This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Incentivizing Socio-Ethical Integrity in Decentralized Machine Learning Ecosystems for Collaborative Knowledge Sharing

Yuanfang Chi, *Student Member, IEEE*, Qiyue Zhang, *Student Member, IEEE*, Jiaxiang Sun, *Student Member, IEEE*, Wei Cai, *Senior Member, IEEE*, Z. Jane Wang, *Fellow IEEE*, and Victor C. M. Leung, *Life Fellow, IEEE*

***Abstract*—To broaden domain knowledge and enable advanced analytics, machine learning (ML) algorithms increasingly utilize comprehensive datasets across diverse sectors. However, these disparate datasets held by various stakeholders raise concerns over data heterogeneity, privacy, and security. Decentralized ML research aims to protect data privacy and integrate knowledge bases, especially knowledge graphs, to address data heterogeneity challenges. Yet, the question of how to foster trustworthy collaborations in decentralized ML ecosystems remains underexplored. This study pioneers two innovative socio-economic mechanisms designed to ensure dependable collaborations with socio-ethical integrity within a decentralized knowledge inference framework, enabling participants to share knowledge while maintaining data privacy and ethical standards. We employ an evolutionary game theory model to analyze the dynamic interactions between requestors and workers, focusing on achieving a stable equilibrium through theoretical and numerical evaluations. Furthermore, we explore how various critical factors, such as incentive schemes and the accuracy of identifying malicious workers, influence the system's equilibrium, providing insights into optimizing collaborative efforts in decentralized ML ecosystems.**

***Index Terms*—Decentralized computing, evolutionary game theory, reputation system.**

I. Introduction

In the digital era, sensors, machines, smart edge devices, or personal mobile computing devices are incessantly producing vast quantities of data on a daily basis, compelling the adoption of sophisticated analytical methodologies. Among these, machine learning (ML) algorithms markedly outshine traditional approaches in navigating the complexities of extensive data warehouses [1], thereby significantly augmenting efficiency and competitive advantage in sectors such as manufacturing, logistics, and supply chain management. The recent evolution of ML, especially through advanced large language models (LLMs) such as GPT-4 and Gemini, has ushered in breakthroughs in natural language processing domains, revolutionizing capabilities in text analysis, linguistic translation, and the generation of domain-specific responses. Such advancements underscore the transformative impact of ML algorithms and LLMs on many industry practices [2], [3].

It is widely acknowledged that the performance of a machine learning (ML) algorithm is intrinsically linked to the volume and integrity of the data it processes. However, this data is frequently distributed across disparate locations, held by various stakeholders. In practice, these stakeholders often possess expertise in distinct facets of the production or distribution chain. Consequently, collaboration among diverse stakeholders becomes essential to navigate the complexities of modern supply chains, driving innovation, ensuring customer satisfaction, and adhering to regulatory standards. For example, the integration of sensors, machines, smart edge devices, and personal mobile computing devices owned by different stakeholders through industrial Internet of Things (IoT) networks leads to the formation of sophisticated industrial IoT ecosystems. The availability of real-time data enables these geographically separated, autonomous systems to work together seamlessly as if they were a single entity. This unified system is capable of conducting

Manuscript received 26 March 2024; revised 4 August 2024; accepted 19 August 2024. This work was supported in part by the National Nature Science Foundation of China under Grant 62102266, in part by Guangdong Pearl River Talent Recruitment Program under Grant 2019ZT08X603, in part by Guangdong Pearl River Talent Plan under Grant 2019JC01X235, in part by Shenzhen Science and Technology Innovation Commission under Grant R2020A045 and Grant KCXFZ20201221173411032, in part by Shenzhen Science and Technology Program under Grant JCYJ20210324124205016, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515012323, and in part by the Canadian Natural Sciences and Engineering Research Council. *(Corresponding author: Victor C. M. Leung.)*

Yuanfang Chi is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: yuanchi@ece.ubc.ca).

Qiyue Zhang, Jiaxiang Sun, and Wei Cai are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: qiyuezhang@link.cuhk.edu.cn; jiaxiangsun@link.cuhk.edu.cn; weicai@ieee.org).

Z. Jane Wang is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: zjanew@ece.ubc.ca).

Victor C. M. Leung is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: vleung@ieee.org).

Digital Object Identifier 10.1109/TCSS.2024.3450494

2329-924X © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

distributed data sharing and advanced analytics, facilitating essential business processes within industrial IoT frameworks. One notable application is the execution of advanced intelligent fault diagnosis procedures, critical for maintaining the system's overall reliability, as highlighted in [4]. Thus, collaboration through knowledge sharing is beneficial when obtaining ML algorithms or LLMs for analysis or predictions for business operations that involve different stakeholders.

However, the pursuit of collaborative efforts in deploying ML techniques across various stakeholders has unearthed two primary socio-technical challenges. First, privacy and security concerns emerge as critical socio-ethical issues, especially within complex systems operated by disparate entities [5]. To address these concerns, distributed ML approaches like federated learning have been extensively explored. Federated learning enables the sharing of knowledge by training an ML model on local datasets held by each stakeholder, thus mitigating data privacy and security issues [6]. An industrial-scale federated learning framework tailored for LLMs is discussed in [7]. Additionally, adhering to the decentralized ethos of the Web 3.0 paradigm offers a pathway to surmount privacy and security challenges. This involves leveraging socio-technical innovations such as blockchain, consensus algorithms, and smart contracts, facilitating collaboration among parties without the need for central oversight or mutual trust [8]. The second challenge revolves around data heterogeneity, complicating distributed ML and knowledge sharing among stakeholders. To navigate this challenge, research has focused on employing universal knowledge bases (KBs), represented through knowledge graphs known for their adeptness at managing data heterogeneity, as a data source for ML algorithms. For instance, the application of intelligent fault diagnosis using KBs represented by knowledge graphs is investigated in [4]. Furthermore, the efficacy of LLMs is enhanced when trained on knowledge graphs as the data source [9], [10], addressing the "black-box" nature and hallucination issues associated with LLMs [11]. In conclusion, for ML algorithms to be effectively integrated into industrial settings and deliver comprehensive analytics or predictions, they must support collaborative knowledge sharing. This involves aggregating domain knowledge held by various socio-economic stakeholders through well-structured knowledge representations, such as knowledge graphs. Doing so in a decentralized fashion is crucial for overcoming the challenges of privacy and security, as well as data heterogeneity.

To facilitate collaborative knowledge sharing across distributed knowledge graphs maintained by distinct stakeholders, Ref. [12] presents a novel decentralized framework. This framework empowers participants to collaboratively train a reasoning model using a distributed path-based reasoning algorithm. This algorithm integrates data from their respective local knowledge graphs into a reasoning model. Knowledge dissemination is facilitated through the sharing of the trained reasoning model, allowing all participants to benefit from the collective insights. The approach addresses privacy, security, and data heterogeneity challenges by leveraging smart contracts and blockchain technology for participant interactions, alongside a decentralized knowledge inference algorithm capable of learning from various independent knowledge graphs without the necessity for direct data exchange. In this framework, a participant, designated as the requestor, initiates the training process, while subsequent participants evaluate their ability to contribute as workers utilizing their local knowledge graphs. The requestor then selects the most suitably evaluated worker for collaboration. Given that knowledge is deemed a critical asset, the development of effective socio-economic incentive mechanisms is essential. These mechanisms should encourage participants to share pertinent knowledge timely and identify and exclude malicious participants to prevent the dissemination of harmful data.

Within the framework proposed in [12], requestors select workers based on their self-evaluation results in descending order. Consequently, the quality of the model trained by the framework largely depends on these self-evaluation results. If requestors incentivize workers to participate in a knowledge-sharing task with rewards, self-interested workers might be motivated to inflate their evaluation results. However, the authors of [12] assume that workers will report their evaluation outcomes honestly, leading to the development of efficient and accurate inference models. This assumption does not consider the potential for worker dishonesty driven by self-interest. Such dishonesty could conflict with the requestor's objectives, highlighting a critical need for further investigation into the framework's design to ensure robustness and reliability in real-world applications. In this research, we pioneer the introduction of two mechanisms aimed at fostering reliable collaborations with socio-ethical integrity within the decentralized framework for knowledge inference proposed in [12]. The first mechanism we introduce is a reputation system engineered to evaluate the trustworthiness of workers. This system enables requestors to make informed selections of workers by considering both the workers' self-assessments and their established reputation scores. The second mechanism, a spot-check system, allows requestors to perform concurrent evaluations by engaging multiple workers in collaborative training tasks. This approach is instrumental in identifying workers who may have inaccurately represented their capabilities, thus promoting socio-ethical integrity. To dissect and understand the dynamics of decision-making between workers and requestors under these mechanisms, we employ evolutionary game theory. This theoretical model aids in analyzing how both parties adapt their strategies over time, based on the outcomes of their interactions. Our findings underscore the critical role of spot-checks in effectively identifying dishonest workers, which is paramount in encouraging integrity among workers and optimizing the collective socio-economic welfare of the system. The core contributions of our study are detailed as follows:

1) This is, to our knowledge, the first work to integrate a reputation system alongside a spot-check mechanism within a decentralized knowledge sharing framework, leveraging a decentralized machine learning algorithm. This novel socio-economic approach aims to ensure worker honesty and reliability.
2) We apply evolutionary game theory to meticulously model the complex interplay between workers and

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

requestors, taking into account their strategic decision-making processes. Our analysis further explores how various key factors, such as the incentives for participation and the effectiveness of identifying dishonest participants, impact the system's equilibrium and socio-ethical integrity.

3) Our theoretical deductions regarding the impact of these factors are validated through numerical evaluations. Notably, our results highlight the paramount importance of the precision of requestors' spot-checks in fostering trustworthy collaborations.

The remainder of this article is structured as follows: Section III provides an in-depth description of the proposed reputation and spot-check mechanisms for ensuring socio-ethical integrity. In Section IV, we present the evolutionary game model and analyze the equilibrium states of the system with regard to socio-ethical integrity. Section V offers a detailed account of the simulations conducted to evaluate the effectiveness of these socio-economic models in maintaining socio-ethical integrity. Section VI concludes the article.

