BWKA: A Blockchain-based Wide-area Knowledge Acquisition Ecosystem

Yang Xu, *Member, IEEE,* Jianbo Shao, *Graduate Student Member, IEEE,* Jia Liu, *Senior Member, IEEE,* Yulong Shen, *Member, IEEE,* Tarik Taleb, *Senior Member, IEEE* and Norio Shiratori, *Life Fellow, IEEE*

Abstract—Benefiting from the booming of big data and artificial intelligence (AI) technologies, data-as-a-service is gradually transforming into knowledge-as-a-service. Extracting knowledge from massive raw data is becoming a popular paradigm to save network resources and improve efficiency, and establishing knowledge markets is receiving increasing attention from academia and industry. In this paper, we propose a one-stop knowledge acquisition ecosystem termed BWKA that covers the whole process from upper-layer knowledge trading to underlying knowledge generation. In the knowledge trading process, the knowledge-as-a-service platform (KSP) is the buyer and publishes knowledge demands to multiple local knowledge sellers (LKSs). In the knowledge generation process, each LKS aggregates data from its sensors and then trains data into knowledge according to the KSP's requirements. We resort to blockchain technology and provide a series of tailored operating rules and functions to protect the truthfulness of data gathering and the fairness of knowledge trading. In addition, we introduce incentive mechanisms to stimulate selfish and rational entities in the BWKA ecosystem to participate in knowledge acquisition. To analyze the strategic interactions among entities theoretically, we develop a nested hierarchical game model, where the upper-layer knowledge trading is evaluated based on the Contract Theory, and the lower-layer knowledge generation is formulated as a two-stage Stackelberg game. By solving the nested hierarchical game in a backward inductive way, we identify the optimal strategy for each entity in closed form. Experiments on the Ethereum blockchain and simulation results demonstrate the practical operability and outstanding performance of the BWKA ecosystem.

Index Terms—Knowledge acquisition ecosystem, blockchain, smart contract, incentive mechanism, hierarchical game.

1 Introduction

With the advent of the data-explosion era, tremendous redundant data has been generated in all walks of life. To make better use of excessive data, extracting meaningful information from big data to improve human society services has rapidly gained popularity in recent years [1] [2] [3]. Such a trend has led to the emergence of many fascinating applications such as medical diagnosing, environment monitoring, market predictions, smart cities, and smart business/inventory product management [4] [5] [6]. In this context, the network information service mode has been gradually transformed from "Data-as-a-Service" into "Knowledge-as-a-Service", which is regarded as a rising star in the family of XaaS (Everything-as-a-Service) [7] [8].

Knowledge is the embodiment of data intelligence and represents the real value of data. For instance, the real-time traffic congestion and accident information on Google Waze [9] (a community-based GPS system) is refined from the locations, densities, and trajectories gathered by mobile devices. A knowledge-as-a-service platform (KSP) serves its users by providing them with knowledge. For example, GasBuddy presents the knowledge on cheap gas stations [10], Pavemint shows the information on available parking spaces [11], and WiFi Finder provides the knowledge on free WiFi hotspots [12]. However, due to resource constraints, the KSP generally acquires knowledge through commissioned production [13], i.e., distributing knowledge acquisition tasks to local knowledge sellers (LKSs) rather than producing knowledge by itself directly, which leads to an urgent demand for meticulous knowledge acquisition mechanisms.

Over the past three years, some efforts have been made to investigate knowledge acquisition through different approaches and perspectives, e.g., crowdsensing [14], crowdlearning [15], federated learning [16], truth discovery [17], information timeliness analysis [18], delay management in data collection [19], and so on. Although these works have remarkably promoted progress in this research field, they mostly consider that the involved entities are voluntary to participate in knowledge acquisition. However, due to the inherent selfishness and rationality of entities, they may be reluctant to engage in knowledge acquisition tasks unless they are satisfactorily paid. Therefore, when designing practical knowledge acquisition systems, the economic and strategic interactions among entities need to be comprehensively addressed.

- *Yang Xu and Yulong Shen are with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: yxu@xidian.edu.cn; ylshen@mail.xidian.edu.cn).*
- *Jianbo Shao is with the Department of Computer Science and Technology, Shanghai University of Finance and Economics, Shanghai 200433, China (email: jianboshao.sufe.cs@gmail.com)*
- *Jia Liu is with the Center for Strategic Cyber Resilience Research and Development, National Institute of Informatics, Tokyo 101-8430, Japan (email: jliu@nii.ac.jp).*
- *Tarik Taleb is with Ruhr University Bochum, Bochum, 44801 Germany (e-mail: tarik.taleb@rub.de).*
- *Norio Shiratori is with the Research and Development Initiative, Chuo University, Tokyo 112-8551, Japan (e-mail: norio.shiratori.e8@tohoku.ac.jp).*

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62372361; in part by JSPS KAKENHI Grant Number JP23K16877; in part by the Project of Cyber Security Establishment with Inter-University Cooperation; in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY24F020011; in part by the AerOS project funded by the European Union's Horizon Europe, the EU's key funding program for research and innovation under Grant No. 101069732, and also conducted at ICTFICIAL Oy, Finland; and in part by Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University.
(Corresponding author: Jia Liu.)

1.1 Motivation

By now, there are some preliminary works to explore the knowledge acquisition entities' interactions by using game theory [8], [20], contract theory [21], and auction theory [22]. They mainly focus on the pricing problem related to knowledge sharing or delivery through an online market, i.e., the interactions between the KSP and LKSs, while largely neglecting the knowledge generation process that involves the interactions between data owners and LKSs, which leads to a lack of understanding of a complete knowledge acquisition ecosystem. Moreover, in commissioned production, the operational effectiveness of completing a task is a crucial indicator for evaluating an LKS. From the KSP's perspective, an efficient LKS deserves more payments. Therefore, differentiated incentive mechanism design is a key concern of the KSP. Recent research also reveals the vulnerability of online knowledge/data trading markets, in which the open and anonymous environment may stimulate participants' misbehaviors (e.g., abortion) for reaping unfair profits [23]. Thus, how to protect the truthfulness of data gathering and the fairness of knowledge trading is another critical issue to be addressed for establishing the knowledge acquisition ecosystem.

1.2 Contributions

Motivated by the above statements, this paper proposes BWKA, a blockchain-based wide-area knowledge acquisition ecosystem, which contains a complete process from the underlying data sensing, aggregating, and knowledge training, to the upper-layer knowledge trading. In the BWKA ecosystem, the KSP publishes knowledge generation tasks into a series of contracts with differentiated requirements and prices, and uploads them on a blockchain. Each LKS makes its independent decision to sign a customized contract with the KSP, and then mobilizes nearby sensors to collect data. Each sensor is a selfish and rational individual to decide the amount of data transmitted to the LKS. To ensure efficient data sensing as well as truthful and transparent data gathering, a blockchain is established to record the collected data and provide sensors with rewards. After aggregating the data, the LKS trains the data into knowledge and sells it to the KSP. By employing the blockchain with smart contracts, we design a novel trading scheme that forces LKSs to disclose their operational effectiveness honestly and prevents misbehaviors of the buyer and sellers during the trading. For the proposed BWKA ecosystem, we also develop a theoretically analytical framework to study the interactions among the KSP, LKSs, and sensors to identify their optimal strategies and utilities.

In summary, this paper makes the following contributions:

- *Novel Knowledge Acquisition Ecosystem Design:* To the best of our knowledge, this is the first work to investigate a complete knowledge acquisition ecosystem from a comprehensive perspective, which includes not only underlying knowledge generation but also upper-layer knowledge trading. We employ two types of blockchains and smart contracts, as well as elaborately design the corresponding operating rules and functions to achieve secure, transparent, and truthful data gathering and reliable and fair knowledge trading. In addition, incentive mechanisms are incorporated into the BWKA ecosystem to stimulate entities to participate in the knowledge acquisition works.
- *Nested Hierarchical Game Modeling:* We develop a nested hierarchical game model to characterize the behaviors of entities involved in the BWKA ecosystem and study their strategic interactions. This game model is composed of two layers, where the upper-layer model formulates the interactions between the KSP and LKSs in the knowledge trading process based on the Contract Theory, and the lower-layer model formulates the interactions between each LKS and its nearby sensors in the knowledge generation process as a two-stage Stackelberg game. In the Stackelberg game, the LKS in Stage I acts as the leader to provide sensors with rewards, while the sensors in Stage II act as the follower to play a non-cooperative game.
- *Optimal Strategy Solution:* We utilize the backward induction method to solve the hierarchical game, such that the closed-form optimal strategies of sensors, LKSs, and the KSP are identified respectively. In particular, we prove the existence and uniqueness of Nash Equilibrium in the non-cooperative game of sensors, and investigate the concavity of LKS's utility function under the constraint of completing the knowledge generation task. Moreover, the KSP's contracts can satisfy the individual rationality (IR) and incentive compatibility (IC) properties, which ensure that the cost of each LKS is properly compensated and each LKS truthfully reveals its operational effectiveness, respectively.
- *Experiments and Simulations:* We implement the BWKA ecosystem on Sepolia Test Network to demonstrate its practical operability. Simulation results are also presented to show the behaviors and performance of entities in the BWKA ecosystem. In addition, we conduct comparative experiments to demonstrate that the proposed knowledge trading scheme can enable the KSP to achieve higher profits than the classical linear pricing scheme.

1.3 Paper Organization

The remainder of this paper is organized as follows. We introduce related work in Section 2. Section 3 presents the BWKA ecosystem. Section 4 develops the nested hierarchical game model. The Stackelberg game analysis for knowledge generation and the contract analysis for knowledge trading are presented in Section 5 and Section 6, respectively. Section 7 shows the simulation results, followed by the conclusion and future work in Section 8.

2 Related Work

With the popularization of edge intelligence, knowledge trading has been introduced into several fields, like IoT systems [8], [24] and the Internet of Vehicles (IoV) [16], to accelerate social industrial development and improve industrial economic benefits. In [8], a peer-to-peer knowledge market for knowledge paid sharing was developed to break islands of knowledge and make knowledge tradable in edge-AI-enabled IoT. Considering energy-constrained IoT, Lin *et al.* [25] proposed underlying energy-knowledge trading mechanisms with the optimal economic incentives and power transmission strategies by constructing a two-stage Stackelberg game. In [16], Chai *et al.* developed a hierarchical blockchain framework and a hierarchical federated learning algorithm for knowledge sharing in IoV, which was modeled as a trading market process by utilizing the multi-leader and multi-player game.

The open and anonymous online environment may bring security risks during knowledge trading. To address such issues, blockchain technology has been widely applied to establish distributed trust. Li *et al.* [26] proposed a blockchain-enabled

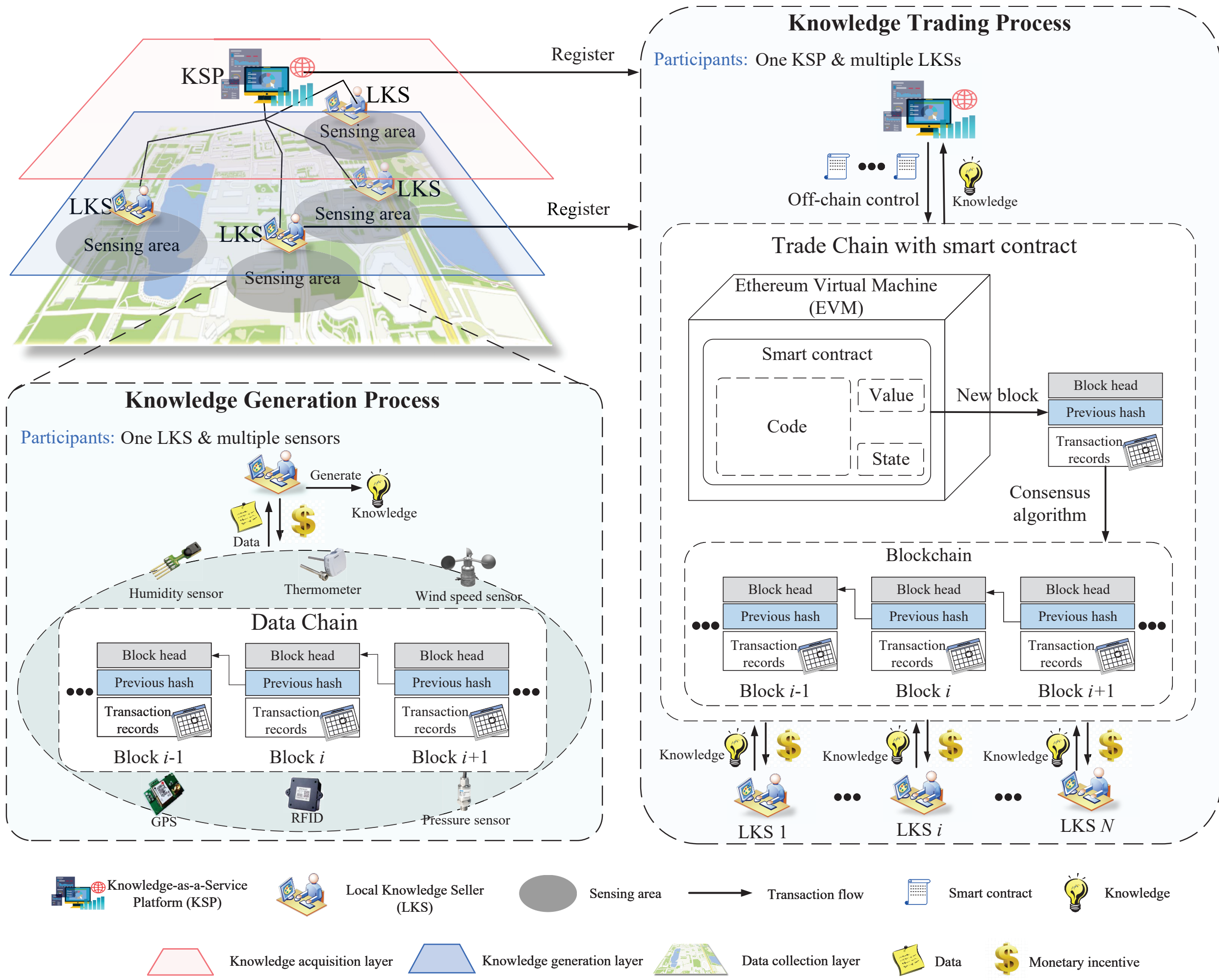


Fig. 1. Blockchain-based wide-area knowledge acquisition (BWKA) ecosystem.

secure energy trading system, in which a consortium blockchain architecture was introduced to guarantee verifiable fairness during energy trading. Fan *et al.* [27] developed a hybrid blockchain-based resource trading system, which combines the advantages of both public and consortium blockchains to enable credible payment transactions between requesters and edge nodes. In [28], a blockchain-based spectrum trading system was proposed, where the sharding technique was adopted to design a consensus mechanism to prevent malicious attacks (e.g., double-spending attacks) during spectrum trading.

