 17161–17171, Dec. 2021.
[37] X. Lin, J. Wu, S. Mumtaz, S. Garg, J. Li, and M. Guizani, "Blockchain-based on-demand computing resource trading in IoV-assisted smart city," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1373–1385, Jul.–Sep. 2021.
[38] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.
[39] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12313–12325, Dec. 2018.
[40] N. Mhaisen, M. S. Allahham, A. Mohamed, A. Erbad, and M. Guizani, "On designing smart agents for service provisioning in blockchain-powered systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 2, pp. 401–415, Mar./Apr. 2022.
[41] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.
[42] S. Sheng, R. Chen, P. Chen, X. Wang, and L. Wu, "Futures-based resource trading and fair pricing in real-time IoT networks," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 125–128, Jan. 2020.
[43] C. Ma *et al.*, "Cooperative spectrum sharing in D2D-enabled cellular networks," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4394–4408, Oct. 2016.
[44] M. K. Farshbafan, M. H. Bahonar, and F. Khaiehraveni, "Spectrum trading for device-to-device communication in cellular networks using incomplete information bandwidth-auction game," in *Proc. 27th Iran. Conf. Elect. Eng. (ICEE)*, 2019, pp. 1441–1447.
[45] J. Qiu, D. Grace, G. Ding, J. Yao, and Q. Wu, "Blockchain-based secure spectrum trading for unmanned-aerial-vehicle-assisted cellular networks: An operator's perspective," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 451–466, Jan. 2020.
[46] L. Xue, W. Yang, W. Chen, and L. Huang, "STBC: A novel blockchain-based spectrum trading solution," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 1, pp. 13–30, Mar. 2022.
[47] Z. Liu, D. Wang, J. Wang, X. Wang, and H. Li, "A blockchain-enabled secure power trading mechanism for smart grid employing wireless networks," *IEEE Access*, vol. 8, pp. 177745–177756, 2020.
[48] X. Lin, J. Wu, A. K. Bashir, J. Li, W. Yang, and J. Piran, "Blockchain-based incentive energy-knowledge trading in IoT: Joint power transfer and AI design," *IEEE Internet Things J.*, early access, Sep. 15, 2020, doi: 10.1109/JIOT.2020.3024246.
[49] M. J. A. Baig, M. T. Iqbal, M. Jamil, and J. Khan, "IoT and blockchain based peer to peer energy trading pilot platform," in *Proc. 11th IEEE Annu. Inf. Technol. Electron. Mobile Commun. Conf. (IEMCON)*, 2020, pp. 402–406.
[50] K. Zhang *et al.*, "Incentive-driven energy trading in the smart grid," *IEEE Access*, vol. 4, pp. 1243–1257, 2016.
[51] N.-T. Nguyen *et al.*, "Energy trading and time scheduling for energy-efficient heterogeneous low-power IoT networks," in *Proc. IEEE Global Commun. Conf.*, 2020, pp. 1–6.
[52] D. Zhang, F. R. Yu, and R. Yang, "Blockchain-based distributed software-defined vehicular networks: A dueling deep *Q*-learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1086–1100, Dec. 2019.
[53] K. R. B. Sri, P. Aneesh, K. Bhanu, and M. Natarajan, "Design analysis of solar-powered unmanned aerial vehicle," *J. Aerosp. Technol. Manage.*, vol. 8, no. 4, hboxpp. 397–407, 2016.
[54] L. D. Nguyen, K. K. Nguyen, A. Kortun, and T. Q. Duong, "Real-time deployment and resource allocation for distributed UAV systems in disaster relief," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2019, pp. 1–5.
[55] B. Wang, Y. Sun, Z. Sun, L. D. Nguyen, and T. Q. Duong, "UAV-assisted emergency communications in social IoT: A dynamic hypergraph coloring approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7663–7677, Aug. 2020.
[56] H. Hydher, D. N. K. Jayakody, K. T. Hemachandra, and T. Samarasinghe, "Intelligent UAV deployment for a disaster-resilient wireless network," *Sensors*, vol. 20, no. 21, p. 6140, 2020.
[57] M. Y. Selim and A. E. Kamal, "Post-disaster 4G/5G network rehabilitation using drones: Solving battery and backhaul issues," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.
[58] N. Malik and B. Joshi, "ECDSA approach for reliable data sharing and document verification using two level QR code," in *Proc. 2nd Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)*, 2018, pp. 434–437.
[59] Z. Su, Y. Wang, Q. Xu, M. Fei, Y. Tian, and N. Zhang, "A secure charging scheme for electric vehicles with smart communities in energy blockchain," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4601–4613, Jul. 2019.
[60] H. N. Abishu, A. M. Seid, Y. H. Yacob, T. Ayall, G. Sun, and G. Liu, "Consensus mechanism for blockchain-enabled vehicle-to-vehicle energy trading in the Internet of Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 946–960, Jan. 2022.
[61] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4585–4600, Jun. 2019.
[62] M. Ferrara, M. Khademi, M. Salimi, and S. Sharifi, "A dynamic Stackelberg game of supply chain for a corporate social responsibility," *Discr. Dyn. Nat. Soc.*, vol. 2017, Feb. 2017, Art. no. 8656174.
[63] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L.-C. Wang, "Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 44–52, Jun. 2019.
[64] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.