II. Related Work

A. Machine Learning Algorithms for Knowledge Graphs

Stakeholders across various domains collaborate to deliver final products, necessitating distributed ML algorithms for enhanced analytics and predictions. This collaboration often faces the challenge of data heterogeneity, as each stakeholder may use unique data structures for their datasets. Addressing this, knowledge graphs serve as a universal knowledge base, utilizing graph-structured models like resource description framework (RDF) triples to store and interlink entity descriptions, encapsulating both entities and their semantic relationships. In [13], the authors introduced TransE, an algorithm that represents entities and relations as low-dimensional vectors in an embedding space, enabling the prediction of relationships between two entities by analyzing the differences between their embeddings. Authors in [14] expanded upon TransE to address one-to-many and many-to-many entity relationships by mapping the knowledge graph into a continuous vector space, enabling the application of machine learning algorithms on these embeddings to deduce specific relationships between entities. Despite the utility of large-scale knowledge graphs, exhaustively verifying the existence of specific relationships between two ontologies is both time and resource-intensive. To address this, knowledge graphs can be analyzed using graph topology algorithms, treating subjects and objects as vertices, and predicates as paths connecting these vertices. The path ranking algorithm (PRA) [15] is a notable method for reasoning within graphs, utilizing a random walk to traverse from a head entity h to an end entity t across specified lengths. For a given entity pair (h, t) and a path r, PRA computes the feature value as the probability of a random walk reaching t from h via r. This probability aids in determining the presence of a particular relation r between (h, t), with path feature weights refined through logistic regression. Although PRA offers a robust framework for reasoning across extensive knowledge graphs, the unguided enumeration of paths remains computationally demanding. To enhance efficiency, [16] introduced a path-constrained version of PRA, focusing random walks on paths pertinent to a target entity within a constrained length during training. Moreover, [17] improved the predictive capability of path-constrained PRA by integrating syntactic patterns from text corpora and semantic patterns from background knowledge. Primarily applied for link prediction and knowledge base completion, PRA aims to discern the existence of a relationship between two nodes, facilitating the discovery of new relations in applications.

B. Decentralized Machine Learning

Decentralized ML has been widely adopted in edge-based ML, which markedly diminishes delays and bolsters ML application efficacy in real-time scenarios. Meanwhile, distributed knowledge inference has been studied to tackle the scalability, performance, and KB isolation issues. However, such edge intelligence mandates the orchestration among diverse ML services across end devices, edge nodes, and cloud platforms, presenting technical challenges. These include optimizing the allocation and security of decentralized computational resources, establishing connectivity among distributed edges for collaborative data management, and safeguarding distributed training and inference processes on confidential datasets [18]. To tackle the privacy and security issue, emerging technologies like blockchain, consensus algorithms, and smart contracts are increasingly explored to facilitate collaboration in decentralized systems without the need for central control or mutual trust. This trend includes blockchain-based methodologies for decentralized federated transfer learning [19], [20], trading systems for federated learning [21], and decentralized mechanism for distributed path-based reasoning using distributed knowledge graphs [12], highlighting the diverse applications and potential of blockchain technology in enhancing decentralized ML processes and collaboration. In this work, we explore incentive mechanisms for the decentralized mechanism for distributed path-based reasoning using distributed knowledge graphs proposed in [12].

C. Incentivization and Evolutionary Game Theory

Focus on incentivization within decentralized machine learning is increasing [22], with smart contracts being explored to motivate end device participation in decentralized learning frameworks [23]. Evolutionary game theory, pivotal in analyzing strategic interactions within dynamically evolving populations, merges classical game theory, evolutionary biology, and mathematical modeling. This approach contrasts with classical game theory's assumption of fully rational players in a transparent rule set. It accounts for bounded rationality and informational constraints, diverging from traditional game-theoretical models that feature abrupt strategic shifts and manipulative strategies [24]. Widely applicable, evolutionary game theory informs strategies in industrial and economic sectors. It elucidates strategic interdependencies among stakeholders, impacting decision-making processes [25] and incorporates psychological factors in strategy formulation [26]. In industry,

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

it is a key tool for analyzing dynamics in different industries [27], [28]. Its utility extends to IoT task offloading [29], crowdsourcing task distribution [30], and blockchain incentive mechanisms [31], demonstrating its broad applicability in solving complex, real-world challenges. Furthermore, integrating IoT devices with blockchain technology ensures a secure and privacy-preserving IoT platform. An evolutionary game-based pool selection algorithm is proposed for IoT devices to choose an optimal cloud mining pool for blockchain mining tasks [32]. In [33], evolutionary game theory is leveraged to assist users in selecting the most suitable server and code configurations for opportunistically coded distributed computing, optimizing the execution time for computationally intensive tasks. While existing studies primarily aim to incentivize computational resource contributions, this paper proposes a novel incentive mechanism designed to facilitate knowledge sharing among participants in a decentralized setting, addressing a vital aspect that has been less explored in the literature. Furthermore, in this work, we use evolutionary game theory to model and analyze the decision process of participants of a decentralized knowledge inference framework with our proposed incentive mechanism.

III. System Design

In this work, we introduce an innovative incentive mechanism tailored for the decentralized framework designed for crowdsourcing knowledge inference tasks, as proposed in [12]. This framework facilitates collaborative training by allowing a requestor to initiate a request, to which workers respond by self-evaluating their capabilities to fulfill the request. They then communicate their self-assessments as confidence indicators to the request matching interface, which is orchestrated through a smart contract. The requestor proceeds to select the worker with the most favorable confidence indicator for collaborative training. However, this model presupposes the veracity of workers' self-reported confidence levels and does not address the critical role of compensation as an incentive for workers to engage in collaborative training tasks.

To mitigate these concerns, we introduce a novel mechanism in which compensation from requestors is securely predeposited into a smart contract and is automatically transferred to workers upon successful task completion, as illustrated in Fig. 1. This mechanism is designed with the aim of optimizing the utility for both workers and requestors. Workers, under this system, have motivate to report higher confidence indicators to increase their chances of being selected for tasks, thus augmenting their potential earnings. Conversely, requestors strive to achieve the most superior results at the lowest possible cost. They favor a system that guarantees worker honesty, thereby averting the risk of financial resources being diverted to fraudulent workers, which could result in inferior models and diminished investment returns.

Evidently, the accuracy of workers' reported confidence levels is pivotal for the system's efficacy. As highlighted in [12], the confidence indicator—reflecting the degree of alignment between the knowledge graphs of requestors and workers—plays a critical role in influencing the outcomes of collaborative training. Generally, a higher confidence indicator is associated with improved training performance. Specifically, the system employs path-based reasoning algorithms for decentralized knowledge inference, executing random walks on the graph to identify paths from a head entity h to a target entity t within defined lengths. An essential component of collaborative is pinpointing the handover entity e_{hq} within the worker's local knowledge base. The absence of e_{hq} can result in the reasoning agent encountering a dead-end, adversely affecting training efficacy. Therefore, even with a high degree of overall correspondence between knowledge graphs, the lack of key entities like e_{hq} can hinder the enhancement of the trained reasoning model. This raises a challenge for requestors in determining whether diminished training quality is due to inaccurately reported confidence levels. To address this, we suggest the implementation of a reputation system for workers alongside a spot-check mechanism for requestors. These measures are aimed at evaluating and fostering worker honesty, thereby enhancing the integrity and effectiveness of the collaborative training process.

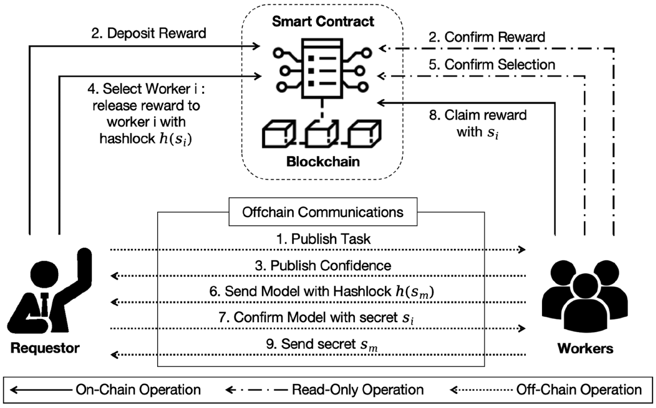


Fig. 1. Workflow of the proposed mechanism.

A. Reputation Mechanism Design

In conventional auction systems, requestors typically rank multiple bidding workers according to assigned weights, selecting the highest-ranked or a few top-ranking individuals for tasks. As highlighted in [12], such rankings have traditionally relied exclusively on workers' self-assessed confidence indicators, thereby creating a loophole for workers to overstate their confidence for better selection prospects. To address this issue, we introduce a novel reputation system, which assigns each worker a dynamic reputation value reflective of their honesty throughout the operation of the system. In this enhanced mechanism, the final ranking of a worker is a function of both their reputation and confidence levels. The smart contract in this framework accordingly arranges the participating workers for collaborative training, taking into account a combination of their confidence indicators and reputation scores. This approach effectively mitigates the risk of selecting malicious workers who frequently submit inflated self-evaluations, thereby safeguarding the integrity and accuracy of the training process.

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

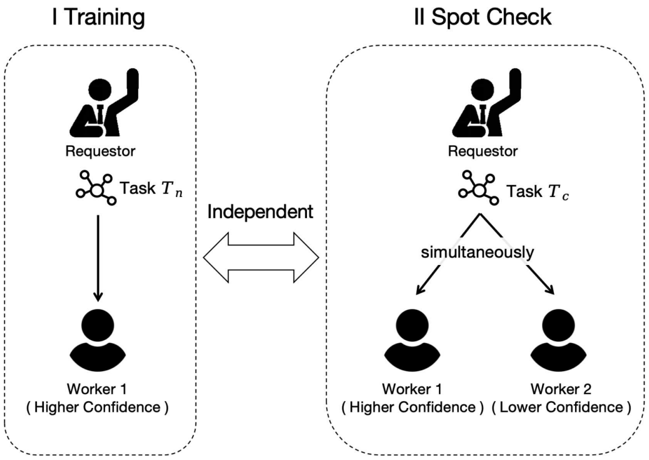


Fig. 2. Proposed spot-check mechanism.