Another highly related research direction is the incentive mechanism design to facilitate knowledge trading and data collection. Specifically, Sun *et al.* [29] designed a personalized privacy-preserving knowledge obtaining scheme based on the Contract Theory, where workers protect their privacy by adding a perturbation to their data, and the knowledge requester utilizes a truth discovery mechanism to aggregate data and obtain knowledge. From a behavioural economics perspective, Liu *et al.* [6] proposed an incentive mechanism to motivate participants to collect data in unpopulated areas for location-based crowdsensing systems. Nie *et al.* [30] considered crowdsensing in social networks and designed incentive mechanisms for profit maximization based on a multi-leader and multi-follower Stackelberg game approach. In [31], the authors proposed a privacy-preserving secure spectrum trading scheme for UAV-assisted cellular networks by leveraging the blockchain technology, where a pricing-based incentive mechanism and a Stackelberg game-based spectrum blockchain framework were developed to improve the trading environment and maximize profits for both the primary mobile network operator and UAV operators.

It is worth noting that this work is distinguishable from the existing ones in the sense that we construct a complete knowledge acquisition ecosystem from a more comprehensive perspective, which includes not only knowledge trading but also underlying knowledge generation. We employ two types of blockchains to protect the truthfulness of data gathering and the reliability of knowledge trading, respectively, and elaborately design the corresponding operating rules and functions. In addition, we establish a nested hierarchical game model to optimize the strategic interactions among related entities in different processes.

3 Blockchain-based Wide-area Knowledge Acquisition Ecosystem

In this section, we first introduce the overall architecture of the blockchain-based wide-area knowledge acquisition ecosystem, and then elaborate on participants' interactions involved in the knowledge generation process and knowledge trading process, respectively. The main notations are listed in Table 1.

TABLE 1
Main Notations.

Notation	Description
$\mathcal{N}$	Set of LKSs
$\mathcal{M}_i$	Set of sensors in the sensing area that is managed by LKS i
V_i	Dataset value of LKS i
X_{S_m}	Data size collected by sensor m
$\psi(V_i)$	Knowledge value of LKS i
η_m	Position weight parameter of sensor m
$\boldsymbol{T}$	Knowledge value specified by the KSP
$\boldsymbol{P}$	Price coefficient corresponds to $\boldsymbol{T}$
$\boldsymbol{\theta}$	Type vector of LKSs
U_{KSP}	Utility of KSP
U_{LKS_i}	Utility of LKS i
U_{S_m}	Utility of sensor m
P_{KNO}	Unit knowledge earning from other markets
R_i	Rewards provided by LKS i
R_i^*	Optimal block rewards paid by LKS i
c	Cost of collecting data per unit size
$\mathbf{X}_S$	Strategy profile containing all the strategies of sensors in $\mathcal{M}_i$
$\mathbf{X}_{-S_m}$	Strategy profile excluding sensor m
$\widehat{U}_i^{NC}$	Utility of LKS i when it refuses $\{T_i, P_i\}$

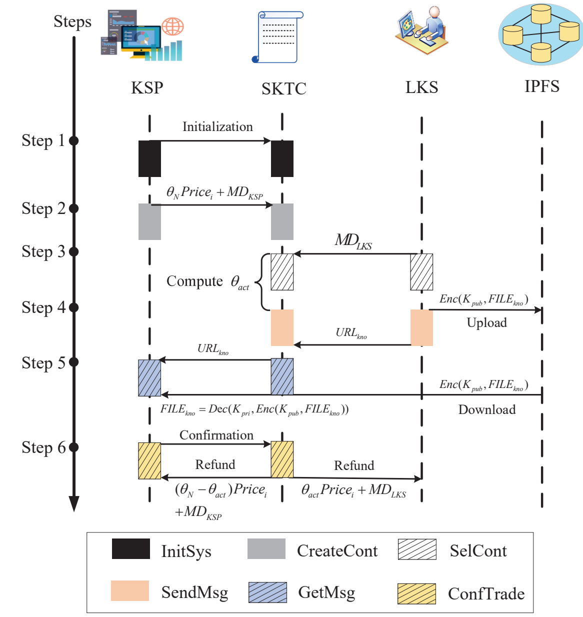


Fig. 2. SKTC-based knowledge trading process.

3.1 Architecture of BWKA Ecosystem

As illustrated in Fig. 1, we propose a blockchain-based wide-area knowledge acquisition (BWKA for short) ecosystem that is composed of a Knowledge-as-a-Service Platform (KSP) and N edge-AI-enabled Local Knowledge Sellers (LKSs). The KSP and LKSs are deployed randomly in a wide area, while each LKS is in control of one data sensing area, which contains multiple types of sensors such as RFIDs, GPSs, and thermometers. In the BWKA ecosystem, the KSP announces a series of knowledge acquisition task-price pairs $\{\boldsymbol{Task}, \boldsymbol{Price}\}$ to LKSs, where $\boldsymbol{Task} = \{Task_1, Task_2, \cdots, Task_N\}$ and $\boldsymbol{Price} = \{Price_1, Price_2, \cdots, Price_N\}$ denote the KSP's task vector and price vector for LKSs, respectively. Once accepting a specific task-price pair, an LKS stimulates data collection of sensors in the corresponding sensing area by executing some incentive mechanism. Then, the LKS aggregates data from sensors and employs the edge-AI to extracts knowledge (e.g., valuable information), which is called the *knowledge generation process*. Afterward, the LKS sells its knowledge to and receives monetary rewards from the KSP according to the admitted task-price pair and pre-agreed trading regulations, which is called the *knowledge trading process*.

It is worth noting that if the gathered data or traded knowledge is a fraud, it inevitably leads to a severe crisis of confidence in the knowledge acquisition ecosystem. Considering the traceability, transparency, and tamper-proof features of blockchain, we develop a blockchain-based trusted knowledge management framework to prevent fraudulent data sensors and knowledge providers. Specifically, we establish two types of blockchains, i.e., N data chains (DCs), and one trade chain (TC) with a deposit-and-refund mechanism [23]. A DC is created in the knowledge generation process and maintained by an LKS and its corresponding sensors to ensure truthful data gathering, while the TC is created in the knowledge trading process and maintained by all LKSs and the KSP to protect the participants from dishonest trading. Next, we introduce the DC-based knowledge generation process and TC-based knowledge trading process in detail.

3.2 Data Chain-based Knowledge Generation Process

In the knowledge generation process, the sensors in a sensing area are the data collectors and providers, the LKS is the data aggregator and the knowledge generator, and the DC is a distributed data-sharing ledger. Considering that it is burdensome for sensors to execute the Proof of Work (PoW) consensus due to their computing capability limitation, we employ the Proof of Contribution (PoC) consensus mechanism that can fit non-monetary blockchain systems perfectly [32]. As the infrastructure provider of DC i, LKS i incentivizes sensors to participate in data collection by providing block rewards to the PoC winner. As the reaction, sensors compete with each other individually for the block rewards by making efforts to collect data. The PoC consensus mechanism characterizes and quantifies the actions and performance of sensors, and the sensor with the largest contribution wins the block rewards and gets the right to generate a new block in DC i.

3.3 Trade Chain-based Knowledge Trading Process

We stipulate that knowledge trading in the BWKA ecosystem is carried out over contracts, where the KSP pays rewards to buy knowledge from LKSs. The KSP regards the operational effectiveness for generating knowledge as the type[1] of an LKS, and due to the discrepancy of AI algorithms, training data, and computing capabilities, the KSP identifies N different types for LKSs. Let $\boldsymbol{\theta} = \{\theta_1, \theta_2, \cdots, \theta_N\}$ denote the type vector of LKSs, where θ_i is in ascending order, i.e., $\theta_1 \leq \theta_2 \leq \cdots \theta_N$. A larger type LKS indicates higher operational effectiveness, which brings greater profit to the KSP, and thus the KSP will provide this LKS with more rewards. However, the operational effectiveness

1. In contract theory, "type" is a commonly used term to describe the attributes or characteristics of participants, the meaning differs from the daily usage of the word "category/type". Throughout this paper, when "type" is associated with LKS, it specifically refers to the operational effectiveness of the LKS for generating knowledge. That is, an LKS corresponds to a "type", indicating its distinct attribute.

InitSys: $t_s = 0,\ t_e = 0,\ level = 0,\ State = 0,\ URL = '\ ',\ SellerAddr = '\ '$

CreateCont: For $i \in \{1, 2, ..., N\}$ do
  On receiving ($\boldsymbol{Task}, \boldsymbol{Price}, \boldsymbol{\theta}, MD_{KSP}, MD_{LKS}, N, t_{dur}$) from KSP
    Require ($msg.sender = Addr_{KSP}$)
    Require ($msg.value = \theta_N Price_i + MD_{KSP}$)
      Deploy SKTC i
End for

SelCont: While !*State* do
  On receiving () from some LKSs
    Require ($msg.sender \in AddrList_{LKS}$)
    Require ($msg.value = MD_{LKS}$)
      Freeze total assets: $\theta_N \mathrm{Price}_i + MD_{KSP} + MD_{LKS}$
      $State$=1
      t_s=now
      $SellerAddr$=$msg.sender$
end

SendMsg: On receiving (URL_{kno}) from some LKSs
  Require ($msg.sender = SellerAddr$)
  URL=URL_{kno}
  t_e=now
  $level = \left\lceil N(1 - \frac{t_e - t_s}{t_{dur}}) \right\rceil$
  $\theta_{act} = \theta_{level}$

GetMsg: Upon receiving () from KSP
  Require ($msg.sender = Addr_{KSP}$)
  Require ($URL\ != '\ '$)
    Return URL

ConfTrade: Upon receiving () from KSP
  Require ($msg.sender = Addr_{KSP}$)
  Require ($SellerAddr\ != '\ '$)
    Send ((θ_N-θ_{act})$Price_i + MD_{KSP}$) Ethers to $Addr_{KSP}$
    Send ($\theta_{act} Price_i + MD_{LKS}$) Ethers to $SellerAddr$

Fig. 3. Functions of a smart knowledge trade contract.

is an LKS's private information that the KSP does not know its instantaneous situation exactly. To this end, we design a smart knowledge trade contract (SKTC) that can determine the real type of LKSs automatically to achieve a reliable and efficient knowledge trading process.

Fig. 2 shows the overall process of SKTC-based knowledge trading, and Fig. 3 presents the functions of an SKTC in detail. In Fig. 3, "on receiving (*msg*) from KSP (resp., LKSs)" indicates that the function accepts a message from the KSP (resp. LKSs), "upon receiving () from KSP" indicates that the function accepts a transaction from the KSP without parameters [23], *msg.sender* is the Ethereum address of the transaction sender, and *msg.value* is the Ethers attached to the current transaction. Then, we elaborate on each step and the corresponding operations of the SKTC-based knowledge trading process.

1) *System Initialization:* The KSP and LKSs become legitimate entities of the TC after registering in the BWKA ecosystem. Then, each entity obtains a unique address, public and private keys, and a digital signature by adopting the elliptic curve digital signature algorithm (ECDSA) [33]. Denote K_{pub} and K_{pri} the public and private keys of KSP, respectively. Let $Addr_{KSP}$ and $AddrList_{LKS}$ denote the address of KSP and the address list of LKSs, respectively, t_s and t_e denote the start time and end time of a task, respectively. $level$ is an integer variable to record the level of operational effectiveness of an LKS who accepts the contract. $State$ is a boolean state variable, where $State = 1$ indicates the contract is accepted by an LKS, while $State = 0$ indicates the contract is not occupied by any LKSs. URL and $SellerAddr$ are string variables used to store the Uniform Resource Locator (URL) of the knowledge and the address of a knowledge seller, respectively. The KSP executes *InitSys* function to initialize t_s, t_e, $level$, $State$, URL, and $SellerAddr$.

2) *Deploy Smart Contracts:* The KSP calls *CreateCont* function to deploy N SKTCs, which stipulate that knowledge trading follows the deposit-and-refund discipline. Thus, the KSP sets its monetary deposits MD_{KSP} and the LKS's deposits MD_{LKS}, the type vector of LKSs $\boldsymbol{\theta}$, and the maximum tolerable task duration t_{dur} in advance. Meanwhile, the KSP also strategizes the task-price pairs $\{\boldsymbol{Task}, \boldsymbol{Price}\}$, expecting to maximize the profit that can be achieved. When deploying SKTC i, the KSP announces the task-price vector $\{Task_i, Price_i\}$ and attaches $\theta_N Price_i + MD_{KSP}$ Ethers, i.e., $msg.value = \theta_N Price_i + MD_{KSP}$, to guarantee that the funds in SKTC i are sufficient to award the highest type (i.e., the most effective) LKS.

3) *Select Optimal Contract:* After observing the task-price pairs of SKTCs, the LKS selects its optimal SKTC and then triggers *SelCont* function with MD_{LKS} Ethers attached, i.e., $msg.value = MD_{LKS}$. Once *SelCont* function is triggered, the SKTC will freeze all the assets $\theta_N Price_i + MD_{KSP} + MD_{LKS}$ and save the current LKS's address as $SellerAddr$. Simultaneously, the current time will be stored in t_s as the task start time.

