Hierarchical Reinforcement Learning for Blockchain-Assisted Software Defined Industrial Energy Market

Yifan Cao, *Student Member, IEEE*, Xiaoxu Ren, *Student Member, IEEE*, Chao Qiu, *Member, IEEE*, and Xiaofei Wang, *Senior Member, IEEE*

***Abstract*—Energy Internet (EI) is developing and booming rapidly with the increase of distributed energy resources, which is beneficial to address the severe condition of industrial energy. However, there are inevitable credit crises and utility optimization challenges in EI that need to be settled. In this article, we propose a blockchain-assisted software defined energy Internet (BSDEI), where a distributed energy market smart contract is designed to ensure transactions executed reliably and participants' accounts dealt accurately. In order to jointly optimize the utilities of operators, retailers, and industrial prosumers in BSDEI, we formulate the whole trading process as a three-stage Stackelberg game, with the proof of existence and uniqueness for the Stackelberg equilibrium. Then, we design a hierarchical distributed policy gradient algorithm to solve the Stackelberg game under incomplete information. We implement a blockchain-based industrial energy trading system using a middleware platform. The smart contract is deployed on the consortium blockchain, providing website interfaces for participants to operate. Furthermore, we conduct experiments for analyzing economic benefits. Our system prototype demonstrates the feasibility of BSDEI and the algorithm exceeds about 18% in total mean reward than comparing algorithms.**

***Index Terms*—Blockchain, industrial energy market, reinforcement learning, software defined network (SDN), Stackelberg game.**

I. Introduction

INDUSTRIAL energy is generally supplied by few utility companies, with high transmission loss in long distance and expensive unit energy price, which hinders the development of industrial energy. With the tendency of distributed energy resource (DER), energy Internet (EI) has rapidly gained the spotlight [1], which is beneficial to address the dilemma of industrial energy. However, widespread DER and ossified control methods overturn the huge benefit of EI. Meanwhile, software defined networking (SDN) offers programmable control for ubiquitous networks, where the control and data are decoupled. This control paradigm flexibly separates distributed energy devices from energy applications [2]. Therefore, the marriage of SDN and EI gives the birth to software defined energy Internet (SDEI), which brings a hope of sorting out these issues, including reliability and flexibility [3].

Due to the reasonable price and effective transmission, the industrial energy market gradually takes shape in SDEI. It transforms ordinary energy consumers to energy retailers, who are capable to generate, store, and sell DER [4]. The transformation potentially brings less transmission loss and lower load peaks to EI. Despite its potential benefits, the industrial energy market currently faces major issues, which hinder its widespread adoption. 1) The credit crisis among different trading entities prevents it from the way of trading energy reliably and credibly. 2) Due to the fact that the utility is monopolistic and difficult to balance among trading entities, the industrial energy market currently is less attractive.

There are two solutions that can be used to address the above challenges. One is blockchain, which works as a distributed ledger to record transactions and provides trustworthy services to a group of nodes without central authority [5]. Specifically, the energy trading information among interest entities is reliably managed and stored on the blockchain. As enforced computer protocols in blockchain [6], smart contracts have the capability to assist trading entities in the industrial energy market to effectively reduce the cost of credit.

The other one is game theory to solve the problem of utility optimization in SDEI. However, to search the equilibrium in the game, some current methods, such as backward induction and some heuristic algorithms, acquiescently suppose that a centralized broker [7] is capable of collecting parameter information from all entities and helping them adopt policies. Unfortunately, parameter information of entities is strictly protected. To prevent privacy from disclosure, we assume all the entities in the industrial energy market as a multiagent system [8]. Compared with traditional heuristics approaches, multiagent reinforcement learning (MARL) converges to a great equilibrium due to the

Manuscript received July 31, 2021; revised October 15, 2021 and November 18, 2021; accepted December 5, 2021. Date of publication January 6, 2022; date of current version June 13, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2101901, in part by the National Science Foundation of China under Grant 62072332, in part by China NSFC (Youth) under Grant 62002260, and in part by the China Postdoctoral Science Foundation under Grant 2020M670654. Paper no. TII-21-3284. *(Corresponding author: Chao Qiu.)*

The authors are with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China (e-mail: yifancao@tju.edu.cn; xiaoxuren@tju.edu.cn; chao.qiu@tju.edu.cn; xiaofeiwang@tju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3140878.

Digital Object Identifier 10.1109/TII.2022.3140878

balance of exploration and learning of all agents, without requiring ideal knowledge about the environment and complete information of other agents [9].

In this article, we design a blockchain-assisted software defined energy Internet (BSDEI), where the consortium blockchain acts as a trusted and decentralized ledger. To reduce the consensus cost and the risk of malicious node attacks [10], we adopt practical byzantine fault tolerance (PBFT) as the consensus mechanism. Moreover, to overcome the credit crisis in large-scale distributed energy trading in SDEI, a trading smart contract for the industrial energy market is designed to assist trading entities to automatically execute their reached agreement and provides a number of open interfaces for participants to trace their transaction records. To solve the utility optimization problems without leaking the privacy of participants in SDEI, we design a hierarchical distributed industrial policy gradient algorithm, under an incomplete information environment to search the game equilibrium. The main contributions are as follows.

1) To address the dilemma of industrial energy, we propose a blockchain-assisted software defined EI architecture. A smart contract is designed to build a self-organizing industrial energy market.
2) We model the energy trading and transmission in BSDEI as a three-stage hierarchical Stackelberg game, giving the proof of existence and uniqueness for the Stackelberg equilibrium (SE).
3) Considering the privacy issue during the game, we design a hierarchical distributed policy gradient (HDPG) algorithm under incomplete information of different entities to solve the game equilibrium.
4) We factually construct a blockchain-based industrial energy trading system, by designing the distributed energy market smart contract to reach the distributed automation control of energy trading and dispatching.

The rest of this article is organized as follows. Section II presents some related works. The architecture introduction and problem formulation are shown in Section III. The three-stage game model with proof is shown in this section. In Section IV, the hierarchical MARL algorithm HDPG is designed for the proposed game model. Furthermore, a smart contract is conceived for the industrial energy market. Section V presents the system deployment and the simulation results. Finally, Section VI concludes this article.

II. Related Work

A. SDN and Blockchain Technologies in Energy Field

The concept of SDEI is primarily proposed by the authors in [11]. They focus on an energy control system based on SDN to achieve programmable control of energy flow from a high-level perspective. To address the single point of failure with one SDN controller, the authors in [12] build a distribution platform based on distributed SDN controllers, which provides energy services in SDEI and resists the network threats encountered through the communication in SDEI. In the face of consensus problem among different controllers in SDN, the authors in [3] merge blockchain as a trusted third party into SDN control plane, which collects and synchronizes traceable control information among different SDN controllers securely.