B. Spot-check Mechanism Design

We introduce a spot-check mechanism (as illustrated in Fig. 2) that allows requestors to adjust workers' reputation scores dynamically based on their performance. Initially, all participating workers are assigned the same reputation score. In each collaboration cycle, the requestor is presented with two options: proceed with standard training with the top-ranked worker or initiate a spot-check. A spot-check entails selecting not only the highest-ranked worker but also one or more lower-ranked workers for the training task. This approach enables the requestor to assess the reliability of workers by comparing their training outcomes and self-reported confidence levels. For simplicity, this model assumes a single requestor and two workers, though the mechanism is designed to accommodate more complex configurations with multiple participants. Both workers undergo model training and spot-checking. In this work, we assume that the spot-check algorithm has already been designed and is implementable by requestors. However, it is crucial to recognize that the results of a spot-check are not absolute; higher confidence levels do not always guarantee better training outcomes.

Thus, in our approach, the reputation score of a worker is determined through a series of spot-checks across T rounds, with the initial reputation score at 1. Throughout these rounds, the requestor assesses the worker's honesty. Let t_1 denote the count of rounds in which the worker is deemed honest, and t_2 the count of rounds in which the worker is considered malicious, fulfilling the equation $t_1 + t_2 = T$. The outcome of the $(T+1)$th round plays a crucial role in adjusting the worker's reputation. Specifically, if worker 1 is found to be malicious in the $(T+1)$th round, their reputation score is reduced by the ratio t_2/T. In contrast, if they are judged to be honest in the $(T+1)$th round, their reputation score is increased by the ratio t_1/T. A worker with a negative reputation score is marked as consistently malicious. To become eligible again for selection in standard training sessions, such a worker must consistently exhibit honest behavior over time.

IV. Game Model

In this section, we utilize game theory to analyze the impact of the spot-check mechanism and reputation system integrated into our proposed framework on promoting worker honesty and enhancing the overall social welfare for both requestors and workers. Traditional game theory often assumes that participants are rational and aim to maximize their utility. However, our framework accommodates a broader range of worker behaviors, including those that are non-cooperative and potentially irrational. The interactions between requestors and workers in our system are dynamic, characterized by continuous adaptation and evolution, akin to an ongoing game. Given the complexity of these interactions, evolutionary game theory emerges as the most suitable analytical tool for our model, allowing us to explore how strategies evolve over time in response to the changing environment and participant behaviors. In evolutionary game theory, the process begins with both parties determining their respective payoff matrices. Using these matrices, each party calculates their dynamic replicator equation, which guides strategy adjustment. At the onset of the game, both parties select an initial set of strategies. As the game progresses, if the dynamic replicator equation yields a positive value for a particular strategy, it indicates that this strategy is more advantageous than others. Consequently, the party will increase the likelihood of selecting this strategy in future iterations. Conversely, if the value is negative, the probability of choosing that strategy will decrease. This iterative adjustment process continues, enabling both parties to refine their strategies based on the outcomes of their dynamic replicator equations. Equilibrium is achieved when the dynamic replicator equations for both parties converge to zero. This indicates that neither party can improve their payoff by unilaterally changing their strategy, resulting in a stable state where strategy probabilities remain constant over time.

A. Payoff Matrix

In our system, we assume workers can either report their actual confidence indicator or an inflated one, while the requestor has the option to conduct either normal training or spot-checking. Moreover, when conducting normal training, the requestor has the option to offer a high or low reward. We summarize key notations of the model in Table I. Following our system setup, the payoff matrix for both requestor and workers is formulated in Table II.

As detailed in the table, when the requestor opts for normal training and the worker accurately reports their confidence level, the requestor incurs costs comprising the worker's reward and the communication cost C_R for each interaction with the smart contract. The net value of the posttraining model is denoted by V. The requestor may choose to offer a high reward R_H or a low reward R_L, resulting in respective payoffs of $V - R_H - C_R$ and $V - R_L - C_R$. Correspondingly, the worker's payoff for reporting truthfully is $\theta R_H - C_T - C_E$ for a high reward, or $\theta R_L - C_T - C_E$ for a low reward. Conversely, if the worker reports a false confidence level during normal training, affecting the model's quality by ΔV and the likelihood of selecting a dishonest worker by ϵ, the requestor's payoffs

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
DESCRIPTION OF NOTATIONS IN OUR SYSTEM MODEL

Notation	Description
α	The probability that requestors offer low rewards
x	The probability that workers report their real confidence
y	The probability that requestors conduct a spot-check
θ	The probability of being selected to train a model when the worker reports its real confidence indicator
ϵ	The probability of being selected to train a model when the worker reports its inflated confidence indicator
p	The probability that requestors make correct judgment in spot-check
R_L	Low rewards offered by requestors
R_H	High rewards offered by requestors
R_C	Check rewards offered by requestors
V	The value of the model a requestor obtains if the worker reports its real confidence
ΔV	The difference of the model value a requestor obtains if the worker reports its inflated confidence, compared to V
V_R	The value requestors receive if a malicious worker
C_R	Cost of requestors to perform an interaction with smart contract
C_T	Cost of workers to train the model
C_E	Cost of workers to do self-evaluation
C_L	Losses of workers being perceived as malicious by the requestors

TABLE II
PAYOFF MATRIX

		Requestor		
		Normal training $(1-y)$		Spot-check (y)
		High reward $(1-\alpha)$	Low reward (α)	Checking reward
Worker	Real confidence (x)	$\theta R_H - C_T - C_E$ $V - R_H - C_R$	$\theta R_L - C_T - C_E$ $V - R_L - C_R$	$R_C - C_T - C_E - (1-p)C_L$ $-2(R_C + C_R)$
	Overvalued confidence $(1-x)$	$\epsilon R_H - C_T - C_E$ $V + \Delta V - R_H - C_R$	$\epsilon R_L - C_T - C_E$ $V + \Delta V - R_L - C_R$	$R_C - C_T - C_E - pC_L$ $-2(R_C + C_R) + V_R$

for choosing high or low rewards are modified to $V + \Delta V - R_H - C_R$ and $V + \Delta V - R_L - C_R$. The worker's potential earnings in such instances are either $\epsilon R_H - C_T - C_E$ for a high reward or $\epsilon R_L - C_T - C_E$ for a low reward. During a spot-check involving an honest worker, the worker is awarded R_C. The requestor's payoff is enhanced by an additional V_R upon detecting a malicious worker, although this comes at double the cost. Should a worker be erroneously classified as malicious, they face a penalty of C_L. Given p as the probability of accurately identifying dishonest behavior, the requestor's payoff is calculated as $-2(R_C + C_R)$, while the honest worker's payoff is $R_C - C_T - C_E - (1-p)C_L$. In cases where a spot-check identifies a malicious worker, the requestor's payoff is adjusted to $-2(R_C + C_R) + V_R$, with the malicious worker's payoff being $R_C - C_T - C_E - pC_L$.

B. Analysis

We define the probability of a worker choosing to report either a real or inflated confidence indicator as x and $1-x$, respectively. Similarly, the likelihood of a requestor opting for normal training versus spot-check is denoted as y and $1-y$, correspondingly. Additionally, in scenarios where the requestor conducts normal training, the probability of offering a low or high reward is represented by α and $1-\alpha$, respectively. Based on these probabilities, we then derive the expected payoff function for workers when they report a real confidence indicator as

$$\begin{aligned} E_{(x)} = y(&R_C - C_T - C_E - (1-p)C_L) \\ &+ (1-y)(\alpha(\theta R_L - C_T - C_E) \\ &+ (1-\alpha)(\theta R_H - C_T - C_E)). \end{aligned} \tag{1}$$

The expected payoff for workers to report an overvalued confidence indicator is

$$\begin{aligned} E_{(1-x)} = y(&R_C - C_T - C_E - pC_L) \\ &+ (1-y)(\alpha(\epsilon R_L - C_T - C_E) \\ &+ (1-\alpha)(\epsilon R_H - C_T - C_E)). \end{aligned} \tag{2}$$

Then, the average return of workers is

$$\overline{E}_{(x)} = xE_{(x)} + (1-x)E_{(1-x)}. \tag{3}$$

In addition, the dynamic replicator equation of workers is

$$\begin{aligned} F_{(x)} = \frac{\mathrm{d}x}{\mathrm{d}t} &= x(E_{(x)} - \overline{E}_{(x)}) \\ &= x(1-x)((1-y)(\theta - \epsilon)(\alpha R_L + (1-\alpha)R_H) \\ &\quad - yC_L + 2ypC_L). \end{aligned} \tag{4}$$

The expected payoff function for requestors to decide to perform spot-checks is

$$\begin{aligned} E_{(y)} &= x(-2(R_C + C_R)) + (1-x)(-2(R_C + C_R) + V_R) \\ &= -2(R_C + C_R) + (1-x)V_R. \end{aligned} \tag{5}$$

The expected payoff for requestors to decide to perform normal training is

$$\begin{aligned} E_{(1-y)} = x(&V - \alpha R_L - C_R - (1-\alpha)R_H) \\ &+ (1-x)(-\alpha R_L + V + \Delta V - C_R - (1-\alpha)R_H) \\ &= -\alpha R_L + V - C_R - (1-\alpha)R_H + (1-x)\Delta V. \end{aligned} \tag{6}$$

The average return of requestors is

$$\overline{E}_{(y)} = yE_{(y)} + (1-y)E_{(1-y)} \tag{7}$$

The dynamic replicator equation of requestors is

$$\begin{aligned} F_{(y)} = \frac{\mathrm{d}y}{\mathrm{d}t} &= y(E_{(y)} - \overline{E}_{(y)}) = y(1-y) \\ &(-2R_C - C_R - V + \alpha R_L + (1-\alpha)R_H \\ &+ (1-x)(V_R - \Delta V)). \end{aligned} \tag{8}$$