4) *Send Knowledge Message:* Once accomplishing the task according to the selected SKTC (i.e., complete the knowledge generation process), the LKS encrypts the knowledge $FILE_{kno}$ by using KSP's public key K_{pub} and uploads the encrypted knowledge $Enc(K_{pub}, FILE_{kno})$ to the Interplanetary File System (IPFS) [34], which is a peer-to-peer distributed file system. Afterward, the LKS calls *SendMsg* function to transmit the IPFS knowledge link URL_{kno} to the SKTC. Meanwhile, the current time is stored in t_e as the task end time. The level of operational effectiveness is automatically calculated as $level = \lceil N(1 - \frac{t_e - t_s}{t_{dur}}) \rceil$. Then, the actual type of the LKS is identified as $\theta_{act} = \theta_{level}$.

5) *Get Knowledge Message:* The KSP calls *GetMsg* function to obtain URL_{kno}. It is noteworthy that, as depicted in Fig. 3, the invocation condition of *GetMsg* function is "Require($msg.sender = Addr_{KSP}$)", indicating that only the KSP is permitted to invoke *GetMsg* function, and consequently, has access permission to the URL of the uploaded knowledge. Then, the KSP downloads the encrypted knowledge $Enc(K_{pub}, FILE_{kno})$ from URL_{kno} and decrypts the knowledge by using its private key K_{pri}, i.e., $FILE_{kno} = Dec(K_{pri}, Enc(K_{pub}, FILE_{kno}))$.

6) *Confirmation:* The KSP triggers *ConfTrade* function to confirm that the knowledge trading process is completed successfully. Then, *ConfTrade* function refunds $(\theta_N - \theta_{act})Price_i + MD_{KSP}$ Ethers and MD_{LKS} Ethers to $Addr_{KSP}$ and $SellerAddr$, respectively, and also sends $\theta_{act} Price_i$ Ethers to $SellerAddr$ as the rewards paid by the KSP.

3.4 Complementary Discussions

Note that fraudulent data/knowledge is not the focus of this paper, but it could indeed occur. Introducing certain data security measures can effectively address this issue. For instance, when the KSP or LKS receives knowledge or data, we can leverage the outlier detection technique [35], which allows for the assessment of the quality of submitted data or knowledge, with the results recorded on the blockchain. Meanwhile, by incorporating a reputation mechanism, providers of high-quality data/knowledge would gain elevated reputations, whereas those of lower-quality data/knowledge would experience a decrease in reputation. Due to the inherent traceability and immutability of blockchain, sellers must be mindful of their reputation, otherwise, selling data/knowledge becomes increasingly difficult. These measures can alleviate the fraudulent data/knowledge issue to a certain extent. In addition, for the data/knowledge already received, the truth discovery technique [36] can be employed, which computes the true values from conflicting data sources through multiple iterations, thereby enhancing the overall quality of data and knowledge.

Regarding the encryption and decryption in Step 4 and Step 5, the most commonly used asymmetric encryption algorithm is RSA [37]. Increasing the key length of RSA will improve security, but the computational speed will be significantly slower. The computational time will also increase with the length of the plaintext (e.g., $FILE_{kno}$), especially for the decryption process, the time may increase exponentially. No matter what asymmetric encryption/decryption algorithm is used, additional overhead will be incurred in terms of computation, energy consumption, storage, and latency, leading to an overall decline in the ecosystem operational efficiency. Hence, there exist tradeoffs between data privacy and ecosystem operational efficiency, which deserves further dedicated study. In addition, more sophisticated encryption mechanisms can be used to improve efficiency. For example, hybrid encryption uses a randomly generated key to encrypt the plaintext in a symmetric encryption manner, uses the recipient's public key to encrypt the randomly generated key, and then provides the ciphertext together with the encrypted key to the recipient. Since the computational time of symmetric encryption is much lower than that of asymmetric encryption, and the symmetric key that needs to be encrypted by the public key is usually short, hybrid encryption takes into account both security and operational efficiency. For more details about encryption/decryption mechanisms and their security and efficiency, please kindly refer to [38].

In this work, we implement the blockchain using Ethereum. As blockchain technology evolved, Ethereum emerged in 2015, introducing a broader vision by incorporating smart contracts and decentralized applications (DApps). Ethereum initially employed "computing power" voting, which requires each peer to find a nonce value, such that when hashed with additional block parameters (e.g., a Merkle hash, the previous block hash), the value of the hash is smaller than the current target value. When such a nonce is found, the miner creates the block and forwards it on the network layer to its peers. Other peers in the network verify the new block by computing the hash of the block and checking whether it satisfies the condition. To address scalability concerns and reduce energy consumption, Ethereum has been transitioned to the "stake capital" voting with Ethereum 2.0 in 2022. It leverages validators to create and validate new blocks based on the amount of cryptocurrency they hold. For more details about the implementation of Ethereum, please refer to [39].

4 Game Theory Modeling for BWKA Ecosystem

In the BWKA ecosystem, three distinct types of entities, i.e., sensors, LKSs, and the KSP, collaborate to accomplish the knowledge acquisition tasks together. Given all entities engaged in the BWKA ecosystem are rational, they will interact strategically with the objective of self-interest maximization. Game theory is a vital tool for studying problems involving multiple participants and their interactive behaviors, and note that the entire knowledge acquisition process consists of the underlying knowledge generation process as well as the upper-layer knowledge trading process. Therefore, in this section, we naturally develop a nested hierarchical game model that corresponds to the interactions between entities in different layers, enabling us to comprehensively characterize the entities' strategic behaviors participating in the BWKA ecosystem. The upper-layer model formulates the interactions between the KSP and LKSs in the knowledge trading process based on the Contract Theory, and the lower-layer model formulates the interactions between each LKS and its sensors in the knowledge generation process as a two-stage Stackelberg game.

4.1 Nested Hierarchical Game Model

In this paper, we consider θ_i the private information that the KSP cannot acquire exactly. For ease of presentation, we introduce two concepts below.

For the proposed BWKA ecosystem, let $\mathcal{N} \triangleq \{1, \cdots, N\}$ denote the set of LKSs, $\mathcal{M}_i \triangleq \{1, \cdots, M_i\}$ denote the set of sensors in the sensing area that is managed by LKS i, and $\mathcal{M} \triangleq \cup_{i=1}^{N} \mathcal{M}_i$. LKS i aggregates the data from sensors into a raw dataset and then trains it into knowledge by using edge-AI. The quality of knowledge is intimately associated with the amount of effective information in the raw dataset. Thus, the dataset value V_i of LKS i is determined as

$$V_i = \sigma(M_i) \cdot \sum_{m=1}^{M_i} \eta_m X_{S_m}, \tag{1}$$

where $\sigma(M_i)$ represents the percentage of effective information included in the dataset, which is usually modeled by a SIGMOD function [40], i.e., $\sigma(M_i) = \frac{1}{1+e^{-M_i}}$, X_{S_m} is the data size collected by sensor m, and η_m is a position weight parameter that captures the importance of sensor m's location. Similar to the concept introduced in [8], we identify the knowledge value by evaluating the gap between knowledge and the raw dataset. Following the generic economic law of "diminishing marginal return" [41], the knowledge value function $\psi(V_i)$ is defined as

$$\psi(V_i) = \alpha \ln(1 + \omega V_i), \tag{2}$$

where α and ω are positive parameters. Eq. (2) is consistent with the intuition that a high-value dataset can generate high-quality knowledge.

Next, we provide an overall description of the game theory modeling for the BWKA ecosystem. As illustrated in Fig. 4, we formulate the intricate interactions among KSP, LKSs, and sensors as a two-layer hierarchical game. In the upper layer, the KSP incentivizes LKSs in different regions to gather raw data and train knowledge by announcing the task-price pairs $\{\boldsymbol{Task}, \boldsymbol{Price}\}$. For ease of description, in the following, we rewrite the task-price pairs $\{\boldsymbol{Task}, \boldsymbol{Price}\}$ as $\{\boldsymbol{T}, \boldsymbol{P}\}$, where $\boldsymbol{T} = \{T_1, T_2, \cdots, T_N\}$ is the dataset value required by the KSP for each LKS, i.e., the KSP stipulates the knowledge value ψ by setting T_i, and $\boldsymbol{P} = \{P_1, P_2, \cdots, P_N\}$ is the price coefficient corresponds to $\boldsymbol{T}$. Note that the rewards to an LKS are also related to its type θ, however, the KSP does not know the actual information of LKS's type when configuring $\{\boldsymbol{T}, \boldsymbol{P}\}$. The information asymmetry between the KSP and LKSs may incur a high incentive cost to the KSP [42]. To address this problem, the KSP adopts Contract Theory in designing $\{\boldsymbol{T}, \boldsymbol{P}\}$ [43].

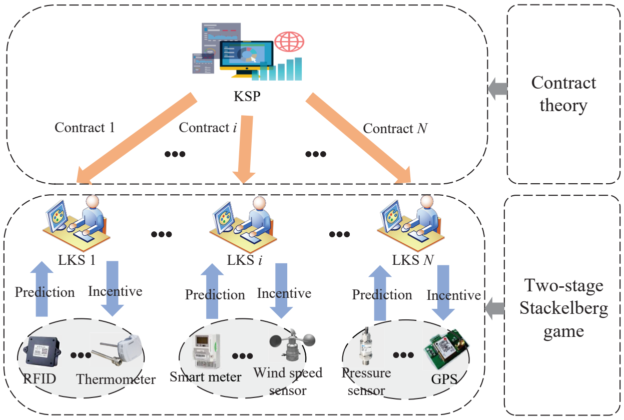


Fig. 4. Nested hierarchical game modeling for the BWKA ecosystem.

In the lower layer, we model the interactions between each LKS and the sensors in the same sensing area as a two-stage Stackelberg game [44]. At Stage I, LKS i plays as a leader who decides the DC's block rewards R_i that is granted to the PoC-wining sensor, for maximizing its utility; at Stage II, each sensor plays as a follower who reacts to the leader's action to strategize its data size X_{S_m}, for maximizing its own utility. The strategy of LKS i is its rewards R_i, and the strategy of sensor m is its data size X_{S_m}. We use $\mathbf{X}_S = (X_{S_1}, X_{S_2}, \cdots, X_{S_{M_i}})$ to denote the strategy profile containing all the strategies of sensors in $\mathcal{M}_i$, and $\mathbf{X}_{-S_m}$ to denote the strategy profile excluding sensor m, i.e., $\mathbf{X}_S = (X_{S_m}, \mathbf{X}_{-S_m})$. Note that due to the selfishness of sensors, at Stage II all sensors compete with each other individually for gaining the block rewards R_i from LKS i. Thus, the competition among sensors at Stage II can be modeled as a non-cooperative game. In the following, we term the two-stage Stackelberg game as the Knowledge Generation (KG) game, and the non-cooperative game at Stage II as the Data Size Determination (DSD) game.

4.2 Utility Functions

We first determine the utility function of KSP. Note that the KSP adopts Contract Theory in designing $\{\boldsymbol{T}, \boldsymbol{P}\}$. The prerequisite constraints implicit in Contract Theory are the necessity to satisfy Incentive Compatibility (IC) and Individual Rationality (IR). The IC constraint indicates that the contract designed for LKS i will definitely be selected by LKS i, otherwise, LKS i's own utility will not be maximized. The IC and IR constraints will be elaborated in Section 6. Let U_{KSP_i} denote the utility of KSP achieved from the task-price pair $\{T_i, P_i\}$. Under the IC constraint, U_{KSP_i} is given by

$$U_{KSP_i} = \mu\phi(\theta_i)\psi(T_i) - \theta_i P_i,$$

where μ is a positive constant, $\phi(\theta_i) = \frac{1}{1-\beta}\theta_i^{1-\beta}$ is the KSP's satisfaction function w.r.t. θ_i [45], [46], [47], here $0 \leq \beta \leq 1$, and $\theta_i P_i$ is the actual rewards that the KSP pays to the LKS. The satisfaction function increases with θ_i and the marginal satisfaction diminishes when θ_i becomes larger.

We consider that the KSP is able to learn the probability distribution of LKSs' type $\{\theta_i, i \in \mathcal{N}\}$ from past statistics [42], [48]. Let λ_i denote the prior probability of type θ_i, and $\sum_{i=1}^{N} \lambda_i = 1$. Then, the utility of KSP U_{KSP} can be formulated as

$$U_{KSP} = \sum_{i=1}^{N} \lambda_i \left\{\mu\phi(\theta_i)\psi(T_i) - \theta_i P_i\right\}. \tag{3}$$

We then define the utility function of LKS i who admits the contract $\{T_i, P_i\}$. Consider that knowledge is a virtual commodity with the property of reproducibility, and thus LKS i's profit is composed of two parts: the rewards $\theta_i P_i$ received from the KSP, and the monetary earning $P_{KNO} \cdot \psi(V_i)$ from other markets, where $P_{KNO} > 0$ denotes the unit knowledge earning. As for the cost, LKS i needs to provide rewards R_i to sensors for data collection, as well as bear a linear training cost $\tau V_i, \tau > 0$ [47]. Therefore, the utility of LKS i U_{LKS_i} is given by

$$U_{LKS_i} = \theta_i P_i + P_{KNO}\psi(V_i) - R_i - \tau V_i. \tag{4}$$

We next determine the utility function for sensors. Note that sensors compete for the DC's block rewards R_i based on their contribution to the data collection task, and the contribution is not only proportional to the data size that a sensor can collect, but also positively correlated with the importance of a sensor's location. The more important location of sensor m, i.e., a larger η_i, the greater contribution it makes to the data collection task. Therefore, we use $\eta_m X_{S_m}$ to measure the contribution of sensor m, and its utility U_{S_m} is given by

$$U_{S_m} = \frac{\eta_m X_{S_m}}{\sum_{m=1}^{M_i} \eta_m X_{S_m}} R_i - c X_{S_m}, \tag{5}$$

where $c > 0$ denotes the cost of collecting data per unit size.