A blockchain-based energy trading system is proposed in [13] to solve the untrustworthy problem in the traditional trading market. Besides, the authors propose the concept of credit banks based on credit mechanism to speed up energy trading. The paper [14] introduces edge computing in energy trading based on blockchain realizing a frequent and convenient loan scheme, solving the "cold start" and "long return" in large-scale efficient problems trading. However, these works only consider the energy trading between buyer and seller and attach less importance to the balance of multiparty utilities.

B. Game Theory to Model Trading Process

On account of its distributed autonomous solution for optimization problems, game theory has been widely used in microgrid energy trading researches. A lot of researches search for the game equilibrium by transforming a two-stage game to a single optimization model with Karush–Kuhn–Tucher conditions [15]. Backward induction [16], as an optimization method, is also frequently used in Stackelberg game. The authors in [7] propose a hierarchical energy market architecture. The power generation company, load aggregator, and microgrid constitute a three-stage dynamic game under complete information to solve the problem of utility coordination in the energy market. In [17], the authors further consider the competition among multiple sellers and the competition between buyers for the seller selection. They adopt a Stackelberg game and evolutionary game to model the relationships, respectively.

C. Deep Reinforcement Learning in Energy Field

The authors in [18] propose a deep Q network (DQN) based energy trading method to solve the mismatch problem between supply and demand based on game theoretic in the energy trading process. In [19], the authors formulate the matching problem between uncertain wind power and electric vehicle charging demand as a bi-level Markov decision process (MDP) model and use a proximal policy optimization (PPO) algorithm to solve it. An energy scheduling policy based on a soft actor-critic (SAC) algorithm is proposed in [20] to minimize operational costs and ensure power supply reliability. However, all these above works neglect the privacy protection of game participants.

III. System Architecture and Game Formulation

A. Different Types of Entities in BSDEI

Our research focuses on the energy transaction interaction between various interest entities in BSDEI. In an energy trading process, there are three types of entities.

1) Prosumer: We define prosumers as industrial energy users who cannot generate energy by themselves or their energy generation is unable to supply their energy consumption. They purchase energy from the utility company or the retailer in BSDEI. Prosumers have different usage scenarios for energy

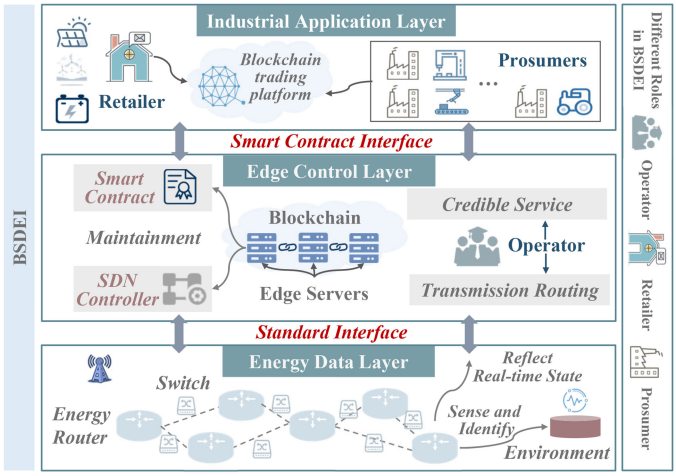


Fig. 1. Architecture of the blockchain-assisted software defined energy Internet.

utilization, such as assembly line production, lathe process, industrial agriculture [21], etc.

2) Retailer: A retailer is defined as the one whose total energy generation is greater than total energy consumption with distributed energy generation devices and energy storage devices. The retailers get payments by providing DER to various industrial applications. Inversely, they have to afford the cost of distributed generation and energy storage and pay the operator for transmission routing service.

3) Operator: The operator becomes the intermediary between industrial production and distributed generation, assisting to accomplish the energy trading process. To provide more convenient service and lower latency, the operator deploys hardware devices, such as edge servers and SDN controllers, at the edge side instead of the cloud side. In return, it charges retailers for transmission routing services and prosumers for credible blockchain services.

B. Layers of Content and Interfaces

BSDEI is divided into industrial application layer (IAL), edge control layer (ECL), and energy data layer (EDL) as shown in Fig. 1. These three planes are mutually independent and interrelated, which decouples the energy routing and dispatching control in BSDEI. They jointly serve for energy trading in the industrial energy market. EDL interacts with ECL through the standard interface OpenFlow, while IAL interacts with ECL through the smart contract interface.

1) Industrial Application Layer: More direct transaction process and low unit energy price than that of the utility company incentivize retailers and prosumers to participate in the industrial energy market. They conduct energy transactions through the client of the blockchain trading platform.

2) Edge Control Layer: Due to the intermittent and uncertainty of DER, large-scale access of DER will bring great challenges to the stability of EI. Therefore, a more intelligent energy dispatching control paradigm is needed. The physical devices in ECL are composed of distributed SDN controllers and edge servers maintained by the operator. These controllers jointly dispatch and control the energy routing of the physical layer. Each of the edge servers acts as a node in the consortium blockchain, taking on the functions of bookkeeping, broadcasting, verification, and consensus. The consortium blockchain provides credible and stable third-party services for energy transactions in IAL. The smart contract in the blockchain provides reliable automated process control for energy trading.

3) Energy Data Layer: The EDL consists of numerous communication devices and energy devices. Communication devices include switches, which receive dispatching instructions issued by ECL, and forward them to the corresponding energy routers along the specified optimal path. With some sensing elements, energy routers automatically sense and reflect the real-time state of the environment to ECL, which facilitates the controllers to modify the dispatching instructions. In addition, energy routers also receive upper level commands from EDL to change their states.

C. System Model

1) Prosumers' Model: In a local industrial energy market, we assume that there are a set of prosumers, presented as $\mathcal{N} = \{1, 2, \ldots, j, \ldots, N\}$. The prosumer j submits its energy demand q_j with unit energy price p to the local retailer. In order to maximize its overall benefit, it will dynamically adjust its energy consumption and demand. Considering the marginal benefit of energy utilization, we model the benefit function as a quadratic function. It should be noticed that prosumers have different industrial intentions to utilize energy [22], defined by $\mathcal{W} = \{1, 2, \ldots, w_j, \ldots, w_N\}$, which are tightly related to their utilities. Therefore, the prosumer j determines its energy demand to maximize the utility $\mathcal{U}_j^p(q_j)$

$$\max_{q_{\min} \le q_j \le q_{\max}} \delta \left(w_j q_j - \frac{1}{2} q_j^2 \right) - mpq_j - r \tag{1}$$

where δ and m are conversion factors. The first term indicates that the prosumer j obtains benefit in industrial production, using its purchased energy. The second term and the third term represent the energy payment pq_j to the retailer and blockchain service payment r to the operator.