By combining (4) and (8), we construct the replication dynamic system. Setting (4) equal to 0 and (8) equal to 0, we can determine the local equilibrium points (LEPs) of the system. These are identified as: $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$, and (x^*, y^*), where

$$x^* = \frac{-2R_C - C_R - V + \alpha R_L + (1-\alpha)R_H + V_R - \Delta V}{V_R - \Delta V} \tag{9}$$

and

$$y^* = \frac{(\theta - \epsilon)(\alpha R_L + (1-\alpha)R_H)}{(\theta - \epsilon)(\alpha R_L + (1-\alpha)R_H) + C_L - 2pC_L}. \tag{10}$$

An evolutionarily stable strategy (ESS) for LEPs can be determined using the Jacobian matrix [24]:

$$J = \begin{bmatrix} \frac{\partial F(x)}{\partial x} & \frac{\partial F(x)}{\partial y} \\ \frac{\partial F(y)}{\partial x} & \frac{\partial F(y)}{\partial y} \end{bmatrix} \tag{11}$$

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
EXPRESSION OF DETJ AND TRJ

Equilibrium point	DetJ	TrJ
(0,0)	$(\theta-\epsilon)(\alpha R_L+(1-\alpha)R_H)(-2R_C-C_R-V+\alpha R_L+(1-\alpha)R_H+V_R-\Delta V)$	$(\theta-\epsilon)(\alpha R_L+(1-\alpha)R_H)+(-2R_C-C_R-V+\alpha R_L+(1-\alpha)R_H+V_R-\Delta V)$
(1,0)	$(2p-1)C_L(2R_C+C_R+V-\alpha R_L-(1-\alpha)R_H-V_R+\Delta V)$	$(2p-1)C_L+(2R_C+C_R+V-\alpha R_L-(1-\alpha)R_H-V_R+\Delta V)$
(0,1)	$(\epsilon-\theta)(\alpha R_L+(1-\alpha)R_H)(-2R_C-C_R-V+\alpha R_L+(1-\alpha)R_H)$	$-(\theta-\epsilon)(\alpha R_L+(1-\alpha)R_H)+(-2R_C-C_R-V+\alpha R_L+(1-\alpha)R_H)$
(1,1)	$(2p-1)C_L(-2R_C-C_R-V+\alpha R_L+(1-\alpha)R_H)$	$-(2p-1)C_L+(2R_C+C_R+V-\alpha R_L-(1-\alpha)R_H)$
(x^*,y^*)	$\frac{(-2R_C-C_R-V+\alpha R_L+(1-\alpha)R_H+V_R-\Delta V)(2R_C+C_R+V-\alpha R_L-(1-\alpha)R_H)(\theta-\epsilon)(\alpha R_L+(1-\alpha)R_H)(1-2p)C_L}{(V_R-\Delta V)((\theta-\epsilon)(\alpha R_L+(1-\alpha)R_H)+C_L-2pC_L)}$	0

where

$$\frac{\partial F(x)}{\partial x} = (1-2x)((1-y)(\theta-\epsilon) \\ (\alpha R_L+(1-\alpha)R_H)-yC_L+2ypC_L) \tag{12}$$

$$\frac{\partial F(x)}{\partial y} = -x(1-x)((\theta-\epsilon) \\ (\alpha R_L+(1-\alpha)R_H)+C_L-2pC_L) \tag{13}$$

$$\frac{\partial F(y)}{\partial x} = y(1-y)(-V_R+\Delta V) \tag{14}$$

$$\frac{\partial F(y)}{\partial y} = (1-2y)(-2R_C-C_R-V \\ +\alpha R_L+(1-\alpha)R_H+(1-x)(V_R-\Delta V)). \tag{15}$$

The determinant and trace of the Jacobian matrix (denoted as detJ and trJ, respectively) are presented in Table III. In addition, an LEP qualifies as an ESS only if detJ is greater than 0 and trJ is less than 0 [24]. We substantiate the ESS status of the five identified LEPs in the subsequent analysis.

To streamline the mathematical representation of both detJ and trJ, we introduce the variable $a=(\theta-\epsilon)(\alpha R_L+(1-\alpha)R_H)$. This variable a encapsulates the differential in average returns for workers when contrasting the scenarios of reporting actual confidence levels versus overvalued confidence. Here, $\alpha R_L+(1-\alpha)R_H$ represents the worker's average return, while $\theta-\epsilon$ quantifies the disparity in the probability of selection by the requester under the conditions of genuine versus inflated confidence reporting. Additionally, we define $b=(2p-1)C_L$, representing the average loss incurred by the worker consequent to penalties applied during spot-checks. Furthermore, we define c and d as the partial derivatives of $F(y)$ with respect to y in points $(0,1)$ and $(1,0)$, respectively. More precisely, $c=2R_C+C_R+V-\alpha R_L-(1-\alpha)R_H-V_R+\Delta V$ is designated as the rate of change in the requester's expected return, considering the implementation of spot-checks. In this context, $2R_C+C_R$ denotes the direct expense associated with spot-checks, $V-\alpha R_L-(1-\alpha)R_H$ symbolizes the expected yield from the model trained by the worker, irrespective of the spot-check application, and $-V_R+\Delta V$ represents the marginal loss or gain relative to the spot-check. Conversely, $d=-2R_C-C_R-V+\alpha R_L+(1-\alpha)R_H$ delineates the rate of change in the requester's expected return, excluding the additional outcomes associated with spot-checks. Table IV succinctly summarizes these introduced notations for ease of reference and clarity.

TABLE IV
NOTATION INTRODUCED TO STREAMLINE DETJ AND TRJ

Symbol	Formula
a	$(\theta-\epsilon)(\alpha R_L+(1-\alpha)R_H)$
b	$(2p-1)C_L$
c	$2R_C+C_R+V-\alpha R_L-(1-\alpha)R_H-V_R+\Delta V$
d	$-2R_C-C_R-V+\alpha R_L+(1-\alpha)R_H$

TABLE V
SIMPLIFIED EXPRESSION OF DETJ AND TRJ

Equilibrium Point	detJ	trJ
(0,0)	$-ac$	$a-c$
(0,1)	bc	$b+c$
(1,0)	$-ad$	$-a+d$
(1,1)	bd	$-b-d$
(x^*,y^*)	$\frac{-cdab}{-(c+d)(a-b)}$	0

Then, with the introduced variables a, b, c, and d, we can simplify the representation of (x^*,y^*) as $x^*=(c/c+d)>0$ and $y^*=(a/a-b)$. In addition, the simplified expression of the detJ and trJ of the five LEPs are summarized in Table V.

Specifically, in the analytical model, $a<0$ signifies a scenario where the worker's expected average return is higher when they report an overvalued confidence level as opposed to their actual confidence, whereas $a>0$ indicates a lower average return for workers when reporting overvalued confidence compared to genuine confidence; $b>0$ denotes that the probability of obtaining an accurate result from a spot-check p exceeds 0.5, whereas $b<0$ indicates that the probability of obtaining an accurate result from a spot-check p is below 0.5. Since for our proposed spot-check mechanism to be feasible, the requestor should be able to obtain an accurate result from spot-check, we exclude the discussion of the scenario $b<0$ in the following discussion; $c>0$ indicates an accelerating rate of change in the requester's expected return when the strategy of performing spot-checks is considered, whereas $c<0$ denotes the rate at which the requestor's expected return changes when considering spot-checks is decreasing; and $d>0$ implies that the rate of change in the requester's expected return without considering the extra outcome of spot-check is also increasing. $d<0$ indicates a decelerating rate of change in the requester's

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

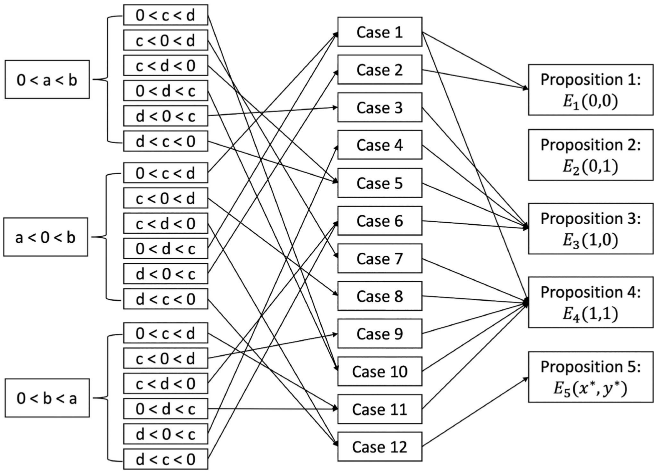


Fig. 3. Classification of different scenarios into five LEPs.

expected return in scenarios where spot-checks are not considered. Furthermore, considering all possible combinations of different conditions of a, b, c, and d, we formulate 12 cases and map them to the identified five LEPs. The detailed categorization is shown in Fig. 3. Then, we formulate the identified five LEPs as five propositions and analyze the ESS of the system in details.

Proposition 1: When $a < 0$ and $c > 0$, $E_1(0,0)$ is an ESS where the requestor chooses normal training while the worker chooses to report overvalued confidence.

Proof: This ESS is related to case 1 and case 2. □

Case 1: When $a < 0 < b$, $c > 0$, and $d > 0$

In this case, given the conditions $c > 0$ and $d > 0$, we can deduce that $c < c + d$, which implies $0 < x^* < 1$ with $x^* = (c/c + d)$. For $y^* = (a/a - b)$, considering the condition $a < 0 < b$, we can get $a - b < 0$ and $a < (a - b)$, which leads to the conclusion $0 < y^* < 1$. Therefore, we conclude (x^*, y^*) exists and there are totally five LEPs: $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$, and (x^*, y^*). Table VI (Case 1) shows the local equilibrium stability of these five points.