4.3 Problem Formulation

Upper layer (knowledge trading process): The KSP determines the optimal contracts $\{\boldsymbol{T}, \boldsymbol{P}\}$ to maximize its utility, which is formulated as the following optimization problem:

$$\max_{\{\boldsymbol{T},\boldsymbol{P}\}} U_{KSP} = \sum_{i=1}^{N} \lambda_i \left\{\mu\phi(\theta_i)\psi(T_i) - \theta_i P_i\right\}.$$

Lower layer (knowledge generation process):

- Stage I. After admitting a contract from the KSP, LKS i acts as the leader of the KG game to decide the optimal block rewards R_i for its utility maximization, which is formulated as the following optimization problem:

$$\max_{R_i} U_{LKS_i} = \theta_i P_i + P_{KNO}\psi(V_i) - R_i - \tau V_i, \text{ for } \forall i \in \mathcal{N}.$$

- Stage II. Given R_i, sensor m strategizes its data size X_{S_m} individually in the DSD game to achieve its maximal utility, which is formulated as the following optimization problem:

$$\max_{X_{S_m}} U_{S_m} = \frac{\eta_m X_{S_m}}{\sum_{m=1}^{M_i} \eta_m X_{S_m}} R_i - c X_{S_m}, \text{ for } \forall m \in \mathcal{M}_i.$$

In what follows, we analyze the nested hierarchical game in a backward inductive way to identify the optimal strategies for the KSP, LKSs, and sensors. That is, we first consider that the contract from the upper layer is given and solve the KG game in the lower layer. Then, we substitute the solution of the lower layer into the upper layer and design the optimal contract. Note that the KG game in the lower layer is a two-stage Stackelberg game, we solve the optimization problems at the two stages also in a backward inductive manner.

5 Stackelberg Game Analysis for Knowledge Generation Process

In this section, we identify the optimal strategies of LKS i and its sensors in the KG game, under the given contract $\{T_i, P_i\}$. To solve the KG game, we need to investigate the following questions.

Q1: For given block rewards R_i, is there a stable strategy profile in the DSD game, such that no sensor can gain more utility by changing its current strategy unilaterally?

Q2: If the answer of Q1 is yes, is the stable strategy profile unique? If unique, then the sensors will certainly choose their strategies in this stable strategy profile.

Q3: How can LKS i select the optimal rewards R_i^* to maximize its utility?

Note that the stable strategy profile in Q1 corresponds to the concept of Nash Equilibrium (NE) in Game Theory [49], which is be defined as follow.

Definition 1 (Nash Equilibrium)**.** A strategy profile $\mathbf{X}_S^{ne} = \{X_{S_1}^{ne}, X_{S_2}^{ne}, \cdots, X_{S_{M_i}}^{ne}\}$ is a Nash Equilibrium of the DSD game, if for any sensor $m \in \mathcal{M}_i$ and any strategy $X_{S_m} \geq 0$ we have

$$U_{S_m}(X_{S_m}^{ne}, \mathbf{X}_{-S_m}^{ne}) \geq U_{S_m}(X_{S_m}, \mathbf{X}_{-S_m}^{ne}).$$

The existence (Q1) and uniqueness (Q2) of NE can enable LKS i to anticipate the actions of its sensors and thus decide the optimal value of R_i (Q3) in a backward inductive way. The unique NE of the DSD game together with the optimal rewards R^* constitute the solution to the KG game.

5.1 Stage II: Collected Data Size Determination

The DSD game is a non-cooperative game, where every sensor competes with each other individually for winning the block rewards R_i. As a notational convention, let $\mathcal{G} = \{\mathcal{M}_i, \{\mathbf{X}_S\}, \{U_{S_m}\}\}$ represent the DSD game, where $\mathcal{M}_i$, $\{\mathbf{X}_S\}$, and $\{U_{S_m}\}$ are the sets of players (i.e., sensors), strategy profiles, and utility functions, respectively. In order to achieve its maximal utility, each player will adopt its best response strategy in the DSD game, which is defined as follows.

Definition 2 (Best Response Strategy)**.** Given $\mathbf{X}_{-S_m}$, a strategy is the best response strategy of sensor m, denoted as $b_m(\mathbf{X}_{-S_m})$, if it satisfies $U_{S_m}(b_m(\mathbf{X}_{-S_m}), \mathbf{X}_{-S_m}) \geq U_{S_m}(X_{S_m}, \mathbf{X}_{-S_m})$ for all $X_{S_m} \geq 0$.

Note that every sensor plays its best response strategy in an NE. Therefore, the strategy profile that all sensors play in the DSD game will converge to an NE, if it exists. Let $\mathbf{b}(\mathbf{X}_S) = \{b_1(\mathbf{X}_{-S_1}), b_2(\mathbf{X}_{-S_2}), \cdots, b_m(\mathbf{X}_{-S_m})\}$ denote the best response correspondence of the strategy profile $\mathbf{X}_S$. Then, an NE is actually a fixed point of the best response correspondence, i.e., $\mathbf{X}_S^{ne} = \mathbf{b}(\mathbf{X}_S)^{ne}$. Before proving the existence and uniqueness of NE in the DSD game, we provide three propositions as follows.

Proposition 1. An NE exists in the DSD game $\mathcal{G} = \{\mathcal{M}_i, \{\mathbf{X}_S\}, \{U_{S_m}\}\}$, if: 1) $\{\mathbf{X}_S\}$ is a nonempty, compact and convex subset of the M_i-dimensional Euclidean space $\mathbf{R}^{M_i}$, and 2) U_{S_m} is concave on X_{S_m}, for $\forall m \in \mathcal{M}_i$ [50, Theorem 1].

Proposition 2. Function $\mathbf{b}(\mathbf{X}_S)$ is *standard* if for any $\mathbf{X}_S > 0$, the following properties are satisfied [51]:

- *Positivity*: $\mathbf{b}(\mathbf{X}_S) > 0$.
- *Monotonicity*: if $\mathbf{X}_S \geq \mathbf{X}_S'$, then $\mathbf{b}(\mathbf{X}_S) \geq \mathbf{b}(\mathbf{X}_S')$.
- *Scalability*: if $\delta > 1$, then $\delta\mathbf{b}(\mathbf{X}_S) > \mathbf{b}(\delta\mathbf{X}_S)$.

Proposition 3. If function $\mathbf{b}(\mathbf{X}_S)$ is standard, then its fixed point is unique [52, Theorem 1].

Then, we have the following theorems.

Theorem 1. The DSD game has a unique NE.

Proof: We prove the existence and uniqueness of NE sequentially.

1) Existence: Since the strategy of sensor m is $X_{S_m} \geq 0$, the strategy space of the DSD game $\{\mathbf{X}_S\}$ is a nonempty, compact, and convex subset of the M_i-dimensional Euclidean space $\mathbf{R}^{M_i}$. Taking the first- and second-order derivatives of U_{S_m} w.r.t. X_{S_m}, which yields

$$\frac{\partial U_{S_m}}{\partial X_{S_m}} = \frac{\eta_m R_i}{\sum_{l\in\mathcal{M}_i}\eta_l X_{S_l}} - \frac{{\eta_m}^2 R_i X_{S_m}}{(\sum_{l\in\mathcal{M}_i}\eta_l X_{S_l})^2} - c, \tag{6}$$

$$\frac{\partial^2 U_{S_m}}{\partial X_{S_m}^2} = \frac{-2{\eta_m}^2 R_i \sum_{l\neq m}\eta_l X_{S_l}}{(\sum_{l\in\mathcal{M}_i}\eta_l X_{S_l})^3} < 0. \tag{7}$$

We can see that U_{S_m} is continuous and differentiable on X_{S_m}, and the second-order derivative of U_{S_m} w.r.t. X_{S_m} is negative. Thus, U_{S_m} is a concave function of X_{S_m}. In the light of Proposition 1, there exists at least an NE in the DSD game.

2) Uniqueness: Due to the concavity of U_{S_m}, we can determine the best response strategy $b_m(\mathbf{X}_{-S_m})$ by setting the first-order derivative of U_{S_m} w.r.t. X_{S_m} to be 0, that is

$$\frac{\eta_m R_i}{\sum_{S_l\in\mathcal{M}_i}\eta_l X_{S_l}} - \frac{{\eta_m}^2 R_i X_{S_m}}{(\sum_{S_l\in\mathcal{M}_i}\eta_l X_{S_l})^2} - c = 0, \tag{8}$$

$$\Rightarrow X_{S_m} = \sqrt{\frac{R_i \sum_{l\in\mathcal{M}_i\setminus\{m\}}\eta_l X_{S_l}}{c\eta_m}} - \frac{1}{\eta_m}\sum_{l\in\mathcal{M}_i\setminus\{m\}}\eta_l X_{S_l}. \tag{9}$$

If the right-hand side of (9) is positive, it is the best response strategy of sensor m; if not, it indicates that sensor m will not participate in the DSD game and thus $X_{S_m} = 0$. Therefore, the best response strategy of sensor m is given by Eq. (10), shown at the bottom of this page. From (10) we can see that the best response function of any sensor who joins the DSD game is always positive and monotonic. Regarding the scalability, we have

$$\delta b_m(\mathbf{X}_{-S_m}) - b_m(\delta\mathbf{X}_{-S_m}) = (\delta - \sqrt{\delta})\sqrt{\frac{R_i \sum_{l\in\mathcal{M}_i\setminus\{m\}}\eta_l X_{S_l}}{c\eta_m}}. \tag{11}$$

For $\forall\delta > 1$ there is $\delta - \sqrt{\delta} > 0$. Thus, (11) is positive and $\delta b_m(\mathbf{X}_{-S_m}) > b_m(\delta\mathbf{X}_{-S_m})$ holds. In the light of Proposition 2 and Proposition 3, $\mathbf{b}(\mathbf{X}_S)$ is a standard function whose fixed point is unique, indicating that the NE in the DSD game is unique. □

$$b_m(\mathbf{X}_{-S_m}) = \begin{cases} 0, & R_i \leq \frac{c}{\eta_m}\sum_{l\in\mathcal{M}_i\setminus\{m\}}\eta_l X_{S_l} \\ \sqrt{\frac{R_i \sum_{l\in\mathcal{M}_i\setminus\{m\}}\eta_l X_{S_l}}{c\eta_m}} - \frac{1}{\eta_m}\sum_{l\in\mathcal{M}_i\setminus\{m\}}\eta_l X_{S_l}, & \text{otherwise} \end{cases} \tag{10}$$

Theorem 2. The unique NE of the DSD game is given by

$$X_{S_m}^{ne} = R_i \cdot \xi_m, \quad \text{for } \forall m \in \mathcal{M}_i \tag{12}$$

where

$$\xi_m = \left[\frac{1}{c} \left(M_i - 1\right) \left(\sum_{l=1}^{M_i} \frac{\eta_m}{\eta_l} - M_i + 1 \right) \left(\sum_{l=1}^{M_i} \frac{\eta_m}{\eta_l} \right)^{-2} \right]^+, \tag{13}$$

and $[z]^+$ denotes $\max\{z, 0\}$.

Proof: Note that a sensor cannot compute its best response strategy based on formula (10) directly since it is coupled with the strategies of all the other sensors. Let $E_m = \sum_{l \in \mathcal{M}_i \setminus \{m\}} \eta_l X_{S_l}^{ne}$. Substituting the NE $\{\mathbf{X}_S^{ne}\}$ into Eq. (8) and performing some algebraic transformations yield

$$\begin{aligned} \eta_m E_m &= \frac{c}{R_i} (E_m + \eta_m X_{S_m}^{ne})^2 = \frac{c}{R_i} \left(\sum_{k \in \mathcal{M}_i} \eta_k X_{S_k}^{ne} \right)^2 \\ &= \frac{c}{R_i} (E_l + \eta_l X_{S_l}^{ne})^2 = \eta_l E_l, \quad \text{for } \forall m, l \in \mathcal{M}_i. \end{aligned} \tag{14}$$

Then, we have $E_l = \frac{\eta_m}{\eta_l} E_m$ for $\forall l \in \mathcal{M}_i$ and obtain the following system of equations:

$$\begin{cases} E_1 = \frac{\eta_m}{\eta_1} E_m \\ E_2 = \frac{\eta_m}{\eta_2} E_m \\ \vdots \\ E_{M_i} = \frac{\eta_m}{\eta_{M_i}} E_m. \end{cases} \tag{15}$$

Making the summation on both sides of (15) yields

$$\sum_{l=1}^{\mathcal{M}_i} E_l = \sum_{l=1}^{M_i} \frac{\eta_m}{\eta_l} E_m. \tag{16}$$

Since $E_l = \sum_{k \in \mathcal{M}_i \setminus \{l\}} \eta_k X_{S_k}^{ne} = E_m + \eta_m X_{S_m}^{ne} - \eta_l X_{S_l}^{ne}$, there is

$$\begin{aligned} \sum_{l=1}^{M_i} E_l &= M_i(E_m + \eta_m X_{S_m}^{ne}) - \sum_{l=1}^{M_i} \eta_l X_{S_l}^{ne} \\ &= M_i(E_m + \eta_m X_{S_m}^{ne}) - (E_m + \eta_m X_{S_m}^{ne}) \\ &= (M_i - 1)(E_m + \eta_m X_{S_m}^{ne}) = \sum_{l=1}^{M_i} \frac{\eta_m}{\eta_l} E_m, \end{aligned} \tag{17}$$

and thus

$$E_m = \frac{(M_i - 1)\eta_m X_{S_m}^{ne}}{\sum\limits_{l=1}^{M_i} \frac{\eta_m}{\eta_l} - M_i + 1}. \tag{18}$$

Substituting (18) into (9) and performing some algebraic operations, we have

$$X_{S_m}^{ne} = \frac{R_i (M_i - 1) \left(\sum\limits_{l=1}^{M_i} \frac{\eta_m}{\eta_l} - M_i + 1 \right)}{c \cdot \left(\sum\limits_{l=1}^{M_i} \frac{\eta_m}{\eta_l} \right)^2} = R_i \cdot \xi_m.$$

□

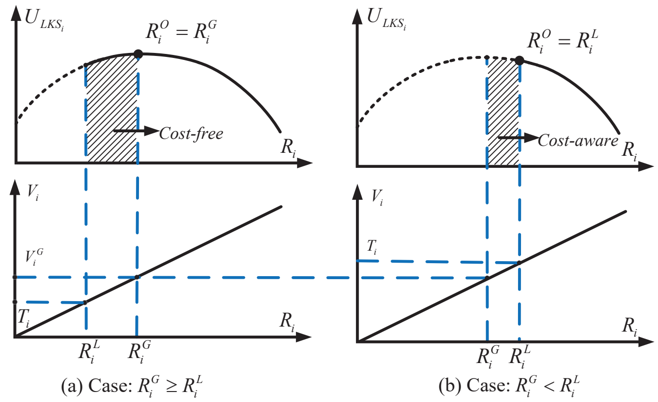


Fig. 5. Illustration for the optimal strategy of LKS i.