2) Retailer's Model: The retailer's utility is determined by the unit energy price p and the total quantity of demand $\sum_{j=1}^{N} q_j$ from prosumers. In turn, the retailer needs to afford the generation cost C_g and storage cost C_s, which are respectively modeled as

$$C_g = a \left(\frac{\sum_{j=1}^{N} q_j}{1-\phi} \right)^2 + b \frac{\sum_{j=1}^{N} q_j}{1-\phi} + k \tag{2}$$

$$C_s = \frac{\sum_{j=1}^{N} q_j}{1-\phi} \frac{c_s}{\xi_c \xi_d} \tag{3}$$

where a, b, and k denote the weighted factors of generating cost [7]. Considering the energy loss during transmission, the actual energy that the retailer needs to generate and store is denoted as $\frac{\sum_{j=1}^{N} q_j}{1-\phi}$, where ϕ denotes the transmission loss rate. c_s is the unit storage cost of energy, while ξ_c and ξ_d are the charging and

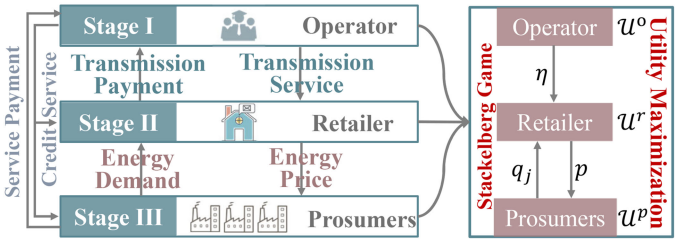


Fig. 2. Three-stage Stackelberg game in BSDEI.

discharging efficiencies of energy storage devices [22]. Besides, the retailer pays for the dispatching and transmission service, with a commission rate η. The retailer intends to maximize its utility $\mathcal{U}^r(p)$ by adjusting the unit energy price

$$\max_{p_{\min}\le p\le p_{\max}} p\sum_{j=1}^{N} q_j - \eta\sum_{j=1}^{N} q_j - C_g - C_s. \tag{4}$$

3) Operator's Model: The operator charges the retailer for dispatching and transmission service. Besides, an incentive mechanism is designed for operating and maintaining the blockchain. The operator acquires an extra fee $\frac{U^m}{s}$ for providing a reliable blockchain service for each transaction:

$$U^m = (R_f + rs)\lambda. \tag{5}$$

On the one hand, R_f is the fixed block allowance, where s is a parameter related to the block size. On the other hand, the operator gets a transaction fee r from prosumers in every transaction. λ is a probability factor in the blockchain. Similar to the storage cost, the transmission cost $C_t = \frac{\sum_{j=1}^{N} q_j}{1-\phi} c_t$ includes the exact prosumers demand portion and the lost energy portion, with unit transmission cost c_t. C_o is considered as fixed cost for operation and maintenance. Hence, the operator intends to maximize its utility $\mathcal{U}^o(\eta)$ by adjusting the unit service price

$$\max_{\eta_{\min}\le\eta\le\eta_{\max}} \eta\sum_{j=1}^{N} q_j + U_m - \frac{\sum_{j=1}^{N} q_j}{1-\phi} c_t - C_o. \tag{6}$$

D. Game Equilibrium Analysis

The three-stage Stackelberg game is shown in Fig. 2. We will give the existence and uniqueness of the SE next.

Theorem 1: The Subgame $\mathcal{G}_p = \{\mathcal{N}, \{\mathcal{U}_j^p\}_{j\in\mathcal{N}}, [0,+\infty)^N\}$ exists a unique equilibrium, which is given by

$$q_j^* = \delta - \frac{m}{w_j}\cdot p. \tag{7}$$

Proof: According to (1), we obtain the first and second derivatives of utility $\mathcal{U}_j^p$ with respect to q_j

$$\frac{\partial \mathcal{U}_j^p}{\partial q_j} = \delta\cdot(w_j - q_j) - m\cdot p \tag{8}$$

and

$$\frac{\partial^2 \mathcal{U}_j^p}{\partial^2 q_j} = -w_j. \tag{9}$$

Since prosumers tend to use energy services, w_j is assumed non-negative, and the second derivative is negative. We can arrive at the conclusion that the utility function is concave. Next, we prove the existence of equilibrium. Let $\mathbf{q} \triangleq (q_1,\ldots,q_N)$, from the $\mathcal{U}^p(\mathbf{q}) \triangleq \mathcal{U}_1^p(\mathbf{q}),\ldots,\mathcal{U}_N^p(\mathbf{q})$, we have point-to-set mapping $\mathbf{F} = \mathbf{F}(\mathcal{U}^p) = [\nabla_{q_j}\mathcal{U}_j^p(\mathbf{q})]_{j=1}^N$, where $\nabla\mathbf{F} = \nabla\mathbf{F}(\mathcal{U}^p(\mathbf{q}))$

$$\begin{aligned}
&= \begin{bmatrix} \nabla_{11}^2\mathcal{U}_1^p(\mathbf{q}) & \nabla_{12}^2\mathcal{U}_1^p(\mathbf{q}) & \ldots & \nabla_{1\text{-}N}^2\mathcal{U}_1^p(\mathbf{q}) \\ \nabla_{21}^2\mathcal{U}_2^p(\mathbf{q}) & \nabla_{22}^2\mathcal{U}_2^p(\mathbf{q}) & \ldots & \nabla_{2\text{-}N}^2\mathcal{U}_2^p(\mathbf{q}) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla_{N1}^2\mathcal{U}_N^p(\mathbf{q}) & \nabla_{N2}^2\mathcal{U}_N^p(\mathbf{q}) & \ldots & \nabla_{NN}^2\mathcal{U}_N^p(\mathbf{q}) \end{bmatrix} \\
&= \begin{bmatrix} -w_1 & 0 & \ldots & 0 \\ 0 & -w_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & -w_N \end{bmatrix}.
\end{aligned} \tag{10}$$

Therefore, $\nabla\mathbf{F} + \nabla\mathbf{F}^{\mathbf{T}}$ is negative definite. Furthermore, $\nabla\mathbf{F}$ is diagonally strictly concave [23]. Then the Subgame $\mathcal{G}_p$ exists a unique equilibrium. We set $\frac{\partial \mathcal{U}_j^p}{\partial q_j} = 0$ and get q_j^*. ■

Theorem 2: When the unit service price η is given, the optimal unit energy price p^* is

$$p^* = \frac{A}{B} \tag{11}$$

where $A = N\cdot\delta + (\eta + \frac{b}{1-\phi} + \frac{C_s}{\xi_c\cdot\xi_d\cdot(1-\phi)} + \frac{2a\cdot\delta}{(1-\phi)^2})\cdot\sum_{j=1}^{N}\frac{m}{w_j}$, $B = 2\sum_{j=1}^{N}\frac{m}{w_j} + \frac{2a}{(1-\phi)^2}\cdot\sum_{j=1}^{N}\frac{m^2}{w_j^2}$.