Case 2: When $a < 0 < b$, and $d < 0 < c$

In the proposed model, the parameters c and d play pivotal roles in determining the existence of the equilibrium point (x^*, y^*). Given the condition $d < 0 < c, c + d = \Delta V - V_R < 0$ implies that the differential in the model's value, as perceived by the requester when the worker reports overvalued confidence, is greater than the value loss incurred in identifying a malicious worker. In this case, $x^* = (c/c + d) < 0$, which results in $x^* < 0$ that contradicts the probabilistic nature of x^* as it should logically be within the range $[0, 1]$. Hence, under these conditions, the equilibrium point (x^*, y^*) does not exist. On the other hand, $d < 0$ implies $c > c + d$, leading to $x^* > 1$. This result is also inconsistent with the probabilistic boundaries of x^*. Therefore, the equilibrium point (x^*, y^*) is non-existent in this scenario as well. As a result, the model restricts the set of potential equilibrium points to four $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Table VI (Case 2) analyzes the local equilibrium stability of the four points respectively.

Proposition 2: $E_2(0,1)$ is not an ESS where the requestor chooses spot-check while the worker chooses to report overvalued confidence.

Proof: Given $b > 0$, it is mathematically impossible for both $bc > 0$ and $b + c < 0$ to occur concurrently. Furthermore, $b > 0$ indicates that the requester is correct in over half of the spot-checks, making it strategically unsound for the worker to report overvalued confidence when spot-checks are consistently conducted. Therefore, under these conditions, $E_2(0,1)$ cannot be an ESS. □

Proposition 3: When $a > 0$ and $d < 0$, $E_3(1,0)$ is an ESS where the requestor chooses normal training while the worker chooses to report real confidence.

Proof: This ESS is related to cases 3-6. □

Case 3: When $0 < a < b$, and $d < 0 < c$

Given that $0 < a < b$, it follows that $a - b < 0$. Consequently, $y^* = (a/a - b) < 0$, which contradicts the identified range of y^*. As analyzed in Case 2, $d < 0 < c$ implies that the identified range of x^* is contradicted. Therefore, the only viable points are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Table VI (Case 3) provides an analysis of the local equilibrium stability for these four points.

Case 4: When $0 < b < a$, and $d < 0 < c$

Given $0 < b < a$, it follows that $a - b > 0$ and $a > a - b$, leading to $y^* > 1$, which does no't satisfy the definition of y^*. Similarly, as analyzed in Case 2, $d < 0 < c$ implies that the identified range of x^* is contradicted, implying that (x^*, y^*) does not exist. Therefore, only four points are viable: $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Table VI (Case 4) examines the local equilibrium stability of these four points.

Case 5: When $0 < a < b$, $c < 0$, and $d < 0$

As analyzed in Case 3, $0 < a < b$ implies $y^* < 0$, which contradicts the identified range of y^*. Therefore, only the points $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ are considered. Table VI (Case 5) is used to analyze the local equilibrium stability of these four points.

Case 6: When $0 < b < a$, $c < 0$, and $d < 0$

Given that $c < 0$ and $d < 0$, it follows that $c + d < 0$ and $c < c + d$, which leads to $0 < x^* < 1$. However, as analyzed in Case 4, $0 < b < a$ lead to $y^* > 1$, which contradicts the identified range of y^*. Consequently, (x^*, y^*) does not exist. Therefore, we only consider the points $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Table VI (Case 6) is utilized to analyze the local equilibrium stability of these four points.

Proposition 4: When $b > 0$ and $d > 0$, $E_4(1,1)$ is an ESS where the requestor chooses spot-check while the worker chooses to report real confidence.

Proof: This ESS is related to cases 1, 7–11. □

Case 1: According to previous analysis, $E_4(1,1)$ is an ESS in Case 1.

Case 7: When $0 < a < b$, and $c < 0 < d$

As analyzed in Case 3, $0 < a < b$ implies $y^* < 0$, which contradicts the identified range of y^*. Therefore, the only feasible points are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Table VI (Case 7) examines the local equilibrium stability of these four points.

Case 8: When $a < 0 < b$, and $c < 0 < d$ as analyzed in Case 1, $a < 0 < b$ implies $0 < y^* < 1$ and y^* is valid. From

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VI
STABILITY ANALYSIS FOR VARIOUS CASES

Case 1: $a < 0 < b, c > 0, d > 0$				**Case 2:** $a < 0 < b, d < 0 < c$				**Case 3:** $0 < a < b, d < 0 < c$				**Case 4:** $0 < b < a, d < 0 < c$			
	detJ	trJ	Result		detJ	trJ	Result		detJ	trJ	Result		detJ	trJ	Result
$(0,0)$	$+$	$-$	Stable	$(0,0)$	$+$	$-$	Stable	$(0,0)$	$-$	$\pm$	Saddle	$(0,0)$	$-$	$\pm$	Saddle
$(0,1)$	$+$	$+$	Unstable	$(0,1)$	$+$	$+$	Unstable	$(0,1)$	$+$	$+$	Unstable	$(0,1)$	$+$	$+$	Unstable
$(1,0)$	$+$	$+$	Unstable	$(1,0)$	$-$	$\pm$	Saddle	$(1,0)$	$+$	$-$	Stable	$(1,0)$	$+$	$-$	Stable
$(1,1)$	$+$	$-$	Stable	$(1,1)$	$-$	$\pm$	Saddle	$(1,1)$	$-$	$\pm$	Saddle	$(1,1)$	$-$	$\pm$	Saddle
(x^*,y^*)	$+$	0	Center												
Case 5: $0 < a < b, c < 0, d < 0$				**Case 6:** $0 < b < a, c < 0, d < 0$				**Case 7:** $0 < a < b, c < 0 < d$				**Case 8:** $a < 0 < b, c < 0 < d$			
	detJ	trJ	Result		detJ	trJ	Result		detJ	trJ	Result		detJ	trJ	Result
$(0,0)$	$+$	$+$	Unstable	$(0,0)$	$+$	$+$	Unstable	$(0,0)$	$+$	$+$	Unstable	$(0,0)$	$-$	$\pm$	Saddle
$(0,1)$	$-$	$\pm$	Saddle	$(0,1)$	$-$	$\pm$	Saddle	$(0,1)$	$-$	$\pm$	Saddle	$(0,1)$	$-$	$\pm$	Saddle
$(1,0)$	$+$	$-$	Stable	$(1,0)$	$+$	$-$	Stable	$(1,0)$	$-$	$\pm$	Saddle	$(1,0)$	$+$	$+$	Unstable
$(1,1)$	$-$	$\pm$	Saddle	$(1,1)$	$-$	$\pm$	Saddle	$(1,1)$	$+$	$-$	Stable	$(1,1)$	$+$	$-$	Stable
Case 9: $0 < b < a, c < 0 < d$				**Case 10:** $0 < a < b, c > 0, d > 0$				**Case 11:** $0 < b < a, c > 0, d > 0$				**Case 12:** $a < 0$, $b < 0$, $c < 0$, $d < 0$			
	detJ	trJ	Result		detJ	trJ	Result		detJ	trJ	Result		detJ	trJ	Result
$(0,0)$	$-$	$\pm$	Saddle	$(0,0)$	$-$	$\pm$	Saddle	$(0,0)$	$-$	$\pm$	Saddle	(0,0)	-	$\pm$	Saddle
$(0,1)$	$-$	$\pm$	Saddle	$(0,1)$	$+$	$+$	Unstable	$(0,1)$	$+$	$+$	Unstable	(0,1)	-	$\pm$	Saddle
$(1,0)$	$+$	$+$	Unstable	$(1,0)$	$-$	$\pm$	Saddle	$(1,0)$	$-$	$\pm$	Saddle	(1,0)	-	$\pm$	Saddle
$(1,1)$	$+$	$-$	Stable	$(1,1)$	$+$	$-$	Stable	$(1,1)$	$+$	$-$	Stable	(1,1)	-	+	Saddle
								(x^*,y^*)	$-$	0	Saddle				

$c < 0 < d$, we can derive that $c + d > 0$ and $x^* < 0$, which contradicts the identified range of x^*. (x^*, y^*) is not an ESS. When $c + d < 0$, we can derive $|c| > |c + d|$, which leads to $x^* > 1$, also indicates (x^*, y^*) is not an ESS. Therefore, only the points $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ are viable. Table VI (Case 8) analyzes the local equilibrium stability of these points.

Case 9: When $0 < b < a$, and $c < 0 < d$

As analyzed in Case 8, when $c < 0 < d$, (x^*, y^*) is not an ESS. As analyzed in Case 4, $0 < b < a$ lead to $y^* > 1$, which contradicts the identified range of y^*, Thus, the only feasible points are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Table VI (Case 9) examines the local equilibrium stability of these points.

Case 10: When $0 < a < b$, $c > 0$, and $d > 0$

As analyzed in Case 3, $0 < a < b$ implies $y^* < 0$, which contradicts the identified range of y^*. Therefore, the only relevant points are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Table VI (Case 10) is dedicated to analyzing the local equilibrium stability of these four points.

Case 11: When $0 < b < a$, $c > 0$, and $d > 0$ as analyzed in Case 1, $c > 0$, and $d > 0$ leads to $0 < x^* < 1$. However, as analyzed in Case 4, $0 < b < a$ lead to $y^* > 1$, which contradicts the identified range of y^*. Consequently, the only valid points are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Table VI (Case 11) provides an analysis of the local equilibrium stability for these points.

Proposition 5: When $a < 0 < b$, $c < 0$ and $d < 0$, $E_5(x^*, y^*)$ is an ESS where the requestor and the worker will reach a mixed equilibrium point.

Proof: This ESS is related to Cases 1, 11–12. □

Case 1: According to previous analysis, $E_5(x^*, y^*)$ is an center point in Case 1.

Case 11: According to previous analysis, $E_5(x^*, y^*)$ does not exist in Case 11.