5.2 Stage I: Block Rewards Determination

Based on the above analysis, LKS i can calculate the dataset value under any given block rewards by substituting formula (12) into (1), that is

$$V_i(R_i) = \frac{R_i \sum_{m=1}^{M_i} \eta_m \xi_m}{1 + e^{-M_i}}. \tag{19}$$

Note that if contract $\{T_i, P_i\}$ has been admitted, the dataset value acquired by LKS i should be no less than that required by the contract, i.e., $V_i \geq T_i$, and thus the following condition needs to be satisfied:

$$R_i \geq \frac{(1 + e^{-M_i})T_i}{\sum_{m=1}^{M_i} \eta_m \xi_m} \triangleq R_i^L, \tag{20}$$

where R_i^L denotes the lower bound of block rewards required for completing the contract and it depends on T_i designed by the KSP, i.e., $V_i(R_i^L) = T_i$. Substituting (2) and (19) into (4), the utility maximization problem of LKS i can be re-formulated as:

$$\begin{aligned} \max_{R_i} U_{LKS_i} = \theta_i P_i + \alpha P_{KNO} \ln(1 + \omega \frac{R_i \sum_{m=1}^{M_i} \eta_m \xi_m}{1 + e^{-M_i}}) \\ - R_i - \tau \frac{R_i \sum_{m=1}^{M_i} \eta_m \xi_m}{1 + e^{-M_i}}, \end{aligned} \tag{21a}$$

$$\text{s.t.} \quad R_i \geq R_i^L. \tag{21b}$$

It can be easily checked from (21a) that U_{LKS_i} is concave on R_i. Let R_i^G denote the globally optimal solution to (21a) without constraint (21b), which can be determined as

$$R_i^G = \frac{\alpha P_{KNO}(1 + e^{-M_i})}{\tau \sum_{m=1}^{M_i} \eta_m \xi_m + 1 + e^{-M_i}} - \frac{1 + e^{-M_i}}{\omega \sum_{m=1}^{M_i} \eta_m \xi_m}.$$

Then, the optimal solution R_i^O to (21) is given by

$$R_i^O = \max\left\{R_i^L, R_i^G\right\}.$$

We show in Fig. 5 the curves of V_i and U_{LKS_i} varying with R_i. For the case $R_i^G \geq R_i^L$ in Fig. 5 (a), we have $R_i^O = R_i^G$ and $V_i(R_i^G) \triangleq V_i^G > V_i(R_i^L) = T_i$, i.e., the optimized dataset value of LKS i exceeds the dataset value stipulated by the contract. It indicates that when the KSP designs the contract, it can request a larger T_i (accordingly, R_i^L increases) without causing more cost for LKS i until R_i^L increases to R_i^G, as shown in the shadow region *Cost-free*. For the case $R_i^L > R_i^G$ in Fig. 5 (b), we have $R_i^O = R_i^L$, and LKS i will bear more cost with any increment in R_i^L, as shown in the shadow region *Cost-aware*. Based on the above analysis, the KSP will always set $T_i \geq V_i^G$ so that the *Cost-free* region is not "wasted". As a result, $R_i^L \geq R_i^G$ will hold for any rational contract $\{T_i, P_i\}$. Let R_i^* denote the optimal block rewards paid by LKS i to its sensors, then we have

$$R_i^* = R_i^L = \frac{(1+e^{-M_i})T_i}{\sum_{m=1}^{M_i} \eta_m \xi_m}. \tag{22}$$

6 Contract Analysis for Knowledge Trading Process

Given a contract, we have identified the optimal strategies of an LKS and sensors in the knowledge generation process. Based on the obtained results, in this section, we focus on the upper layer of the nested hierarchical game to reveal how the KSP designs the optimal contract in the knowledge trading process.

To facilitate the following analysis, we rewrite the optimal utility of LKS i by substituting $R_i^* = R_i^L$ into (21a), which yields

$$\begin{aligned} U_{LKS_i}^* &= \theta_i P_i + \alpha P_{KNO} \ln(1+\omega T_i) - \tau T_i - \frac{(1+e^{-M_i})T_i}{\sum_{m=1}^{M_i} \eta_m \xi_m} \\ &= \theta_i P_i - (\Gamma_i T_i - P_{KNO}\psi(T_i)) \\ &\triangleq \theta_i P_i - \Upsilon_{i,i}, \end{aligned} \tag{23}$$

where $\Gamma_i = \frac{(\tau \sum_{m=1}^{M_i} \eta_m \xi_m + 1 + e^{-M_i})}{\sum_{m=1}^{M_i} \eta_m \xi_m}$ and it can be regarded as an auxiliary parameter that integrates the parameters related to LKS i except for T_i. Since $\{\theta_i\}$ has been sorted in ascending order, and a higher type LKS means that bigger datasets are collected, leading to a larger $\sum_{m=1}^{M_i} \eta_m \xi_m$ and further a smaller Γ_i. Then, we have

$$\Gamma_1 \geq \Gamma_2 \geq \cdots \Gamma_N. \tag{24}$$

Regarding $\Upsilon_{i,i}$, its first and second subscripts denote the index related to LKS i (i.e., Γ_i) and the index related to the accepted contract (i.e., T_i), respectively.

As mentioned previously, to address the problems of information asymmetry and incentive cost-saving, the KSP needs to design differentiated contracts for different types of LKSs. According to the Contract Theory [43], the KSP must seek its utility maximization under the following constraints when setting up a series of feasible contracts:

(1) *Individual Rationality (IR)*: IR indicates that LKS i could gain a non-negative payoff by accepting the contract. Without loss of generality, we assume that an LKS will voluntarily accept the contract if it is not worse off by doing so, and the KSP can always inspire the LKS by offering an additional small amount of benefit [53]. Then, based on (23), IR can be expressed as

$$\theta_i P_i - \Upsilon_{i,i} \geq \widehat{U}_i^{NC}, \text{ for } \forall i \in \mathcal{N},$$

where $\widehat{U}_i^{NC}$ is LKS i's utility when it refuses $\{T_i, P_i\}$.

(2) *Incentive Compatibility (IC)*: IC indicates that each type-i LKS will achieve utility maximization if it chooses its own contract $\{T_i, P_i\}$ instead of any others [54], [55]. i.e.,

$$\theta_i P_i - \Upsilon_{i,i} \geq \theta_i P_j - \Upsilon_{i,j}, \text{ for } \forall i, j \in \mathcal{N}, i \neq j.$$

Therefore, the KSP's utility maximization is reformulated as problem **P**1.

$$\mathbf{P}1: \max_{\{\boldsymbol{T},\boldsymbol{P}\}} U_{KSP} = \sum_{i=1}^{N} \lambda_i \left\{ \frac{\mu\alpha}{1-\beta} \theta_i^{1-\beta} \ln(1+\omega T_i) - \theta_i P_i \right\} \tag{25a}$$

$$\begin{aligned} \text{s.t.} \quad & \theta_i P_i - \Upsilon_{i,i} \geq \widehat{U}_i^{NC}, \ \forall i \in \mathcal{N}, & \text{(25b)} \\ & \theta_i P_i - \Upsilon_{i,i} \geq \theta_i P_j - \Upsilon_{i,j}, \ \forall i, j \in \mathcal{N}, i \neq j, & \text{(25c)} \\ & \theta_1 \leq \theta_2 \leq \cdots \leq \theta_N, & \text{(25d)} \\ & T_i \geq V_i^G, \ \forall i \in \mathcal{N}, & \text{(25e)} \\ & P_i \geq 0, \ \forall i \in \mathcal{N}. & \text{(25f)} \end{aligned}$$

Note that **P**1 is a complicated non-convex optimization problem with $\frac{N(N-1)}{2} + 3N + 1$ constraints. To tackle this challenging problem, we provide the following lemmas to reduce the complexity caused by the IR and IC constraints.

Lemma 1. For any contracts $\{T_i, P_i\}$ and $\{T_j, P_j\}$, $i, j \in \mathcal{N}$ and $i \neq j$, we have $P_i \geq P_j$ and $T_i \geq T_j$ if and only if $\theta_i \geq \theta_j$.

Proof: According to the IC constraints, we have

$$\theta_i P_i - \Upsilon_{i,i} \geq \theta_i P_j - \Upsilon_{i,j},$$
$$\theta_j P_j - \Upsilon_{j,j} \geq \theta_j P_i - \Upsilon_{j,i},$$

which can be rewritten as

$$\theta_i(P_i - P_j) \geq \Upsilon_{i,i} - \Upsilon_{i,j}, \tag{26}$$
$$\theta_j(P_j - P_i) \geq \Upsilon_{j,j} - \Upsilon_{j,i}. \tag{27}$$

Obviously, $\Upsilon_{i,i}$ is a convex function of T_i and monotonically increases with T_i when $T_i \geq V_i^G$. If $T_i \geq T_j$, there is $\Upsilon_{i,i} \geq \Upsilon_{i,j}$, and then from (26) we have $P_i \geq P_j$. If $P_i \geq P_j$, then from (27) we have $\Upsilon_{j,j} \leq \Upsilon_{j,i}$ and thus $T_i \geq T_j$. Therefore, $T_i \geq T_j$ is equivalent to $P_i \geq P_j$.

Adding both sides of (26) and (27) respectively, the following inequality holds:

$$(\theta_i - \theta_j)(P_i - P_j) \geq (\Gamma_i - \Gamma_j)(T_i - T_j). \tag{28}$$

If $\theta_i \geq \theta_j$, there is $\Gamma_i \leq \Gamma_j$, and then from (28) we have $P_i \geq P_j$ and $T_i \geq T_j$. If $P_i \geq P_j$ and $T_i \geq T_j$, since $(\theta_i - \theta_j)(\Gamma_i - \Gamma_j) \leq 0$, from (28) it can be derived that $\theta_i \geq \theta_j$. Therefore, we have $\theta_i \geq \theta_j \Leftrightarrow P_i \geq P_j \Leftrightarrow T_i \geq T_j$. This completes the proof. □

Lemma 2. Once the IR constraint of type-1 LKS is satisfied, the IR constraints of other type LKSs will also hold.

Proof: According to the IC constraints, for $\forall i \in \{2, \cdots, N\}$, there is

$$\theta_i P_i - \Upsilon_{i,i} \geq \theta_i P_1 - \Upsilon_{i,1}. \tag{29}$$

Since $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_N$ and $\Gamma_1 \geq \Gamma_2 \geq \cdots \geq \Gamma_N$, we have

$$\begin{aligned} &(\theta_i P_1 - \Upsilon_{i,1}) - (\theta_1 P_1 - \Upsilon_{1,1}) \\ =&(\theta_i - \theta_1)P_1 + (\Gamma_1 - \Gamma_i)T_1 \geq 0. \end{aligned} \tag{30}$$

From (29) and (30), we can obtain

$$\theta_i P_i - \Upsilon_{i,i} \geq \theta_1 P_1 - \Upsilon_{1,1}. \tag{31}$$

It indicates that the utility of type-1 LKS is the lower bound for that of all LKSs. Thus, once the IR constraint of type-1 LKS is satisfied, all the IR constraints will hold. □

Lemma 3. The IC constraints can be reduced to the local downward incentive compatibility (LDIC):

$$\theta_i P_i - \Upsilon_{i,i} \geq \theta_i P_{i-1} - \Upsilon_{i,i-1},\ \forall i \in \{2, \cdots, N\}, \tag{32}$$

and the local upward incentive compatibility (LUIC):

$$\theta_i P_i - \Upsilon_{i,i} \geq \theta_i P_{i+1} - \Upsilon_{i,i+1},\ \forall i \in \{1, \cdots, N-1\}. \tag{33}$$

Proof: We first check the LDIC. Consider three continuous types of LKSs, i.e., $\theta_{i-1} \leq \theta_i \leq \theta_{i+1}, i \in \{2, \cdots, N-1\}$, given the following LDIC holds:

$$\theta_{i+1} P_{i+1} - \Upsilon_{i+1,i+1} \geq \theta_{i+1} P_i - \Upsilon_{i+1,i}, \tag{34}$$

$$\theta_i P_i - \Upsilon_{i,i} \geq \theta_i P_{i-1} - \Upsilon_{i,i-1}. \tag{35}$$

From Lemma 1 there is

$$\begin{aligned} &(\theta_{i+1} - \theta_i)(P_i - P_{i-1}) \geq 0 \\ \Rightarrow &\theta_{i+1}(P_i - P_{i-1}) \geq \theta_i (P_i - P_{i-1}). \end{aligned} \tag{36}$$

Combining (35) and (36) yields

$$\theta_{i+1}(P_i - P_{i-1}) \geq \Upsilon_{i,i} - \Upsilon_{i,i-1}. \tag{37}$$

In the light of Lemma 1, we have

$$\begin{aligned} &\Gamma_i(T_i - T_{i-1}) \geq \Gamma_{i+1}(T_i - T_{i-1}), \\ \Rightarrow &\Gamma_i T_i - \Gamma_i T_{i-1} + \Gamma_{i+1} T_{i-1} - \Gamma_{i+1} T_i \geq 0, \\ \Rightarrow &-\alpha P_{KNO}\psi(T_i) + \Gamma_i T_i + \alpha P_{KNO}\psi(T_{i-1}) \\ &- \Gamma_i T_{i-1} - \alpha P_{KNO}\psi(T_{i-1}) + \Gamma_{i+1} T_{i-1} \\ &+ \alpha P_{KNO}\psi(T_i) - \Gamma_{i+1} T_i \geq 0 \\ \Rightarrow &\Upsilon_{i,i} - \Upsilon_{i,i-1} \geq \Upsilon_{i+1,i} - \Upsilon_{i+1,i-1}. \end{aligned} \tag{38}$$

Combining (37) and (38) yields

$$\theta_{i+1} P_i - \Upsilon_{i+1,i} \geq \theta_{i+1} P_{i-1} - \Upsilon_{i+1,i-1}. \tag{39}$$