Proof: From the first-order condition, we obtain the first derivative of utility $\mathcal{U}^r$ with respect to unit energy price p

$$\begin{aligned}
\frac{\partial \mathcal{U}^r(p)}{\partial p} = &\sum_{j=1}^{N} q_j^* + p\cdot\sum_{j=1}^{N}\frac{\partial q_j^*}{\partial p} - \eta\cdot\sum_{j=1}^{N}\frac{\partial q_j^*}{\partial p} - \\
&\frac{2a\cdot\sum_{j=1}^{N} q_j^*\cdot\frac{\partial q_j^*}{\partial p}}{(1-\phi)^2} - \frac{b}{1-\phi}\cdot\sum_{j=1}^{N}\frac{\partial q_j^*}{\partial p} - \\
&\frac{C_s}{\xi_c\xi_d(1-\phi)}\cdot\sum_{j=1}^{N}\frac{\partial q_j^*}{\partial p}.
\end{aligned} \tag{12}$$

Then we set $\frac{\partial \mathcal{U}^r(p)}{\partial p} = 0$. According to (7), we obtain the $\frac{\partial q_j^*}{\partial p} = -\frac{m}{w_j}$, and we substitute it to (12) to obtain the expression of p^*. ■

Theorem 3: With the best response of the prosumers q_j^* and retailer p^*, the optimal unit service price of operator η^* is expressed by

$$\eta^* = \frac{N\cdot\delta\cdot B - (C - \frac{c_t}{1-\phi})\cdot(\sum_{j=1}^{N}\frac{m}{w_j})^2}{2\cdot(\sum_{j=1}^{N}\frac{m}{w_j})^2} \tag{13}$$

where $C = N\cdot\delta + (\frac{b}{1-\phi} + \frac{C_s}{\xi_c\cdot\xi_d\cdot(1-\phi)} + \frac{2a\cdot\delta}{(1-\phi)^2})\cdot\sum_{j=1}^{N}\frac{m}{w_j}$.

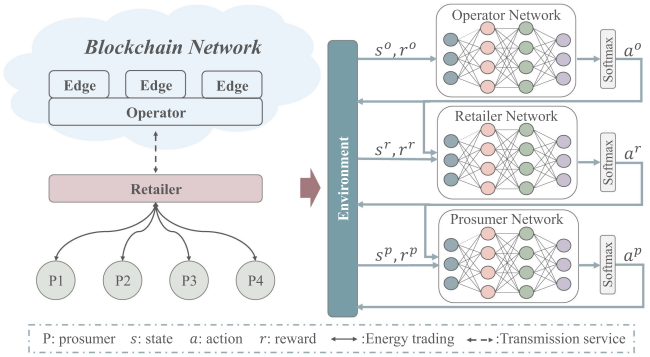


Fig. 3. Architecture of hierarchical multiagent reinforcement learning algorithm in energy trading process.

Proof: From the first-order condition, we obtain the first derivative of utility $\mathcal{U}^o$ with respect to unit service price η

$$\frac{\partial \mathcal{U}^o(\eta)}{\partial \eta} = \sum_{j=1}^{N} q_j^* - \frac{c_t}{1-\phi} \sum_{j=1}^{N} \frac{\partial q_j^*}{\partial p} \cdot \frac{\partial p^*}{\partial \eta} + \eta \cdot \sum_{j=1}^{N} \frac{\partial q_j^*}{\partial p} \cdot \frac{\partial p^*}{\partial \eta}. \tag{14}$$

Then we set $\frac{\partial \mathcal{U}^o(\eta)}{\partial \eta} = 0$. According to (11), we obtain the $\frac{\partial p^*}{\partial \eta} = \sum_{j=1}^{N} \frac{m}{w_j}$, and we substitute it to (14) to obtain the expression of η^*. ■

Theorem 4: The proposed sequential three-stage game has the unique SE.

Proof: In the proposed three-stage sequential game, each stage has its optimal closed-form solution respectively: the unit service price η^* in (13), the unit energy price p^* in (11), and the energy demand $\{q_j^*\}$ in (7). As each stage has the perfect and unique equilibrium, the SE for the proposed three-stage game model exists and is unique as well.

IV. Hierarchical MARL Algorithm and Distributed Energy Market Smart Contract

A. Implementation of Hierarchical MARL

To solve the three-stage Stackelberg game, we model the whole procedure of industrial energy trading in BSDEI as a Markov decision process. Considering the Stackelberg game under incomplete information, we design a hierarchical MARL algorithm shown in Fig. 3 to solve the problem, instead of leaking the privacy parameters of participants in other heuristic algorithms. Hierarchical MARL focuses on models including multiple agents that learn policies dynamically as they interact with the environment. Compared with single-agent reinforcement learning, the proposed algorithm avoids serious oscillation problems that come from huge state space and action space of an agent, which is more suitable for the dynamic energy market.

We design the action spaces $\mathcal{A}_o$, $\mathcal{A}_r$, and $\mathcal{A}_p$ of the operator, retailer, and prosumers, respectively. At time slot t, the operator first sets the unit transmission price $\eta^t \in \mathcal{A}_o$ based on its observation $s_o^t = \{p^{t-1}, [q_j^{t-1}]_{i \in N}\}$ from the lower level games at time slot $(t-1)$. Its reward function is denoted as (6). Then, we define the state of the retailer as $s_r^t = \{\eta^t, [q_j^{t-1}]_{i \in N}\}$. Its action

Algorithm 1: HDPG for the Three-Stage Stackelberg Game.