Case 12: When $a < 0 < b$, $c < 0$, and $d < 0$

As analyzed in Case 1, $a < 0 < b$ confirms $0 < y^* < 1$. As analyzed in Case 6, $c < 0$ and $d < 0$ leads to $0 < x^* < 1$. Hence, (x^*, y^*) exists, leading to five LEPs: $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$, and (x^*, y^*).

Table VI reveals that $E_1(0,0)$, $E_2(0,1)$, $E_3(1,0)$, $E_4(1,1)$, and $E_5(x^*, y^*)$ are all saddle points within this case scenario, indicating instability. Therefore, $E_5(x^*, y^*)$ is not an ESS, disproving the proposition.

C. Summary

Out of the five LEPs $E_1(0,0)$, $E_2(0,1)$, $E_3(1,0)$, $E_4(1,1)$, and $E_5(x^*, y^*)$, we exclude the point $E_2(0,1)$ since we only discuss the scenario $b > 0$. Also, further analysis indicates that $E_5(x^*, y^*)$ is not an ESS, leaving $E_1(0,0)$, $E_3(1,0)$, and $E_4(1,1)$ as actual ESSs under specific conditions, as defined in the propositions: 1) For the state $E_1(0,0)$, the system is expected to evolve towards a scenario where workers report overvalued confidence while requestors conduct normal training. The ESS condition for this state is characterized by $a < 0$ and $c > 0$. Here, $a < 0$ implies that workers have a higher probability of being selected by the requestor when they report overvalued confidence. 2) The state $E_3(1,0)$ signifies a progression in the system where workers report real confidence and requestors conduct normal training. For this ESS, the condition is $a > 0$ and $d < 0$. Specifically, $a > 0$ indicates a higher likelihood of worker selection by the requestor when real confidence is reported, aligning with the ESS criteria for this scenario. 3) In the state $E_4(1,1)$, the system is expected to evolve towards workers reporting real confidence and requestors conducting spot-checks. This state meets ESS criteria when $b > 0$ and $d > 0$. Here, $b > 0$ indicates that the effectiveness of spot-checks is greater than 0.5, aligning with the conditions for an ESS in this scenario.

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VII
DEFAULT PARAMETERS FOR SIMULATION SETUP

	R_L	R_H	V	ΔV	C_R	R_C	V_R	C_T	C_E	C_L	α	θ	ϵ	p
$E_1(0,0)$	4	20	8	4	1	2	2	4	0.5	3	0.2	0.1	0.9	0.6
$E_3(1,0)$	3	6	5	2	1	2	6	4	1	50	0.5	0.5	0.4	0.6
$E_4(1,1)$	4	20	8	4	1	2	2	4	0.5	3	0.2	0.1	0.9	0.6
$E_5(x^*,y^*)$	4	17	8	4	1	3	6	4	0.5	3	0.2	0.1	0.9	0.6

V. NUMERICAL SIMULATION

This section simulates the dynamic decision-making between workers and requestors, focusing on how factors such as reward values and the accuracy of identifying malicious workers affect system equilibrium. We omit the analysis of LEP $E_2(0,1)$ (proposition 2), assuming the feasibility of the spot-check mechanism requires requestors to detect malicious workers with a probability greater than 0.5.

A. System Simulation

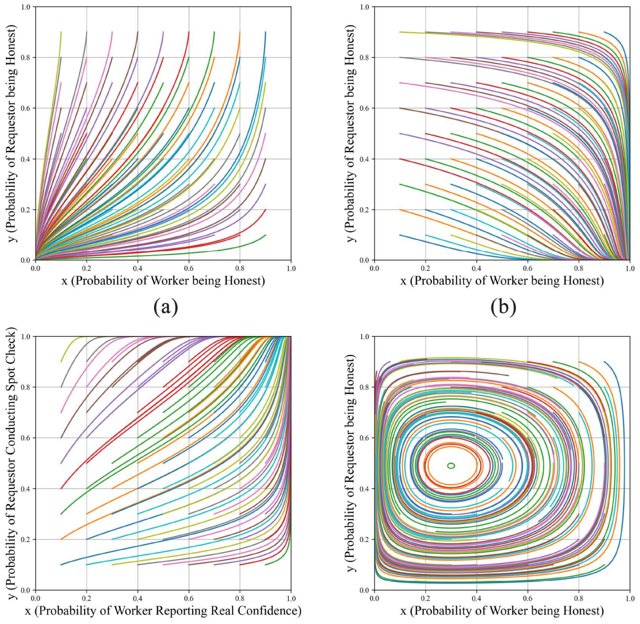
Following the default parameters in Table VII, the simulation results in four propositions are shown as follows:

Figs. 4(a) and 5(a) illustrate that the evolutionary model consistently evolves towards the stable point $E_1(0,0)$, starting from initial points $(0.1, 0.1)$ through $(0.9, 0.9)$. This result confirms that $E_1(0,0)$ is an ESS. The convergence of the evolutionary path towards $E_1(0,0)$ aligns with the theoretical model's analysis. Similarly, Figs. 4(b) and 5(b) demonstrate that the evolutionary model consistently gravitates towards the stable point $E_3(1,0)$, starting from initial points $(0.1, 0.1)$ to $(0.9, 0.9)$. The convergence of the evolutionary path towards $E_3(1,0)$ aligns with the analysis of the model, which identifies $E_3(1,0)$ as an ESS. Furthermore, Figs. 4(c) and 5(c) reveal that the evolutionary model consistently evolves towards $E_4(1,1)$, starting from initial points $(0.1, 0.1)$ to $(0.9, 0.9)$. Hence, $E_4(1,1)$ is established as an ESS, as analyzed using our theoretical model. On the other hand, Fig. 4(d) indicates $E_5(x^*, y^*)$ is not an ESS. The system's final state is not fixed but varies dynamically, influenced by the initial strategies of both requestors and workers, as illustrated in Fig. 5(d) with workers and requestors strategies starting from initial points $(0.1, 0.1)$ to $(0.9, 0.9)$. The evolutionary path depicted forms a continuous cycle without reaching a stable point, aligning with our model's analysis. This result highlights the complexity and variability inherent in this particular system configuration.

In summary, Fig. 4(a)–(c) reveal the dynamic evolution of strategies between requestors and workers, demonstrating that irrespective of their initial strategies, e.g., different choices of reward values (R_C) or spot-check rewards (R_C), the system ultimately converges to the corresponding ESS. In contrast, as depicted in Fig. 4(d), no ESS emerges. The final outcome here is fluid, influenced by the initial strategies of both requestors and workers, reflecting the system's inherent dynamism and variability.

B. Parameter Simulation

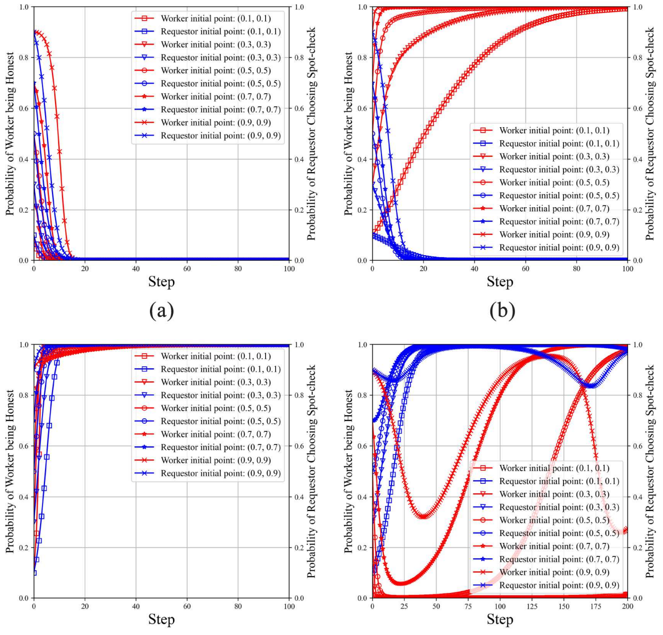
Furthermore, our study explores how different reward values (R_H), model values (V), and spot-check rewards (R_C) influence worker and requestor decision-making across scenarios. Omitting further analysis on the unstable $E_5(x^*, y^*)$, we consolidate $E_1(0,0)$ and $E_4(1,1)$ into Scenario 1 and designate $E_3(1,0)$ as Scenario 2. Simulations initiate with both workers' honesty and requestors' spot-check accuracy set at a probability of 0.5. This standard starting point, while keeping other variables fixed, allows the study to isolate and analyze the impact of



(a) (b)

(c) (d)

Fig. 4. Dynamic evolution of equilibrium points. (a) $E_1(0, 0)$. (b) $E_3(1, 0)$. (c) $E_4(1, 1)$. (d) $E_5(x^*, y^*)$.



(a) (b)

(c) (d)

Fig. 5. Dynamic evolutionary paths with different initial points. (a) $E_1(0, 0)$. (b) $E_3(1, 0)$. (c) $E_4(1, 1)$. (d) $E_5(x^*, y^*)$.