From (34) and (39) there is

$$\theta_{i+1} P_{i+1} - \Upsilon_{i+1,i+1} \geq \theta_{i+1} P_{i-1} - \Upsilon_{i+1,i-1}. \tag{40}$$

Then, we can extend (40) to prove that the LDIC holds until type-1:

$$\begin{aligned} \theta_{i+1} P_{i+1} - \Upsilon_{i+1,i+1} &\geq \theta_{i+1} P_i - \Upsilon_{i+1,i} \\ &\geq \cdots \geq \theta_{i+1} P_1 - \Upsilon_{i+1,1}. \end{aligned} \tag{41}$$

In a similar way, we can derive the LUIC. This completes the proof. □

By utilizing Lemma 1-3, we can transform problem **P**1 into problem **P**2 as follows.

$$\mathbf{P}2: \max_{\{\boldsymbol{T},\boldsymbol{P}\}} U_{KSP} = \sum_{i=1}^{N} \lambda_i \left\{ \frac{\mu\alpha}{1-\beta} \theta_i^{1-\beta} \ln(1+\omega T_i) - \theta_i P_i \right\} \tag{42a}$$

$$\text{s.t.} \quad \theta_1 P_1 - \Upsilon_{1,1} = \widehat{U}_1^{NC}, \tag{42b}$$

$$\theta_i P_i - \Upsilon_{i,i} = \theta_i P_{i-1} - \Upsilon_{i,i-1},\ \forall i \in \{2, \cdots, N\}, \tag{42c}$$

$$\theta_1 \leq \theta_2 \leq \cdots \leq \theta_N, \tag{42d}$$

$$T_i \geq V_i^G,\ \forall i \in \mathcal{N}, \tag{42e}$$

$$P_i \geq 0,\ \forall i \in \mathcal{N}. \tag{42f}$$

By calculating the sum of all the IC constraints, we can obtain

$$P_i = \frac{\widehat{U}_1^{NC} + \Upsilon_{1,1}}{\theta_1} + \sum_{k=2}^{i} \frac{\Upsilon_{k,k} - \Upsilon_{k,k-1}}{\theta_k}. \tag{43}$$

Substituting (43) into **P**2, then **P**2 can be transformed into **P**3 as follows.

$$\begin{aligned} \mathbf{P}3: \max_{\{T_i\}} U_{KSP} = \sum_{i=1}^{N} \lambda_i \Bigg\{ &\frac{\mu\alpha}{1-\beta} \theta_i^{1-\beta} \ln(1+\omega T_i) - \\ &\frac{\theta_i(\widehat{U}_1^{NC} + \Upsilon_{1,1})}{\theta_1} + \sum_{k=2}^{i-1} \frac{\Upsilon_{k,k} - \Upsilon_{k,k-1}}{\theta_k} - \Upsilon_{i,i} + \Upsilon_{i,i-1} \Bigg\} \end{aligned} \tag{44a}$$

$$\text{s.t.} \quad \theta_1 \leq \theta_2 \leq \cdots \leq \theta_N, \tag{44b}$$

$$T_i \geq V_i^G,\ \forall i \in \mathcal{N}. \tag{44c}$$

Taking the first- and second-order derivatives of U_{KSP} w.r.t. T_i, we have

$$\begin{aligned} \frac{\partial U_{KSP}}{\partial T_i} &= \lambda_i \Big(-\Gamma_i + \frac{\alpha\omega P_{KNO}}{1+\omega T_i} + \frac{\frac{\omega\mu\alpha}{1-\beta}\theta_i^{1-\beta}}{1+\omega T_i}\Big), \\ \frac{\partial^2 U_{KSP}}{\partial T_i^2} &= -\frac{\lambda_i \alpha \omega^2 (P_{KNO} + \frac{\mu}{1-\beta}\theta_i^{1-\beta})}{(1+\omega T_i)^2} < 0. \end{aligned}$$

Therefore, U_{KSP} is concave on T_i and **P**3 is a convex optimization problem. By solving the equation $\frac{\partial U_{KSP}}{\partial T_i} = 0$, the optimal task value T_i^* of the contract for type-i LKS is determined as

$$T_i^* = \max\left\{ \frac{\frac{\mu\alpha}{1-\beta}\theta_i^{1-\beta} + \alpha P_{KNO}}{\Gamma_i} - \frac{1}{\omega}, V_i^G \right\}. \tag{45}$$

Accordingly, the optimal price P_i^* of the contract can be obtained from (43).

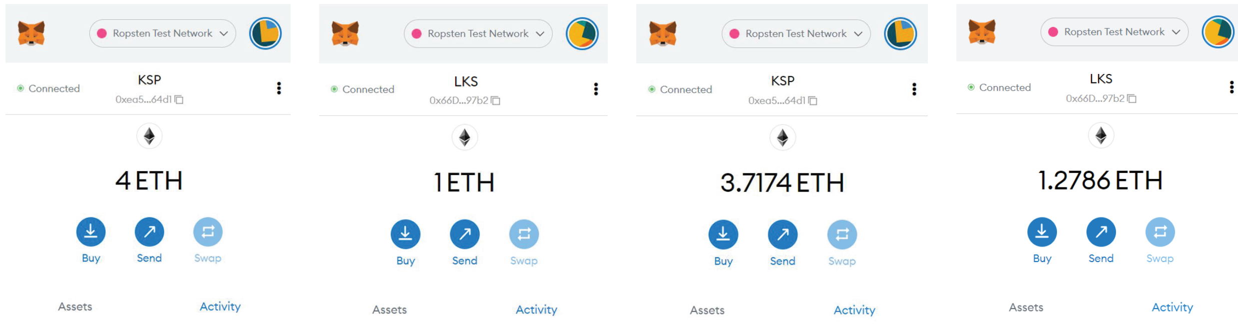
7 Simulation Results

In this section, we consider a BWKA ecosystem with one KSP and 10 LKSs, and each LKS governs several sensors. We implement the SKTC-based knowledge trading on Sepolia Test Network and present simulation results to demonstrate the performance behaviors of the BWKA ecosystem. The detailed simulation settings are given in Table 2.

7.1 Implementation of SKTC-based Knowledge Trading

Environment. We implement the smart knowledge trade contract with the Solidity programming language of Ethereum under the compiler version $0.4.26+commit.4563c3fc$, and deploy it on the Sepolia Test Network by MetaMask, which can be found at [56]. The IPFS version is $0.4.14$. Without loss of generality, we only presents the interactions of the KSP and one LKS (a type-9 LKS) in the implementation. Based on formulas (45) and (43), the KSP can calculate that $T_9 = 7.01$, $P_9 = 31$. Due to the preciousness of Ether, we set the unit of P to be 0.01 ETH, i.e., 10 Finney (1 ETH =1000 Finney).

1) *System Initialization:* The KSP's Ethereum address is $0xea55434eDfC06dD8A41A7DdadcFF5f52237a64d1$, and the initial balance of this account is 4 Ethers. The LKS's Ethereum address is $0x66DB9Eda3B1206e8377bf4faB496599f9b3f97b2$, and the initial balance is 1 Ether. As shown in Fig. 6 (a).



(a) (b)

Fig. 6. Account balance of KSP and LKS: (a) Initial balance; (b) Balance after knowledge trading.

TABLE 3
Smart Contract Cost and Balance of Entities.

Steps	Gas cost (Original value)	Total gas cost (ETH)	Total gas cost (USD)	Balance of KSP (ETH)		Balance of LKS 9 (ETH)		Balance of SKTC 9 (ETH)
				Theoretical	Actual	Theoretical	Actual	
Initial balance				4		1		0
Trigger *CreateCont*	1400694	2.8×10^{-3}	7.812	3.590	3.587	1	1	0.41
Trigger *SelCont*	90771	1.8×10^{-4}	0.502	3.590	3.587	0.9	0.89	0.51
Trigger *SendMsg*	211219	4.2×10^{-4}	1.171	3.590	3.587	0.9	0.89	0.51
Trigger *GetMsg*	0	0	0	3.590	3.587	0.9	0.89	0.51
Trigger *ConfTrade*	60080	1.2×10^{-4}	0.335	3.721	3.718	1.279	1.278	0
Ether transfer				-0.279	-0.282	+0.279	+0.278	0

TABLE 2
Simulation Settings.

Types of LKSs	**Settings**
$\theta_1 = 0.1$	$\eta_1 = 2, \eta_2 = 3, \eta_3 = 4$
$\theta_2 = 0.2$	$\eta_1 = 1, \eta_2 = 3, \eta_3 = 3, \eta_4 = 4$
$\theta_3 = 0.3$	$\eta_1 = 3, \eta_2 = 4, \eta_3 = 9$
$\theta_4 = 0.4$	$\eta_1 = 4, \eta_2 = 5, \eta_3 = 6$
$\theta_5 = 0.5$	$\eta_1 = 5, \eta_2 = 6, \eta_3 = 7$
$\theta_6 = 0.6$	$\eta_1 = 6, \eta_2 = 7, \eta_3 = 8$
$\theta_7 = 0.7$	$\eta_1 = 6, \eta_2 = 6, \eta_3 = 8, \eta_4 = 8$
$\theta_8 = 0.8$	$\eta_1 = 8, \eta_2 = 9, \eta_3 = 10$
$\theta_9 = 0.9$	$\eta_1 = 9, \eta_2 = 10, \eta_3 = 11$
$\theta_{10} = 1.0$	$\eta_1 = 11, \eta_2 = 12, \eta_3 = 13$
Pre-defined parameters	$\mu = 10, \omega = 3, c = 0.5$ $P_{KNO} = 2, \tau = 2, \beta = 0.3, \alpha = 1$

2) *Deploy Smart Contracts:* The KSP calls *CreateCont* function to deploy SKTC 9. The KSP sets $T_9 = 7.01$, and $P_9 = 0.31\,\text{ETH}$, $\boldsymbol{\theta} = \{0.1, 0.2, \cdots, 1.0\}$, $t_{dur} = 1000\,\text{sec}$, $MD_{KSP} = MD_{LKS} = 0.1\,\text{ETH}$. Therefore, the KSP needs to attach $msg.value = 0.41\,\text{ETH}$ in SKTC 9.

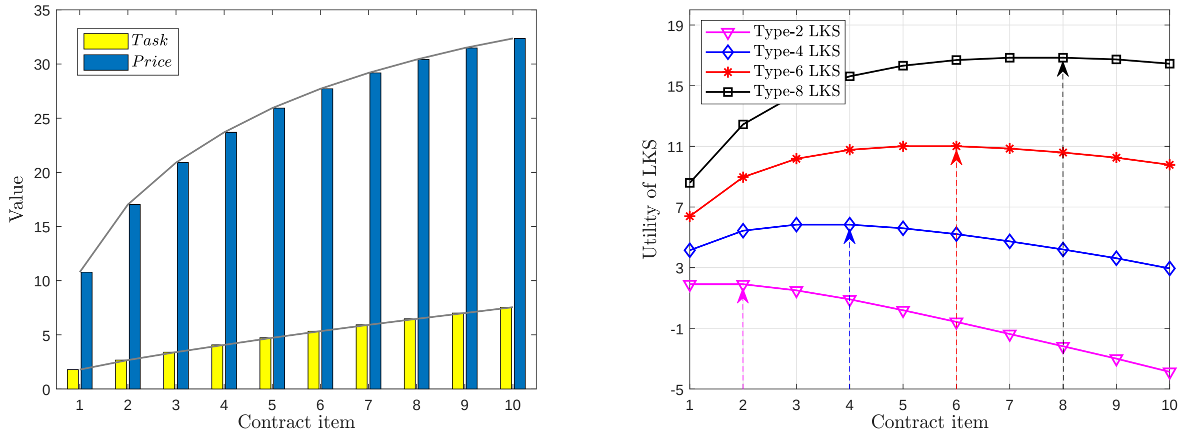
3) *Select Optimal Contract:* LKS 9 triggers *SelCont* function with 0.1 Ethers attached to select SKTC 9. Meanwhile, the task start time is recorded as $t_s = 1645860918$.

4) *Send Knowledge Message:* After accomplishing the task according to SKTC 9, LKS 9 encrypts the knowledge $FILE_{kno}$ by using KSP's public key K_{pub} and uploads the encrypted knowledge $Enc(K_{pub}, FILE_{kno})$ to IPFS. Then, LKS 9 transmits the knowledge link URL_{kno} =*https://ipfs.io/ipfs/KnowledgeHashCode* to SKTC 9 through *SendMsg* function. The task end time is recorded as $t_e = 1645861057$ and the actual type of LKS 9 is calculated as $\theta_{act} = \theta_{N - \lceil N(t_e - t_s) t_{dur}^{-1} \rceil + 1} = \theta_9 = 0.9$.

5) *Get Knowledge Message:* The KSP calls *GetMsg* function to obtain URL_{kno} =*https://ipfs.io/ipfs/KnowledgeHashCode*. Then, the KSP downloads the encrypted knowledge $Enc(K_{pub}, FILE_{kno})$ from URL_{kno} and decrypts the knowledge by using its private key K_{pri}, i.e., $FILE_{kno} = Dec(K_{pri}, Enc(K_{pub}, FILE_{kno}))$.

6) *Confirmation:* After obtaining the knowledge, the KSP triggers *ConfTrade* function to confirm that the knowledge trading process is completed successfully. *ConfTrade* function sends $(1.0 - 0.9)0.31 + 0.1 = 0.131$ Ethers and $0.9 \times 0.31 + 0.1 = 0.379$ Ethers to the KSP and LKS 9, respectively. The balance of the KSP and LKS 9 after knowledge trading is shown in Fig. 6 (b).

The gas cost of functions and the balance of entities during each step in the implementation are summarized in Table 3, where we suppose $1\,\text{Gas} = 2\,\text{Gwei}$, i.e., $1\,\text{Gas} = 2 \times 10^{-9}\,\text{ETH}$ [57], and the gas cost is converted into dollars by using the exchange rate of $1\,\text{ETH} \approx 2790\,\text{USD}$ in February 2022. We can observe that the cost for implementing most of these functions is lower than $1\,\text{USD}$. *GetMsg* function has no cost, because it is a pure *call* function that does not change the state of the blockchain and thus does not consume gas. Although *CreateCont* function causes a relatively high cost, it is invoked only once by the KSP. Due to the existence of gas cost, the actual balance of the KSP and LKS 9 is less than the corresponding theoretical balance, respectively.