Input: α_o, α_r, α_p, γ, E, T, M, $\mathcal{A}^o, \mathcal{A}^r, \mathcal{A}^p, \mathcal{S}^o, \mathcal{S}^r, \mathcal{S}^p$.
Initialization:
Initialize $\theta_o(s_o^t, a_o^t)$, $\theta_r(s_r^t, a_r^t)$ and $\theta_p(s_p^t, a_p^t)$ arbitrarily.
for $episode = 1, 2, \ldots, E$ **do**
Reset actions of all agents.
for $t = 1, 2, \ldots, T$ **do**
1. Operator step:
Observe the state $s_o^t : p^{t-1}, q_1^{t-1}, \ldots, q_N^{t-1}$
$\theta_o \leftarrow \theta_o + \alpha_o \nabla_{\theta_o} \log \pi_o(s_o^t, \eta^t; \theta_o) R_o^t$.
2. Retailer step:
Observe the state $s_o^t : \eta^t, q_1^{t-1}, \ldots, q_N^{t-1}$.
$\theta_r \leftarrow \theta_r + \alpha_r \nabla_{\theta_r} \log \pi_r(s_r^t, p^t; \theta_r) R_r^t$.
3. Prosumers step:
for $j = 1, 2, \ldots, N$ **do**
Observe the state $s_{p_j}^t : p^t$.
$\theta_{p_j} \leftarrow \theta_{p_j} + \alpha_{p_j} \nabla_{\theta_{p_j}} \log \pi_p(s_{p_j}^t, q_j^t; \theta_p) R_{p_j}^t$.
end for
end for
end for
Output: π_o, π_r, π_p

is signified by $p^t \in \mathcal{A}_r$, while the reward function is expressed as the utility function in (4). After observing the actions of the operator and retailer at time slot t, each of the prosumers determines its submitted purchase action $q_j^t \in \mathcal{A}_p$ based on the observed states $s_{p_j}^t = p^t$. The reward function of prosumer j is expressed as the utility function in (1).

A MARL algorithm called HDPG is designed for searching the SE in incomplete information environments constructed by multiagent, as shown in Algorithm 1. Let α_o, α_r, and $\alpha_p \in (0, 1]$ represent the learning rate of policy network about operators, retailers, and prosumers, respectively. $\gamma_e \in (0, 1]$ is the future reward discount factor. E denotes the total episode, T denotes maximum time-step in each episode, while M means the batch size of training networks. Each policy network consists of an input layer, an output layer, and two fully connected layers.

First, the algorithm randomly initializes the policy networks $\pi_o(s_o^t, \eta^t; \theta_o)$, $\pi_r(s_r^t, p^t; \theta_r)$ and $\pi_p(s_{p_j}^t, q_j^t; \theta_p)$, with their weights θ_o, θ_r, θ_p. As shown in Fig. 3, agents' actions in each round consist of three steps for the operator, retailer, and prosumers, respectively. The operator, as the high-level leader of the game, preferentially updates the weights θ_o in operator policy network with s_o^t and selects an action η^t. The retailer, in the middle-level of the game, observes the operator's action in this step and prosumers' action in the last step. Then, it updates the retailer policy network and takes its action p^t. In the prosumers step, prosumers observe the high-level action and middle-level action from the above-mentioned steps. After updating weights θ_p in the prosumer policy network, each of them chooses its action q^t individually based on the state and the intention of using energy.

For each episode of the outer, middle, and inner loops, the time complexity is $O(E)$, $O(T)$, and $O(N)$, respectively. According

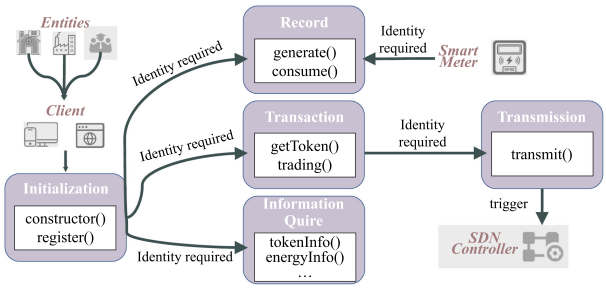


Fig. 4. Trading process in the smart contract.

to [24], the time complexity of a fully connected layer is denoted as $T_f = O(\sum_{l=1}^{L} K_l K_{l-1})$, where L represents the layers of neural networks, and K_l means the number of neural units in a fully connected layer. For calculating θ_o, θ_r, and θ_p, each policy network utilizes two fully connected layers to output an action. Hence, the total time complexity of Algorithm 1 is regarded as $O(ETN(T_f))$.

B. Implementation of Smart Contract for Energy Market

For credit crisis among trading entities in industrial energy market, a smart contract is designed to realize the automatic execution of trading and clearing of accounts. The smart contract consists of five modules as shown in Fig. 4, i.e., Registration, Transaction, Transmission, Record, and Information Quire. To realize the unified management, we design a circulation currency called $etcoin$.

First, the contract deployer is registered as the initial administrator of the trading system, and some public parameters are initialized by $constructor()$. Then, participants register accounts with their usernames, account addresses, and account types by $register()$ function, which initializes these accounts' information, including their available energy and etcoins as well as total generated and used energy. Participants could acquire etcoins by $getToken()$ function. When transactions begin, prosumers select the retailer and confirm the quantity of energy. Before $trading()$ function finishes the update of the corresponding accounts information, some requirements will verify the accounts' rights and ensure the balance adequate. The $transimit()$ function is called automatically following the $trading()$. It not only clears the transactions between retailers and operators but also triggers the SDN controllers, which ensures the energy flow from retailers to prosumers. According to information from smart meters in retailers' and prosumers' location, $generate()$ and $consume()$ functions are responsible for recording the total generated and used energy of participants. Furthermore, we design several quire functions to conveniently acquire accounts' information, such as $tokenInfo()$, $energyInfo()$, etc.

Algorithm 2: Distributed Energy Market Smart Contract.

$constructor() \rightarrow$ Initialize:
  $administrator, totalTokens, tokenPrice$
  $unitEnergyPrice, unitTransPrice.$
function $register(address, name, type)$
  Require $(administrator)$
  $accountName, accountAddr, accountType.$
  $availEnergy = 0, availTokens = 0.$
  $totalGenEnergy = 0, totalUseEnergy = 0.$
function $getToken(address, value)$
  Require
 $(sender == address, value <= totalTokens)$
  Etcoins balance clearing
function
 $trading(seller, operator, value, ePrice, tPrice)$
  Require$(value <= seller.availEnergy)$
  Require$(value * ePrice <= sender.availTokens)$
  Trade energy and etcoins balance clearing
function
 $transmit(from, to, by, value, ePrice, tPrice)$
  Require $(value <= from.availEnergy)$
  Require $(value * ePrice <= to.availTokens)$
  $sdnController.start()$
  Transmit energy and etcoins balance clearing
function $generate(), consume()$
  Require $(sender == administrator)$
  Smart meters update energy record
function $tokenInfo(), energyInfo()$
  Inquire available etcoins, consumed, generated, and available energy for utilization and selling.