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

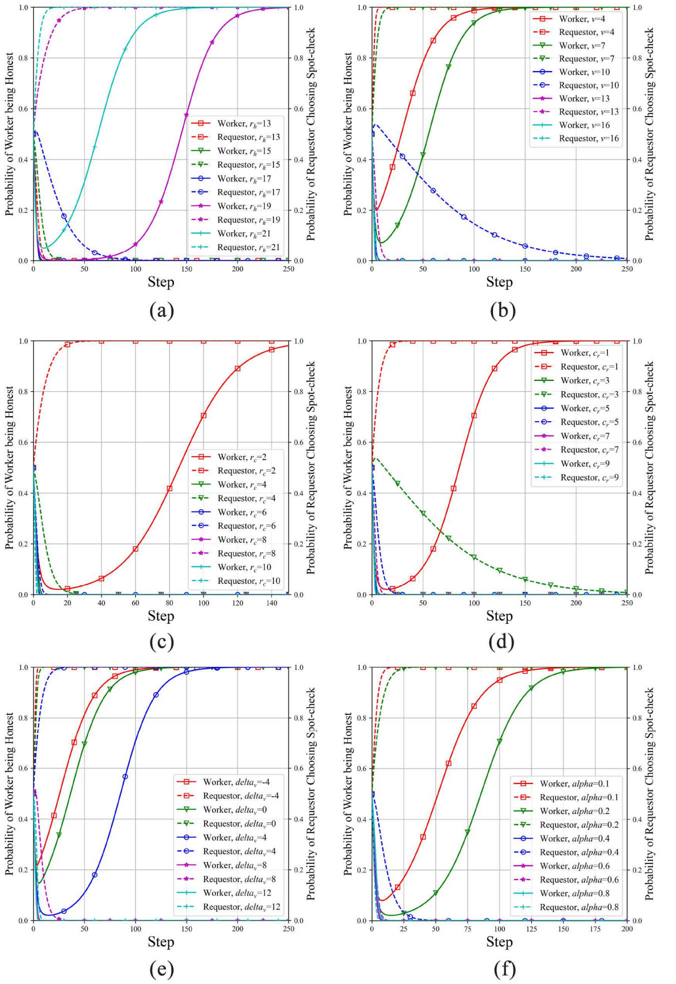


Fig. 6. Sensitivity analysis of parameters for Scenario 1. (a) Sensitivity of R_H. (b) Sensitivity of V. (c) Sensitivity of R_C. (d) Sensitivity of C_R. (e) Sensitivity of ΔV. (f) Sensitivity of R_L.

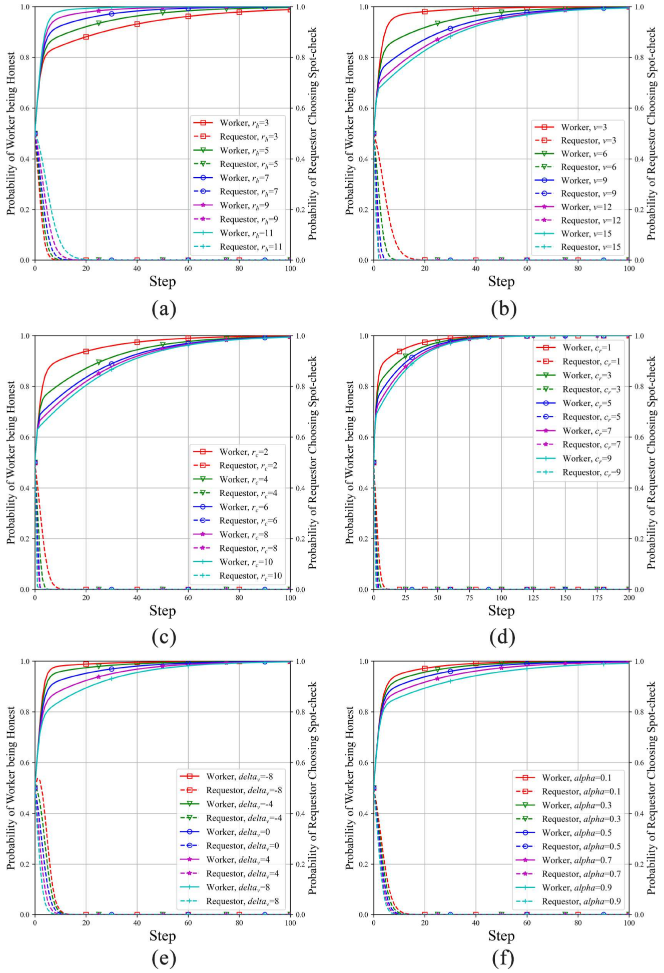


Fig. 7. Sensitivity analysis of parameters for scenario 2. (a) Sensitivity of R_H. (b) Sensitivity of V. (c) Sensitivity of R_C. (d) Sensitivity of C_R. (e) Sensitivity of ΔV. (f) Sensitivity of R_L.

distinct factors on the system's dynamics. Results for Scenario 1 and 2 are depicted in Figs. 6 and 7, respectively.

Fig. 6(a) and 6(b) shows how the system's ESSs change with different R_H (valued at 13, 15, 17, 19, 21) and V (valued at 4, 7, 10, 13, 16), respectively. When $R_H > 17$ or $V < 7$, the system gravitates to a strategy of overvalued confidence reporting with more spot-checks. Conversely, with $R_H \leq 17$ or $V > 7$, the requestor tends to perform more normal training. When $V > 7$, the worker tends to be honest. Thus, High rewards incentivize requestors towards spot-checks to mitigate worker dishonesty, nudging workers towards honesty to avoid penalties. Higher V values expedite the requestor's transition to normal training, suggesting that a sufficiently high model value from honest workers encourages requestors to adopt this strategy.

Fig. 6(c) and 6(d) shows the impact of varying R_C (valued at 2, 4, 6, 8, 10) and C_R (valued at 1, 3, 5, 7, and 9). For $R_C > 2$ or $C_R > 3$, workers quickly shift towards reporting overvalued confidence, while the requestor would perform less spot-checks with higher R_C and predominantly opt for normal training with higher C_R. Conversely, with $R_C \leq 2$ or $C_R < 3$, workers are more inclined to report real confidence, prompting requestors to favor spot-checks. Essentially, a high spot-check reward or communication costs discourage requestors from conducting checks, prompting workers towards overvalued confidence reporting.

Fig. 6(e) and 6(f) shows the impact of different ΔV (set to -4, 0, 4, 8, and 12) and α (set to 0.1, 0.2, 0.4, 0.6, and 0.8), respectively. With $\Delta V > 4$ or $\alpha \geq 0.4$, workers tend to adopt overvalued confidence reporting while the requestor tends to shift to normal training swiftly. With $\Delta V \leq 4$ or $\alpha < 0.4$, a smaller ΔV or α accelerates the system's evolution towards real confidence reporting from workers and more spot-checks from requestors. This suggests that when the difference in the model value is small or negative, the requestor tends to perform more spot-checks, while a larger or positive value above four discourages them. Moreover, a lower likelihood of low rewards prompts requestors towards spot-checks, while a higher likelihood encourages avoidance of spot-checks.

Fig. 7(a) illustrates the response of workers and the requestor to varying R_H values (set at 3, 5, 7, 9, and 11). An increase in R_H leads to a quicker evolution of worker honesty and a reduced tendency of the requestor to perform spot-checks.

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

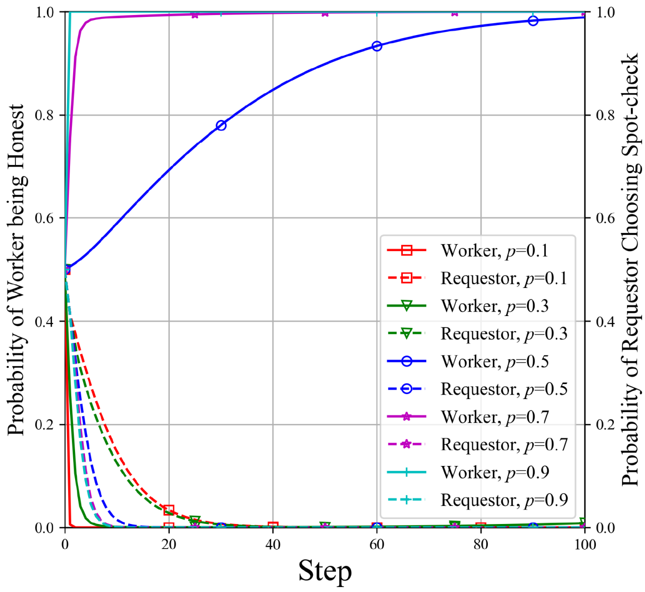


Fig. 8. Sensitivity of the probability p that requestors make correct judgment in spot-checks.

Similarly, Fig. 7(b) demonstrates the impact of different V values (set at 3, 6, 9, 12, and 15). As the value V of the model trained by an honest worker increases, both worker honesty and the requestor's reduced inclination for spot-checks are observed. Additionally, Fig. 7(c), setting the value of R_C as 2, 4, 6, 8, and 10, reveals that a higher R_C encourages workers evolving towards honesty and requestors less likely to engage in spot-checks.

Fig. 7(d)–(f) shows the impact of different C_R (set to 1, 3, 5, 7, and 9), ΔV (set to -8, -4, 0, 4, and 8), and α (set to 0.1, 0.3, 0.5, 0.7, and 0.9), respectively. Higher C_R, ΔV or α encourages the requestor to perform more normal training. Meanwhile, workers are more inclined to report overvalued confidence as C_R rises. On the other hand, an increase in ΔV or α slows down the evolution towards workers reporting real confidence.

Furthermore, Fig. 8 presents the reaction of workers and the requestor to varying p values, which represent the probability of correctly identifying a malicious worker. When p is below 0.5, there is a tendency for workers to behave maliciously and for the requestor to engage in spot-checks more frequently. At p equal to 0.5, both workers and the requestor take longer to reach the equilibrium point $E_3(1, 0)$. Conversely, when p exceeds 0.5, the time taken to evolve to the equilibrium point decreases. Thus, as p increases from 0.5, workers are more likely to act honestly, and requestors are less inclined to perform spot-checks. This finding confirms that, for our proposed spot-check mechanism to be feasible, the requestor should be able to obtain an accurate result from spot-checks.

C. Discussion

The simulation results offer critical insights into how various factors, such as reward levels and the likelihood of detecting malicious behavior, influence the system's balance. Notably, the precision of the requestor's spot-checks plays a pivotal role. An error rate above 0.5, indicating frequent misjudgments regarding worker honesty, may drive the system towards an unstable equilibrium characterized by inflated confidence reports from workers and frequent spot-checks by requestors. This scenario leads to escalated costs for requestors and an increased risk of dishonest workers being erroneously deemed honest, thereby impairing the system's overall efficiency and effectiveness.

Furthermore, the implementation of regular spot-checks by the requestor is crucial for the reputation system's efficacy. Typically, the requestor assigns tasks to the highest-ranked worker, whose ranking is influenced by their reputation score multiplied by their self-assessed confidence level. The adjustment of worker reputation scores following spot-checks ensures that those accurately reporting their confidence are more likely to be chosen for standard training tasks, thus effectively separating them from those who inflate their confidence levels. Consequently, this mechanism allows for a more precise ranking of workers, fostering a balanced ecosystem within the system.