(a) $\{Task, Price\}$ of different contracts.

(b) Utility of LKS with different contracts.

Fig. 8. Task-price pairs of contracts and utilities of LKSs under different contracts.

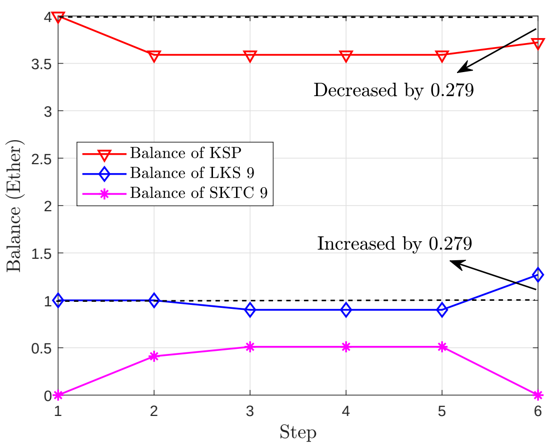


Fig. 7. Balance of KSP, LKS, and smart contract.

We plot Fig. 7 to show the theoretical balance changes of the KSP, LKS 9, and SKTC 9 on each implementation step. We can see that through the SKTC-based knowledge trading, the balance of the KSP is decreased by 0.279 Ethers, the balance of LKS 9 is increased by 0.279 Ethers, and the effect of the smart knowledge trade contract is to manage assets temporarily.

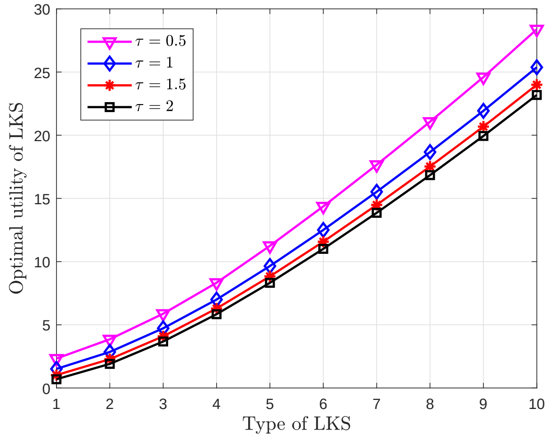
7.2 Performance Evaluation

We summarize in Fig. 8 the task-price pairs of contracts designed by the KSP and the utilities of LKSs under different contracts. Fig. 8(a) shows that both the task value and price value of a contract increases with the LKS's type θ_i. It indicates the optimal strategy of KSP is to allocate more tasks to an LKS with higher operational effectiveness and provide it with more rewards, which corresponds to the monotonicity property of contracts proved in Lemma 1. A very important observation from Fig. 8(b) is that only when an LKS selects the contract designed for its own type can it achieve the maximum utility. For example, the utility of type-4 LKS achieves the peak value when it selects the contract $\{T_4, P_4\}$. It indicates that the *Individual Rationality* (IR) constraint and *Incentive Compatibility* (IC) constraint are satisfied for our SKTC-based knowledge trading scheme. Another careful observation from Fig. 8(b) is that the optimal utility of an LKS becomes larger with the increase of LKS's type, which indicates that the cost of an LKS for completing more knowledge generation tasks can be well compensated with much higher rewards.

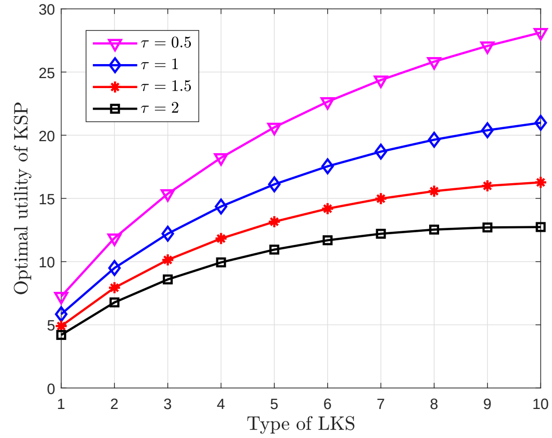
We present in Fig. 9 the utility behaviors of LKSs and KSP under different knowledge training unit cost τ, where we set $\tau \in \{0.5, 1, 1.5, 2\}$. Fig. 9 shows that a higher cost can lead to a decrease in the utilities of all LKSs and the KSP. However, by comparing Fig. 9(a) with Fig. 9(b) carefully, we can observe that the utility of KSP drops much more significantly than that of LKS, i.e., the KSP is more sensitive to changes in τ. This is because that when LKSs suffer from a high knowledge training cost, the KSP have to provide LKSs with more monetary incentives to compensate for their cost. In other words, it is the KSP rather than LKSs, that takes more risk for the increase in training cost. Another interesting observation is that the utility increment of KSP from hiring a higher type LKS gradually diminishes as the training cost τ grows. It indicates the benefits brought by an LKS with high operational effectiveness are offset by the overpaid monetary compensation.

In Fig. 10, we take type-2 LKS and sensors within its sensing area as an example to show the strategic interactions among them. We can observe that each sensor's utility U_{S_m} increases monotonously with type-2 LKS's rewards R_2. This is because the growth of R_2 can raise the expected profit of sensors in its sensing area. A sensor with a larger position weight parameter η can obtain a higher utility because the more important sensor's location is, the greater contributions it makes to the knowledge generation process, and thus the bigger benefits can be expected. The utility of sensor with $\eta = 1$ is 0 since it quits the DSD game and sets its strategy X_{S_m} to be 0. On the other hand, the utility of type-2 LKS is strictly concave, and the black dashed line represents the constraint for completing the contract, as given in (21b). Accordingly, sensors can only pursue their maximal utilities under the given rewards R_2^L.

We plot Fig. 11 to show the effect of an LKS's rewards on the optimal data size of an sensor, where we take type-2 LKS and the sensor with $\eta = 3$ as the example, and set the unit cost for collecting data $c \in \{0.5, 1, 1.5, 2\}$. From Fig. 11, we can observe



(a) Optimal utility of LKS.



(b) Optimal utility of KSP.

Fig. 9. Optimal utilities of LKS and KSP under different knowledge training cost.

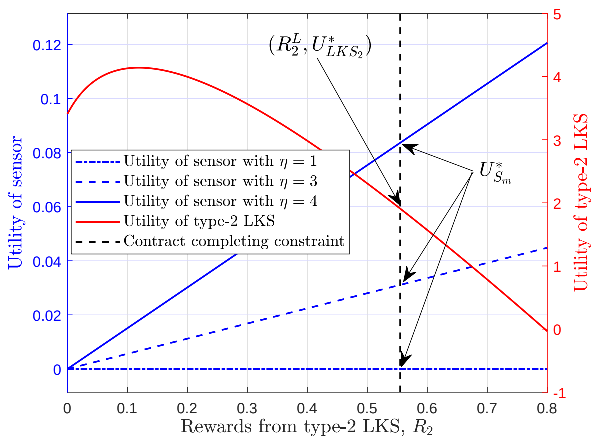


Fig. 10. Utilities of type-2 LKS and its sensors under the variation of rewards R_2.

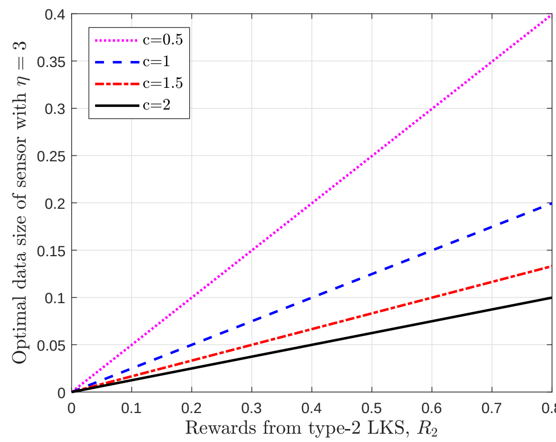


Fig. 11. Optimal data size of sensor under the variation of LKS's rewards.

that the sensor's optimal data size increases linearly as the rewards R_2 increases, which corresponds to the Nash Equilibrium of the DSD game calculated by (12). Another observation from Fig. 11 is that a higher cost for collecting data will lead to a smaller data size of the sensor. This is because the growth of c reduces the expected benefits of sensors, which makes them less willing to contribute more data.

7.3 Comparison Results

We take two typical schemes in trading markets for comparison and apply them to the BWKA ecosystem, which are defined as follows.

- *Complete Information* scheme [54]. With the complete information scheme, it is assumed that the KSP knows exactly the instantaneous type of each LKS, and thus it can reduce the utilities of LKSs to the non-contract scenario.
- *Linear Pricing* scheme [55]. With the linear pricing scheme, the KSP specifies a pre-defined price parameter for LKSs' knowledge generation. The pre-defined price parameter is the same for all LKSs, i.e., $Price_1 = Price_2 \cdots Price_N$.

For ease of presentation, the SKTC-based knowledge trading scheme for the considered BWKA ecosystem with incomplete information is referred to as the *Asymmetric Information* scheme.

Fig. 12 shows the optimal utilities that can be achieved by LKSs under different knowledge trading schemes. We can observe that for the *Asymmetric Information* scheme and the *Linear Pricing* scheme, the utilities of LKSs increase with their types (i.e., the operational effectiveness); while for the *Complete Information* scheme, the KSP is aware of all the types of LKSs, so it can squeeze LKSs' utilities by designing specific contracts targeting for every LKS. As a result, the LKS achieves the lowest utility with the *Complete Information* scheme. Note that even the lowest utility of an LKS is still higher than 0, this is because each LKS can obtain a non-negative utility $\widehat{U}_i^{NC}$ when it refuses all the contracts.

Fig. 13 shows how the optimal utility that can be achieved by the KSP varies with the type of LKS under different knowledge trading schemes. It can be observed that the KSP's optimal achievable utility increases with the type of LKS, which is consistent with the intuition that the LKS with higher operational effectiveness can bring greater profit to the KSP. Another interesting

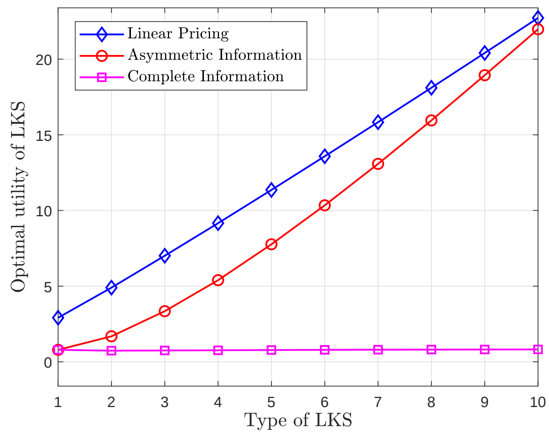


Fig. 12. Optimal utility of LKS under different knowledge trading schemes.

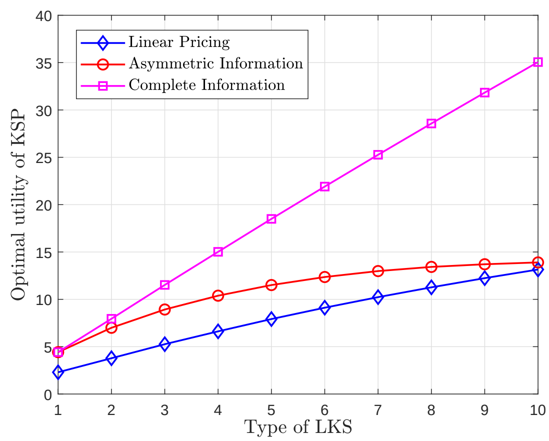


Fig. 13. Optimal utility of KSP under different knowledge trading schemes.

observation is that the KSP's optimal utility under the *Complete Information* scheme is superior to that under the *Asymmetric Information* scheme, and further superior to that under the *Linear Pricing* scheme. This phenomenon is the opposite of the trend presented in Fig. 12, indicating that the optimal achievable utility of KSP is negatively correlated to that of LKSs.

8 Conclusion and Future Work

In this paper, we have proposed a full-fledged knowledge acquisition ecosystem BWKA, which contains a complete process from the underlying data sensing, aggregating, and knowledge training, to the upper-layer knowledge trading. Leveraging the emerging blockchain technology, we have designed a secure and truthful data aggregating scheme based on the PoC consensus mechanism, as well as a reliable and fair knowledge trading scheme based on smart contracts. Incentive mechanisms have also been incorporated to stimulate selfish and rational entities to participate in the knowledge acquisition works. To identify the strategic interactions in the BWKA ecosystem, we have developed a nested hierarchical game model, where the upper layer is modeled by the Contract Theory and the lower layer is modeled as a two-stage Stackelberg game. By solving the hierarchical game in a backward inductive way, we obtained the optimal strategies of different entities. Experiments and simulations have been conducted to demonstrate the practical operability and performance behaviors of the BWKA ecosystem.

In this work, all entities involved in the BWKA ecosystem are perfectly rational, exclusively pursuing their self-interest maximization. Future research will explore the concept of bounded rationality among participants, introducing the theory of behavioral economics to analyze their strategic interactions that closely reflect real market dynamics. Also, note that in this work, the immutability and traceability of blockchain can, to a certain extent, deter the provision of fraudulent data and knowledge, but the detection and response to such malicious behaviors are not addressed. Therefore, the next step of this work will focus on refining the detection of false data and knowledge, as well as introducing corresponding trust and penalty mechanisms. Another interesting direction for future work involves developing similar knowledge acquisition ecosystems using alternative means other than blockchain technology. By conducting comparative experiments, we expect to validate the security performance gains attributed to blockchain utilization.

References

[1] C. Cai, Y. Zheng, A. Zhou, and C. Wang, "Building a secure knowledge marketplace over crowdsensed data streams," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 6, pp. 2601–2616, 2019.

[2] H. Jin, L. Su, D. Chen, H. Guo, K. Nahrstedt, and J. Xu, "Thanos: Incentive mechanism with quality awareness for mobile crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1951–1964, 2018.