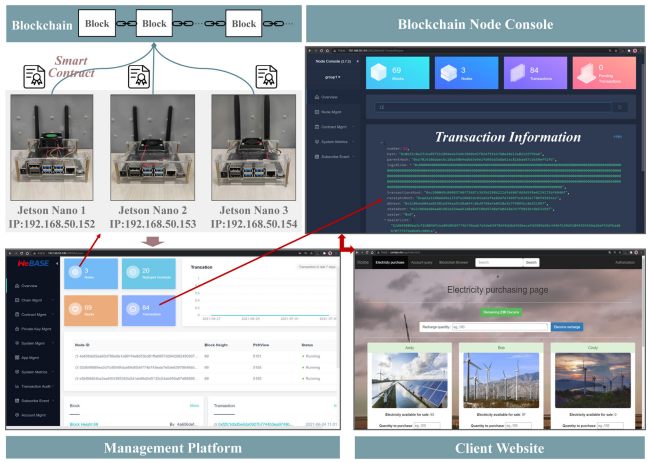


Fig. 5. Blockchain-based industrial energy trading system.

V. Simulation Results and System Deployment

A. System Implementation

To better complete our work, we implement a consortium blockchain-based industrial energy trading system, which is demonstrated in Fig. 5.

1) Consortium Blockchain Configuration: To factually provide blockchain services for energy trading, we build a consortium blockchain across physical hardware based on FISCO BCOS,[1] which is a stable and efficient blockchain underlying

[1][Online]. Available: https://github.com/FISCO-BCOS/FISCO-BCOS

platform. As shown in the top left of Fig. 5, the devices' configuration in implementation is three Jetson Nanos with 4-GB RAM, Quad-core ARM A57, a laptop with 16-GB RAM, Intel i5-10210 U as the management device, and a router for network access service. Each Jetson Nano represents an edge server in BSDEI and acts as a consensus node providing credible service in the consortium blockchain.

2) Platform Implementation and Application Deployment: In order to provide better transaction services for participants and to more conveniently manage the system, a common set of components are built between the industrial energy market application and consortium blockchain nodes based on WeBASE,[2] which is a comprehensive middleware platform assisting to develop blockchain-based distributed application. It provides unified managements of chains, contracts, private keys, and applications. To provide distributed blockchain service, we open node front function for each Jetson Nano, as shown in the top right of Fig. 5. Using the built management platform shown in the bottom left of Fig. 5, we compile and deploy the smart contract on the constructed blockchain. Additionally, after testing the smart contract, strict authentication of identities and calling modes are set to prevent some malicious attacks caused by permission problems. In order to facilitate prosumers to conveniently operate their accounts and conduct transactions, a client website is designed for prosumers to call the function as shown in the bottom right of Fig. 5. Besides, prosumers could interact with the consortium blockchain for information, such as the current block and transaction details, through the client.

B. Parameter Setting

During the simulation, we set $A_o \in [2, 12]$, $A_o \in [15, 30]$, and $A_x \in [0, 70]$. For parameters in HDPG algorithm, the learning rates are set as $\alpha_o = \alpha_r = \alpha_p = 0.0004$ and the future discount factor $\gamma = 0.9$. The total episode E is set as 700, the maximum time-step T is set as 700, and the batchsize M is set as 128. For parameters in the models of industrial energy market [22], we set $\lambda = 0.8, \phi = 0.85, c_t = 1, c_s = 2, \xi_c = 0.85, \xi_d = 0.9$, and $\delta = 70$. For parameters in blockchain [13], we consider $R_f = 5, r = 2$, and $s = 5$.

C. Simulation Results

In this section, we provide convergence analysis and economic analysis to illustrate the equilibrium performance. We mainly focus on the convergence performance, the economic benefit comparison, and the effect of diverse experiment parameters on the utilities of different entities in BSDEI.

1) Convergence Analysis: First, as shown in Fig. 6, we verify the convergence performance of HDPG algorithm under the unified pricing mechanism. To clearly present the tendency, we set one operator, one retailer, and ten prosumers in this experiment.

Before game equilibrium analysis, we plot the model of each party in BSDEI under different variables shown in Fig. 7. For each case, we annotate the theoretical optimal solution

[2][Online]. Available: https://github.com/WeBankFinTech/WeBASE

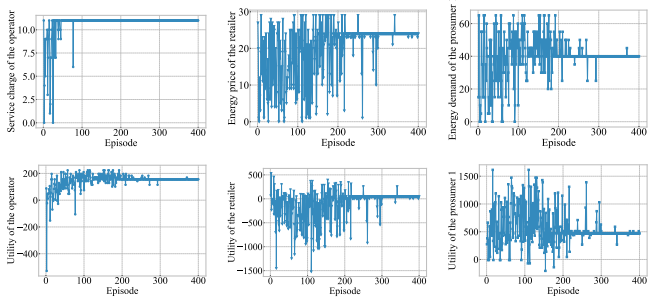


Fig. 6. Convergence of the game under HDPG.

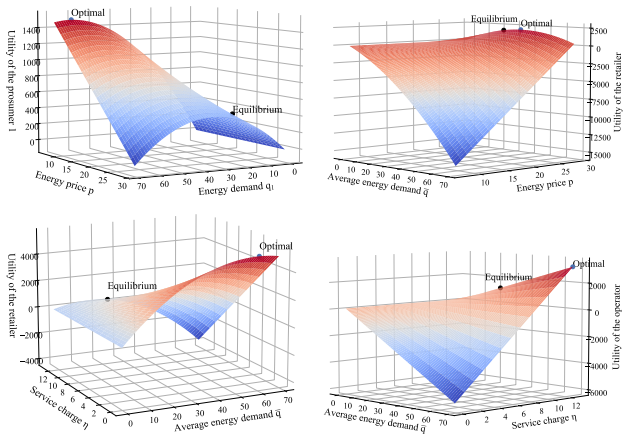


Fig. 7. Performance of the system model.

considering only unilateral model and game equilibrium under HDPG. For each unilateral model, the entity has its optimal actions without considering other entities' policies. In Stackelberg game, to reach a relatively fair equilibrium, each entity makes its actions to cope with other entities' policies, which results in the utility being far from the single optimal solution. However, the actual optimization problems in energy trading are complex game processes instead of single objective optimizations of multiparties.

Under the unified pricing mechanism, prosumers' optimal demands are quite similar. Thus, we chose one of the prosumers to present. Both the leader operator and the subleader retailer obtain a splendid utility, especially the top leader with the first mover advantage. Under the operator's high-level policy and retailer's middle-level policy, prosumers also converge to a relatively good solution. In addition, the convergence sequence of different entities accords with the action sequence in the three-stage Stackelberg game, which is reflected in that the leader rapidly converges to its solution.