[65] Z. Li and C. Guo, "Multi-agent deep reinforcement learning based spectrum allocation for D2D underlay communications," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1828–1840, Feb. 2020.
[66] T. Yuan, W. D. R. Neto, C. E. Rothenberg, K. Obraczka, C. Barakat, and T. Turletti, "Dynamic controller assignment in software defined Internet of Vehicles through multi-agent deep reinforcement learning," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 585–596, Mar. 2021.
[67] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[68] A. M. Seid, G. O. Boateng, B. Mareri, G. Sun, and W. Jiang, "Multi-agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4531–4547, Dec. 2021.
[69] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

Abegaz Mohammed Seid (Member, IEEE) received the B.Sc. degree in computer science from Ambo University in 2010, the M.Sc. degree in computer science from Addis Ababa University, Ethiopia, in 2015, and the Ph.D. degree in computer science and technology from the University of Electronic Science and Technology of China in 2021. He is currently a Postdoctoral Fellow with the College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar. He served as a Graduate Assistant and a Lecturer, as well as a member of the academic committee, and an Associate Registrar with Dilla University, Ethiopia, from 2010 to 2016. He has published more than seven scientific conferences and journal papers. His research interests include a wireless network, mobile edge computing, blockchain, machine learning, vehicular network, IoT, machine learning, UAV network, IoT, and 5G/6G wireless network.

Hayla Nahom Abishu received the B.Sc. degree in computer science and information technology from Haramaya University in 2007, and the M.Sc. degree in computer science and networking from Dilla University in 2017, Ethiopia. He is currently pursuing the Ph.D. degree in computer science and technology with the University of Electronic Science and Technology of China, where he is also a member with the Mobile Cloud-Network Research Team. His research interests include mobile computing, wireless network, blockchain, UAV network, IoT, network security, and machine learning.

Yasin Habtamu Yacob received the B.Sc. degree in information technology from Addis Ababa University, Addis Ababa, Ethiopia, in 2005, and the M.Sc. degree in computer science and networking from Dilla University, Dilla, Ethiopia, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, University of Electronic Science and Technology of China. He was a Senior Instructor of Cisco Networking Academy for more than five years. His current research interests include blockchain, mobile edge computing, wireless networks, IoT, machine learning and network security. He won the Cisco Advanced Level Instructors Award of 2015 and 2016.

Tewodros Alemu Ayall received the B.Sc. degree in computer science from University of Gondar, Ethiopia, in 2010, the M.Sc. degree in computer science from Andhra University, India, in 2015, and the Ph.D. degree in computer science and technology from the University of Electronic Science and Technology of China, China, in 2021. He is engaged in research of distributed graph processing, distributed graph database, big data processing, big graph partitioning, and blockchain research.

Aiman Erbad (Senior Member, IEEE) received the B.Sc. degree in computer engineering from the University of Washington, Seattle, in 2004, the Master of Computer Science degree in embedded systems and robotics from the University of Essex, U.K. in 2005, and the Ph.D. degree in computer science from the University of British Columbia, Canada, in 2012. He is an Associate Professor and ICT Division Head with the College of Science and Engineering, Hamad Bin Khalifa University. Prior to this, he was an Associate Professor with the Computer Science and Engineering Department and the Director of Research Planning and Development, Qatar University until May 2020. His research received funding from the Qatar National Research Fund, and his research outcomes were published in respected international conferences and journals. His research interests span cloud computing, edge intelligence, Internet of Things (IoT), private and secure networks, and multimedia systems. 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 2020 Best Research Paper Award from Computer Communications, the IWCMC 2019 Best Paper Award, and the IEEE CCWC 2017 Best Paper Award. He also served as the Director of Research Support responsible for all grants and contracts from 2016 to 2018 and as the Computer Engineering Program Coordinator from 2014 to 2016. He is an Editor for *KSII Transactions on Internet and Information Systems* and the *International Journal of Sensor Networks* and a Guest Editor for IEEE NETWORK. He also served as the Program Chair of the International Wireless Communications Mobile Computing Conference (IWCMC 2019), as the Publicity chair of the ACM MoVid Workshop 2015, as the Local Arrangement Chair of NOSSDAV 2011, and as the Technical Program Committee Member in various IEEE and ACM international conferences (GlobeCom, NOSSDAV, MMSys, ACMMM, IC2E, and ICNC). He is a senior member of ACM.

Mohsen Guizani (Fellow, IEEE) received the B.S. degree (with distinction) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor as appointed as an Associate Provost for Faculty Affairs and Institutional Advancement with Mohamed Bin Zayed University of Artificial Intelligence, United Arab Emirates. Previously, he served in different academic and administrative positions with the University of Idaho, Western Michigan University, University of West Florida, University of Missouri-Kansas City, University of Colorado-Boulder, and Syracuse University. He is the author of nine books and more than 600 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. Throughout his career, he received three teaching awards and four research awards. He is the recipient of the 2017 IEEE Communications Society Wireless Technical Committee Recognition Award, the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks, and the 2019 IEEE Communications and Information Security Technical Recognition Award for outstanding contributions to the technological advancement of security. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He also served as a member, the Chair, and the General Chair of a number of international conferences. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He is currently the Editor-in-Chief of the *IEEE Network Magazine*, serves on the editorial boards of several international technical journals and the Founder and the Editor-in-Chief of *Wireless Communications and Mobile Computing* journal (Wiley). He guest edited a number of special issues in IEEE journals and magazines. He is Senior Member of ACM.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:17:10 UTC from IEEE Xplore. Restrictions apply.