[3] Z. Shi, G. Yang, X. Gong, S. He, and J. Chen, "Quality-aware incentive mechanisms under social influences in data crowdsourcing," *IEEE/ACM Trans. Netw.*, 2021.

[4] C. Cai, Y. Zheng, and C. Wang, "Leveraging crowdsensed data streams to discover and sell knowledge: A secure and efficient realization," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 589–599.

[5] X. Zhang, K.-K. R. Choo, and N. L. Beebe, "How do I share my IoT forensic experience with the broader community? An automated knowledge sharing IoT forensic platform," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6850–6861, 2019.

[6] J. Liu, Y. Yang, D. Li, X. Deng, S. Huang, and H. Liu, "An incentive mechanism based on behavioural economics in location-based crowdsensing considering an uneven distribution of participants," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 44–62, 2020.

[7] H. Lin, Y. Fan, J. Zhang, B. Bai, Z. Xu, and T. Lukasiewicz, "Toward knowledge as a service (KaaS): Predicting popularity of knowledge services leveraging graph neural networks," *IEEE Trans. Services Comput.*, 2022.

[8] X. Lin, J. Li, J. Wu, H. Liang, and W. Yang, "Making knowledge tradable in edge-AI enabled IoT: A consortium blockchain-based efficient and incentive approach," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6367–6378, 2019.

[9] *Waze Mobile App*. [Online]. Available: https://www.waze.com/

[10] *GasBuddy Mobile App*. [Online]. Available: https://www.gasbuddy.com/

[11] *Pavemint Mobile App*. [Online]. Available: https://www.pavemint.com/

[12] *WiFi Finder Connect Internet Mobile App*. [Online]. Available: https://itunes.apple.com/us/app/wifi-finder-connect-internet/id1011519183?mt=8

[13] C. Zhao, S. Yang, and J. A. McCann, "On the data quality in privacy-preserving mobile crowdsensing systems with untruthful reporting," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 647–661, 2019.

[14] P. Sun, Z. Wang, L. Wu, Y. Feng, X. Pang, H. Qi, and Z. Wang, "Towards personalized privacy-preserving incentive for truth discovery in mobile crowdsensing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 352–365, 2020.

[15] B. Li and J. Liu, "Achieving information freshness with selfish and rational users in mobile crowd-learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1266–1276, 2021.

[16] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3975–3986, 2020.

[17] X. Pang, Z. Wang, D. Liu, J. C. Lui, Q. Wang, and J. Ren, "Towards personalized privacy-preserving truth discovery over crowdsourced data streams," *IEEE/ACM Trans. Netw.*, 2021.

[18] Z. Liu, Y. Sang, B. Li, and B. Ji, "A worst-case approximate analysis of peak age-of-information via robust queueing approach," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*. IEEE, 2021, pp. 1–10.

[19] X. Xiang, W. Liu, T. Wang, M. Xie, X. Li, H. Song, A. Liu, and G. Zhang, "Delay and energy-efficient data collection scheme-based matrix filling theory for dynamic traffic IoT," *Eurasip J. Wirel. Commun. Netw.*, vol. 2019, no. 1, pp. 1–25, 2019.

[20] Y. Zou, F. Shen, F. Yan, J. Lin, and Y. Qiu, "Reputation-based regional federated learning for knowledge trading in blockchain-enhanced IoV," in *Proc. IEEE Wirel. Commun. Netw. Conf. (WCNC)*. IEEE, 2021, pp. 1–6.

[21] P. Sun, H. Che, Z. Wang, Y. Wang, T. Wang, L. Wu, and H. Shao, "Pain-FL: Personalized privacy-preserving incentive for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3805–3820, 2021.

[22] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3034–3048, 2020.

[23] S. Wu, Y. Chen, Q. Wang, M. Li, C. Wang, and X. Luo, "CReam: A smart contract enabled collusion-resistant e-auction," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1687–1701, 2018.

[24] R. Wang, J. Yan, D. Wu, H. Wang, and Q. Yang, "Knowledge-centric edge computing based on virtualized D2D communication systems," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 32–38, 2018.

[25] 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.*, 2020.

[26] M. Li, D. Hu, C. Lal, M. Conti, and Z. Zhang, "Blockchain-enabled secure energy trading with verifiable fairness in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6564–6574, 2020.

[27] S. Fan, H. Zhang, Y. Zeng, and W. Cai, "Hybrid blockchain-based resource trading system for federated learning in edge computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2252–2264, 2020.

[28] L. Xue, W. Yang, W. Chen, and L. Huang, "STBC: A novel blockchain-based spectrum trading solution," *IEEE Trans. Cogn. Commun. Netw.*, 2021.

[29] P. Sun, Z. Wang, Y. Feng, L. Wu, Y. Li, H. Qi, and Z. Wang, "Towards personalized privacy-preserving incentive for truth discovery in crowd-sourced binary-choice question answering," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*. IEEE, 2020, pp. 1133–1142.

[30] J. Nie, J. Luo, Z. Xiong, D. Niyato, P. Wang, and H. V. Poor, "A multi-leader multi-follower game-based analysis for incentive mechanisms in socially-aware mobile crowdsensing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1457–1471, 2020.

[31] 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, 2020.

[32] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, "A survey on blockchain for information systems management and security," *Inform Process Manag.*, vol. 58, no. 1, p. 102397, 2021.

[33] K. Liu, X. Qiu, W. Chen, X. Chen, and Z. Zheng, "Optimal pricing mechanism for data market in blockchain-enhanced Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9748–9761, 2019.

[34] J. Benet, "IPFS-content addressed, versioned, P2P file system," 2014, [Online]. Available: https://arxiv.org/abs/1407.3561.

[35] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, pp. 85–126, 2004.

[36] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, "Efficient and privacy-preserving truth discovery in mobile crowd sensing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3854–3865, 2019.

[37] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[38] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*. New York, NY, USA: Cambridge University Press, 2009.

[39] *Ethereum Whitepaper*. [Online]. Available: https://ethereum.org/en/whitepaper/.

[40] X. Ding, J. Guo, D. Li, and W. Wu, "An incentive mechanism for building a secure blockchain-based Internet of Things," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 477–487, 2020.

[41] A. Mas-Colell, M. D. Whinston, J. R. Green *et al.*, *Microeconomic Theory*. Oxford, U.K.: Oxford Univ. Press, 1995, vol. 1.

[42] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of Vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, 2019.

[43] M. P. Bolton, *Contract Theory*. Cambridge, MA, USA: MIT Press, 2005.

[44] H. Von Stackelberg, *The Theory of the Market Economy*. London, U.K.: Oxford Univ. Press, 1952.

[45] H. Yu, M. H. Cheung, L. Gao, and J. Huang, "Public Wi-Fi monetization via advertising," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2110–2121, 2017.

[46] H. Yu, E. Wei, and R. A. Berry, "A business model analysis of mobile data rewards," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*. IEEE, 2019, pp. 2098–2106.

[47] Z. Xiong, J. Kang, D. Niyato, P. Wang, H. V. Poor, and S. Xie, "A multi-dimensional contract approach for data rewarding in mobile networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5779–5793, 2020.

[48] M. Zeng, Y. Li, K. Zhang, M. Waqas, and D. Jin, "Incentive mechanism design for computation offloading in heterogeneous fog computing: A contract-based approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*. IEEE, 2018, pp. 1–6.

[49] R. B. Myerson, *Game Theory*. Cambridge, MA, USA: Harvard Univ. Press, 2013.

[50] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, pp. 520–534, 1965.

[51] Y. Xu, J. Liu, Y. Shen, X. Jiang, Y. Ji, and N. Shiratori, "QoS-aware secure routing design for wireless networks with selfish jammers," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4902–4916, 2021.

[52] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, 1995.

[53] M. Zhang, J. Huang, and R. Zhang, "Wireless power transfer with information asymmetry: A public goods perspective," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 276–291, 2019.

[54] C. Yang, W. Lou, Y. Liu, and S. Xie, "Resource allocation for edge computing-based vehicle platoon on freeway: A contract-optimization approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15 988–16 000, 2020.

[55] J. Du, E. Gelenbe, C. Jiang, H. Zhang, and Y. Ren, "Contract design for traffic offloading and resource allocation in heterogeneous ultra-dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2457–2467, 2017.

[56] *SKTC-based Knowledge Trading*, Dec., 2023. [Online]. Available: https ://sepolia.etherscan.io/address/0x998bbee6289985062d9bdcd4e4df2a218a513781#code.

[57] W. Zhang, Z. Hong, and W. Chen, "Hierarchical pricing mechanism with financial stability for decentralized crowdsourcing: A smart contract approach," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 750–765, 2020.

Yang Xu (Member, IEEE) received the B.E. degree from the School of Telecommunications Engineering and the Ph.D. degree from the Department of Communication and Information Systems, Xidian University, Xi'an, China, in 2006 and 2014, respectively, where she is currently an associate professor with the School of Computer Science and Technology. She has published over 50 academic papers in premium journals and conferences, like IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING (TDSC), IEEE TRANSACTIONS ON MOBILE COMPUTING (TMC), IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (TWC), the IEEE INTERNET OF THINGS JOURNAL, and the IEEE INFOCOM. Her research interests include wireless communications security, mobile crowdsensing, network economics, space-air-ground integrated networks, routing protocol design, etc.

Jianbo Shao (Graduate Student Member, IEEE) received the B.E. degree and M.S. degree from the School of Economics and Management, Xidian University, Xi'an, Chian, in 2020 and 2023, respectively. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Shanghai University of Finance and Economics. His research interests include intelligence analysis, blockchain, game theory, network economics, and so on.

Jia Liu (Senior Member, IEEE) received the B.E. degree from the School of Telecommunications Engineering, Xidian University, Xi'an, China, in 2010, and received the Ph.D. degree from the School of Systems Information Science, Future University Hakodate, Japan, in 2016. His research interests include wireless systems security, space-air-ground integrated networks, Internet of Things, 6G, etc. He has published over 70 academic papers in premium journals and conferences, like IEEE Transactions on Dependable and Secure Computing (TDSC), IEEE Transactions on Mobile Computing (TMC), IEEE Transactions on Information Forensics and Security (TIFS), and IEEE INFOCOM. He received the 2016 and 2020 IEEE Sapporo Section Encouragement Award.

Yulong Shen (Member, IEEE) received the B.S. and M.S. degrees in computer science and Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2002, 2005, and 2008, respectively. He is currently a Professor with the School of Computer Science and Technology, Xidian University, where he is also an Associate Director of the Shaanxi Key Laboratory of Network and System Security and a member of the State Key Laboratory of Integrated Services Networks. His research interests include wireless network security and cloud computing security. He has also served on the technical program committees of several international conferences, including ICEBE, INCoS, CIS, and SOWN.

Tarik Taleb (Senior Member, IEEE) received the B.E. degree (with distinction) in information engineering and the M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively. He is currently a Full Professor at Ruhr University Bochum, Germany. He was a Professor with the Center of Wireless Communications, University of Oulu, Oulu, Finland. He is the founder of ICTFICIAL Oy, and the founder and the Director of the MOSA!C Lab, Espoo, Finland. From October 2014 to December 2021, he was an Associate Professor with the School of Electrical Engineering, Aalto University, Espoo, Finland. Prior to that, he was working as a Senior Researcher and a 3GPP Standards Expert with NEC Europe Ltd., Heidelberg, Germany. Before joining NEC and till March 2009, he worked as Assistant Professor with the Graduate School of Information Sciences, Tohoku University, in a lab fully funded by KDDI. From 2005 to 2006, he was a Research Fellow with the Intelligent Cosmos Research Institute, Sendai. Taleb has been directly engaged in the development and standardization of the Evolved Packet System as a member of the 3GPP System Architecture Working Group. His current research interests include AI-based network management, architectural enhancements to mobile core networks, network softwarization and slicing, mobile cloud networking, network function virtualization, software-defined networking, software-defined security, and mobile multimedia streaming.

Norio Shiratori (Life Fellow, IEEE) received his Ph.D. degree from Tohoku University in 1977. He became an Assistant Professor and Associate Professor at the Research Institute of Electrical Communication, Tohoku University, in 1977 and 1984, respectively. In 1990, he was promoted to Full Professor at the School of Engineering, Tohoku University. In 1993, he commenced his role as Full Professor at the Research Institute of Electrical Communication, Tohoku University. In 1997, Prof. Shiratori became a Visiting Professor at UCLA (University of California, Los Angeles). In 1998, he was elevated to the status of IEEE Fellow. In 2004, Prof. Shiratori assumed the role of Japan representative for the International Federation for Information Processing (IFIP). In 2009, he served as the President of the Information Processing Society of Japan.

In 2010, following his retirement from Tohoku University, Prof. Shiratori took on the positions of Professor Emeritus at Tohoku University and Board Member at Future University Hakodate. In the same year, he became the Chair of the IEEE Sendai Section. In 2012, he was appointed as a Professor at the Global Information and Telecommunication Institute, Waseda University. In 2013, Prof. Shiratori assumed the role of Vice Chair of the IEEE Japan Council. Recognizing his significant contributions, he was honored with the title of IEEE Life Fellow in 2017. Since 2017, he has been serving as a Professor at Research and Development Initiative, Chuo University.

Prof. Shiratori has published over 15 books and over 600 refereed papers in computer science and related fields. He is a fellow of the Japan Foundation of Engineering Societies (JFES), the Information Processing Society of Japan (IPSJ), and the Institute of Electronics, Information and Communication Engineers (IEICE). He was a recipient of the Minister of MEXT Award from the Japanese Government in 2016, the Science and Technology Award from the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) in 2009, the IEICE Achievement Award in 2001, the IEICE Contribution Award in 2011, the IPSJ Contribution Award in 2008, the IEICE Honorary Member in 2012, the IPSJ Honorary Member in 2013, IPSJ Memorial Prize Winning Paper Award in 1985, the IPSJ Best Paper Award in 1997, the IEICE Best Paper Award in 2001, the IEEE 5th SCE01 Best Paper Award in 2001, the IEEE ICPADS 2000 Best Paper Award in 2000, and the IEEE 12th ICOIN Best Paper Award.