2) Utility Performance: To present the remarkable performance of the HDPG algorithm for the three-stage Stackelberg game, we compare it with some prevalent deep reinforcement learning algorithms as shown in Fig. 8. We average the result data from ten times of experiments to reduce the random error. From Fig. 8(a), it is obvious that HDPG gains more total reward of all entities than PPO, SAC, and DQN algorithms. Different

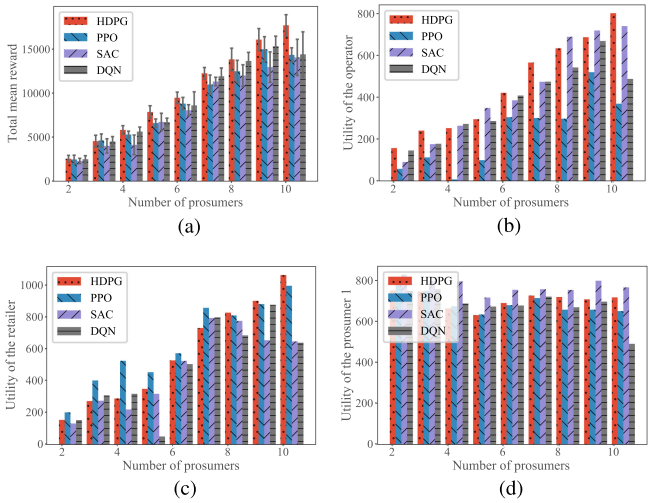


Fig. 8. Performance contrast on different types of reward. (a) Contrast on total reward. (b) Contrast on operator's reward. (c) Contrast on retailer's reward. (d) Contrast on prosumers' reward.

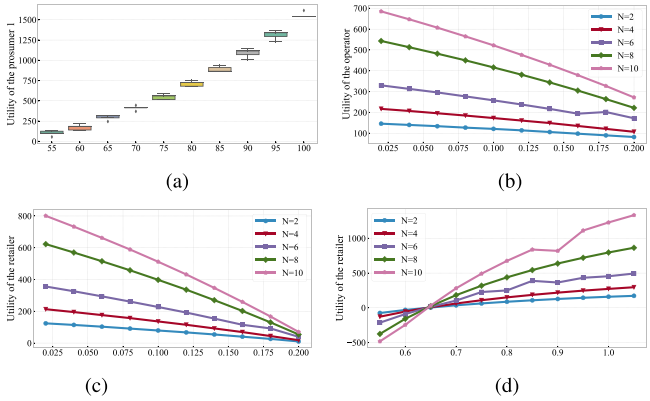


Fig. 9. Utilities under different parameters. (a) $\mathcal{U}^p(q)$ vs. δ. (b) $\mathcal{U}^o(\eta)$ vs. ϕ. (c) $\mathcal{U}^r(p)$ vs. ϕ. (d) $\mathcal{U}^r(p)$ vs. ξ_c.

algorithms have respective characteristics. For example, SAC assists prosumers to get the highest utility, while it gets poor performance in the operator's policy. In comparison, HDPG assists the leader operator and the subleader retailer achieves higher utilities. Besides, the prosumers also achieve a relatively good utility with HDPG. The hierarchical design of multiagent's action and learning process assists agents to learn their policies according to the rival policies, which potentially contributes to the performance of HDPG.

Then, we analyze the parametric sensitivity of the utilities as the number of prosumers varies. On account of that the utility of prosumers shows not so much sensibility with the number of participants, we use the boxplot to meticulously describe the influence of prosumers' willingness on the utility of prosumers. As shown in Fig. 9(a), it is sensible for prosumers with high-value industrial production to participate in the industrial energy market. From Fig. 9(b) and (c), we observe that the transmission loss rate ϕ decides the utilities of the operator and retailer to a large extent. Thus, it is of great significance to reduce the transmission loss rate using more advanced technology in

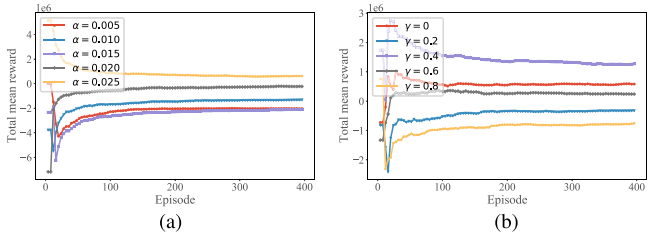


Fig. 10. Total reward under different parameters. (a) Total reward under different α. (b) Total reward under different γ.

energy transmission and distribution network. As the increasing prosumers, the utilities of the operator and retailer rise steadily, while the utility of each prosumer shows a slight downward trend. It should be noted that the increase of prosumers may affect each prosumers' policy, which results in some fluctuations to the utilities of all the entities. Fig. 9(d) presents the variation trend of the retailer's utility under different efficiencies of energy storage, which is a crucial part in BSDEI. Low energy storage efficiency may lead to negative utility, while higher efficiency of the energy storage devices achieves better utility of the retailer. However, the cost and difficulty of reducing transmission loss rate and raising energy storage efficiency are also hard obstructions in BSDEI.

As shown in Fig. 10, we further find different values of learning rate α and future discount factor γ has some influence on the convergence performance of the HDPG algorithm. After evaluation, it is better to fix $\alpha = 0.025$ and $\gamma = 0.6$.

VI. Conclusion

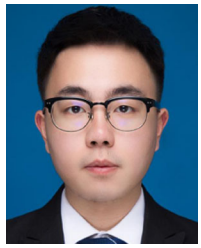
To better provide industrial energy trading services and minimize the impact of DER accessing to EI, we proposed a blockchain-assisted software defined EI architecture. For the industrial energy market in BSDEI, we used blockchain to ensure the reliability between trading entities and avoids the risk of a single point of failure. To jointly optimize the utilities of the operator, retailer, and industrial prosumers in BSDEI, we modeled the whole energy trading process as a three-stage Stackelberg game, with the proof of existence and uniqueness for the SE. Then, we designed a hierarchical MARL algorithm HDPG to solve the Stackelberg game under incomplete information, which exceeds about 18% in total mean reward than that of some prevalent algorithms. To better serve for demand of industrial energy trading, we implemented the industrial energy trading system using a middleware blockchain platform, by deploying the distributed energy market smart contract on the consortium blockchain.

References

[1] H. Zhang, Y. Li, D. W. Gao, and J. Zhou, "Distributed optimal energy management for energy Internet," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3081–3097, Dec. 2017.

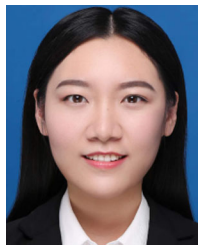
[2] G. S. Aujla, M. Singh, A. Bose, N. Kumar, G. Han, and R. Buyya, "BlockSDN: Blockchain-as-a-service for software defined networking in smart city applications," *IEEE Netw.*, vol. 34, no. 2, pp. 83–91, Mar./Apr. 2020.

[3] J. Luo, Q. Chen, F. R. Yu, and L. Tang, "Blockchain-enabled software-defined industrial Internet of Things with deep reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5466–5480, Jun. 2020.
[4] Z. Xu, G. Han, L. Liu, M. Martínez-García, and Z. Wang, "Multi-energy scheduling of an industrial integrated energy system by reinforcement learning-based differential evolution," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1077–1090, Sep. 2021.
[5] C. Qiu, H. Yao, X. Wang, N. Zhang, F. R. Yu, and D. Niyato, "AI-chain: Blockchain energized edge intelligence for beyond 5G networks," *IEEE Netw.*, vol. 34, no. 6, pp. 62–69, Nov./Dec. 2020.
[6] F. Shamieh, X. Wang, and A. R. Hussein, "Transaction throughput provisioning technique for blockchain-based industrial IoT networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 3122–3134, Oct.–Dec. 2020.
[7] S. D. Manshadi and M. E. Khodayar, "A hierarchical electricity market structure for the smart grid paradigm," *IEEE Trans. Smart Grid*, vol. 7, no. 4, pp. 1866–1875, Jul. 2016.
[8] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial," *IEEE Commun. Surv. Tut.*, vol. 23, no. 2, pp. 1226–1252, Apr.–Jun. 2021.
[9] 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.
[10] C. Qiu, H. Yao, C. Jiang, S. Guo, and F. Xu, "Cloud computing assisted blockchain-enabled Internet of Things," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2019.2930259.
[11] W. Zhong, R. Yu, S. Xie, Y. Zhang, and D. H. Tsang, "Software defined networking for flexible and green energy Internet," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 68–75, Dec. 2016.
[12] D. Jin *et al.*, "Toward a cyber resilient and secure microgrid using software-defined networking," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2494–2504, Sep. 2017.
[13] 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.
[14] Z. Li, Z. Yang, S. Xie, W. Chen, and K. Liu, "Credit-based payments for fast computing resource trading in edge-assisted Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6606–6617, Aug. 2019.
[15] B. Gu, X. Yang, Z. Lin, W. Hu, M. Alazab, and R. Kharel, "Multiagent actor-critic network-based incentive mechanism for mobile crowdsensing in industrial systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6182–6191, Sep. 2021.
[16] Z. Xiong, S. Feng, D. Niyato, P. Wang, Y. Zhang, and B. Lin, "A Stackelberg game approach for sponsored content management in mobile data market with network effects," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5184–5201, Jun. 2020.
[17] A. Paudel, K. Chaudhari, C. Long, and H. B. Gooi, "Peer-to-peer energy trading in a prosumer-based community microgrid: A game-theoretic model," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6087–6097, Aug. 2019.
[18] X. Lu, X. Xiao, L. Xiao, C. Dai, M. Peng, and H. V. Poor, "Reinforcement learning-based microgrid energy trading with a reduced power plant schedule," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10728–10737, Dec. 2019.
[19] T. Long, X. Ma, and Q.-S. Jia, "Bi-level proximal policy optimization for stochastic coordination of EV charging load with uncertain wind power," in *Proc. IEEE Conf. Control Technol. Appl.*, 2019, pp. 302–307.
[20] B. Zhang *et al.*, "Soft actor-critic -based multi-objective optimized energy conversion and management strategy for integrated energy systems with renewable energy," *Energy Convers. Manage.*, vol. 243, 2021, Art. no. 114381.
[21] Y. Liu, X. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, "From industry 4.0 to agriculture 4.0: Current status, enabling technologies, and research challenges," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4322–4334, Jun. 2021.
[22] J. Lee, J. Guo, J. K. Choi, and M. Zukerman, "Distributed energy trading in microgrids: A game-theoretic model and its equilibrium analysis," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3524–3533, Jun. 2015.
[23] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.
[24] Y. Liu, H. Wang, M. Peng, J. Guan, and Y. Wang, "An incentive mechanism for privacy-preserving crowdsensing via deep reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8616–8631, May 2021.



Yifan Cao (Student Member, IEEE) received the B.S. degree in electrical engineering and automation in 2020 from the School of Electrical and Information Engineering, Tianjin University, Tianjin, China, where he is currently working toward the M.S. degree in computer science and technology with the College of Intelligence and Computing.

His current research interests include energy trading, blockchain, and reinforcement learning.



Xiaoxu Ren (Student Member, IEEE) received the B.S. degree in information and computing science from the College of Science, Inner Mongolia University of Technology, Hohhot, China, in 2016. She is currently working toward the Ph.D. degree in computer applications technology with the College of Intelligence and Computing, Tianjin University, Tianjin, China.

Her current research interests include machine learning, computing power networking, and blockchain.

Chao Qiu (Member, IEEE) received the B.S. degree in communication engineering from China Agricultural University, Beijing, China, in 2013, and the Ph.D. degree in information and communication engineering from the Beijing University of Posts and Telecommunications, Beijing, in 2019.

From 2017 to 2018, she was with Carleton University, Ottawa, ON, Canada, as a Visiting Scholar. She is currently a Lecturer with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, China. Her current research interests include machine learning, software defined networking, and blockchain.

Xiaofei Wang (Senior Member, IEEE) received the master's and Ph.D. degrees in computer science and engineering from Seoul National University, Seoul, South Korea, in 2008 and 2013, respectively.

He was a Postdoctoral Fellow with The University of British Columbia, Vancouver, BC, Canada, from 2014 to 2016. He is currently a Professor with the Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology, Tianjin University, Tianjin, China. Focusing on the research of social-aware cloud computing, cooperative cell caching, and mobile traffic offloading, he has authored more than 140 technical papers in publications such as IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, IEEE COMMUNICATIONS, IEEE TRANSACTIONS ON MULTIMEDIA, *IEEE International Conference on Computer Communications*, and *IEEE International Conference on Sensing, Communication and Networking*.

Dr. Wang was the recipient of the "Scholarship for Excellent Foreign Students in IT Field" by NIPA of South Korea from 2008 to 2011, the "Global Outstanding Chinese Ph.D. Student Award" by the Ministry of Education of China in 2012, and the Peiyang Scholar from Tianjin University. In 2017, he was the recipient of the "Fred W. Ellersick Prize" from the IEEE Communication Society.