VI. Conclusion

This work pioneers the integration of a reputation system coupled with a spot-check mechanism, aimed at fostering socio-ethical integrity among workers within the context of decentralized knowledge inference for knowledge graphs. Empirical findings underscore the significance of effectively detecting socially deviant behaviors among workers and the role of incentive structures in promoting system-wide integrity. Specifically, we observe that higher probabilities of identifying malicious workers and offering increased rewards are instrumental in accelerating the evolution toward an equilibrium. These insights contribute to our understanding of how to optimize strategic interactions within decentralized knowledge inference ecosystems to ensure their success, sustainability, and equitable knowledge sharing in real-world applications.

References

[1] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine learning for industrial applications: A comprehensive literature review," *Expert Syst. Appl.*, vol. 175, 2021, Article no. 114820.

[2] Y. Han and J. Tao, "Revolutionizing pharma: Unveiling the AI and LLM trends in the pharmaceutical industry," 2024, *arXiv:2401.10273*.

[3] O. Ogundare, G. Q. Araya, I. Akrotirianakis, and A. Shukla, "Resiliency analysis of LLM generated models for industrial automation," 2023, *arXiv:2308.12129*.

[4] Y. Chi, Y. Dong, Z. J. Wang, F. R. Yu, and V. C. M. Leung, "Knowledge-based fault diagnosis in industrial internet of things: A survey," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 12886–12900, 2022.

[5] K. Zanbouri, M. Darbandi, M. Nassr, A. Heidari, N. Navimipour, and S. Yalcin, "A GSO-based multi-objective technique for performance optimization of blockchain-based industrial internet of things,"*Int. J. Commun. Syst.*, vol. 7, pp. 1–22, 2024.

[6] M. Chen et al., "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021.

[7] T. Fan et al., "Fate-LLM: A industrial grade federated learning framework for large language models," 2023, *arXiv:2310.10049*.

[8] C. Li, X. Qu, and Y. Guo, "TFcrowd: A blockchain-based crowdsourcing framework with enhanced trustworthiness and fairness," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–20, 2021.

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[9] F. Moiseev, Z. Dong, E. Alfonseca, and M. Jaggi, “SKILL: Structured knowledge infusion for large language models,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Human Lang. Technol.*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds., Jul. 2022, pp. 1581–1588.

[10] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, “Unifying large language models and knowledge graphs: A roadmap,” 2023, *arXiv:2306.08302*.

[11] R. Logan, N. F. Liu, M. E. Peters, M. Gardner, and S. Singh, “Barack’s wife hillary: Using knowledge-graphs for fact-aware language modeling,” in *Proc. 57th Ann. Meeting Association. Comput. Linguistics*, Florence, Italy, 2019.

[12] Y. Chi, H. Duan, W. Cai, Z. J. Wang, and V. C. M. Leung, “Knowledge inference over web 3.0 for intelligent fault diagnosis in industrial internet of things,” *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 5, pp. 3955–3968, Sep./Oct. 2024.

[13] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, Dec. 2013, pp. 2787–2795.

[14] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 28, no. 1, Jun. 2014.

[15] N. Lao and W. W. Cohen, “Relational retrieval using a combination of path-constrained random walks,” *Mach. Learn.*, vol. 81, no. 1, pp. 53–67, Jul. 2010.

[16] N. Lao, T. Mitchell, and W. W. Cohen, “Random walk inference and learning in a large scale knowledge base,” in *Proc. Conf. Empirical Methods in Natural Lang. Process.*, Jul. 2011, pp. 529–539.

[17] N. Lao, A. Subramanya, F. Pereira, and W. W. Cohen, “Reading the web with learned syntactic-semantic inference rules,” in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, Jul. 2012, pp. 1017–1026.

[18] X. Wang, X. Ren, C. Qiu, Z. Xiong, H. Yao, and V. C. M. Leung, “Integrating edge intelligence and blockchain: What, why, and how,” *IEEE Commun. Surv. Tut.*, vol. 24, no. 4, pp. 2193–2229, Fourthquarter 2022.

[19] W. Zhang, Z. Wang, and X. Li, “Blockchain-based decentralized federated transfer learning methodology for collaborative machinery fault diagnosis,” *Rel. Eng. Syst. Saf.*, vol. 229, 2023, Article no. 108885.

[20] Arash Heidari, Nima Jafari Navimipour and M. Unal, “A novel blockchain-based deepfake detection method using federated and deep learning models,” *IEEE Trans. Cloud Comput.*, vol. 16, p. 1073–1091, 2024.

[21] 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, Feb. 2021.

[22] X. Wang, Y. Zhao, C. Qiu, Z. Liu, J. Nie, and V. C. M. Leung, “Infedge: A blockchain-based incentive mechanism in hierarchical federated learning for end-edge-cloud communications,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3325–3342, Dec. 2022.

[23] H. Yu, H.-Y. Chen, S. Lee, S. Vishwanath, X. Zheng, and C. Julien, “IDML: Incentivized decentralized machine learning,” 2023, *arXiv:2304.05354*.

[24] D. Friedman, “Evolutionary games in economics,” *Econometrica: J. Econometric Soc.*, vol. 59, no. 3, pp. 637–666, May 1991.

[25] K. Fan and E. C. Hui, “Evolutionary game theory analysis for understanding the decision-making mechanisms of governments and developers on green building incentives,” *Building Environ.*, vol. 179, 2020, Article no. 106972.

[26] X. Bai, Y. Ye, T. Chen, and N. Xie, “The evolutionary game of emotions considering the influence of reputation,” *Appl. Math. Computation*, vol. 474, 2024, Article no. 128709.

[27] M.-H. Chen, H. Wei, M. Wei, H. Huang, and C.-H. J. Su, “Modeling a green supply chain in the hotel industry: An evolutionary game theory approach,” *Int. J. Hospitality Manage.*, vol. 92, 2021, Article no. 102716.

[28] Z. Wang, Q. Wang, B. Chen, and Y. Wang, “Evolutionary game analysis on behavioral strategies of multiple stakeholders in e-waste recycling industry,” *Resour. Conservation Recycling*, vol. 155, 2020, Article no. 104618.

[29] H. Mahini, A. M. Rahmani, and S. M. Mousavirad, “An evolutionary game approach to iot task offloading in fog-cloud computing,” *J. Supercomputing*, vol. 77, pp. 5398–5425, 2021.

[30] Y. Zhao, K. Zheng, J. Guo, B. Yang, T. B. Pedersen, and C. S. Jensen, “Fairness-aware task assignment in spatial crowdsourcing: Game-theoretic approaches,” in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*. Piscataway, NJ, USA: IEEE Press, 2021, pp. 265–276.

[31] S. Motepalli and H.-A. Jacobsen, “Reward mechanism for blockchains using evolutionary game theory,” in *Proc. 3rd Conf. Blockchain Res. & Appl. Innovative Networks Services (BRAINS)*. Piscataway, NJ, USA: IEEE Press, 2021, pp. 217–224.

[32] T. Mai, H. Yao, N. Zhang, L. Xu, M. Guizani, and S. Guo, “Cloud mining pool aided blockchain-enabled internet of things: An evolutionary game approach,” *IEEE Trans. Cloud Comput.*, vol. 11, no. 1, pp. 692–703, Jan. 2023.

[33] Y. Han, D. Niyato, C. Leung, and D. I. Kim, “Opportunistic coded distributed computing: An evolutionary game approach,” in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2021, pp. 1430–1435.

Yuanfang Chi (Student Member, IEEE) received the B.A.Sc. and M.A.Sc. degrees in electrical and computer engineering from The University of British Columbia (UBC), Vancouver, BC, Canada, in 2012 and 2015, respectively, where she is currently working toward the Ph.D. degree in electrical and computer engineering.

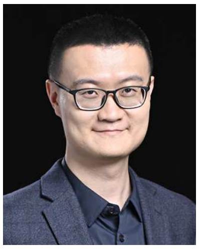
She is a visiting Ph.D. Student with the College of Computer Science and Software Engineering, Shenzhen University, China. Her research interests include distributed machine learning, knowledge graphs, fault diagnosis, and industrial Internet of Things.

Qiyue Zhang (Student Member, IEEE) received the B.Eng. in electronic information engineering from The Chinese University of Hong Kong, Shenzhen, China, in 2024. She is currently working toward the master’s degree in electrical and computer engineering with the University of California, San Diego, CA, USA.

Her research interests include Blockchain, Game Theory, and Network Security.

Jiaxiang Sun (Student Member, IEEE) received the B.Sc. degree in environmental science and the B.Sc. degree in computer science and technology from Peking University, China, in 2022. He is currently working toward the M.Phil. degree in computer and information engineering with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China, under the supervision of Prof. Wei Cai.

His research interests include blockchain and game theory.



Wei Cai (Senior Member, IEEE) received the B.Eng. degree in software engineering from Xiamen University, China, in 2008, the M.S. degree in electrical engineering and computer science from Seoul National University, Korea, in 2011, and the Ph.D. degree in electrical and computer engineering from The University of British Columbia (UBC), Vancouver, BC, Canada, in 2016.

From 2016 to 2018, he was a Postdoctoral Research Fellow with UBC. Currently, he is an Assistant Professor in computer engineering with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen. He has co-authored more than 100 journals and conference papers in the areas of distributed/decentralized systems and interactive multimedia. His research interests include the topic of human-centered computing for metaverse, including blockchain, Web3, digital games, social computing, human–computer interaction, and computational art.

Dr. Cai serves as an Associate Editor of IEEE Transactions on Computational Social Systems (IEEE TCSS), IEEE Transactions on Cloud Computing (IEEE TCC), and *ACM Transactions on Multimedia Computing, Communications and Applications* (ACM TOMM). He was a Program Co-Chair for ACM NOSSDAV’23, the Open-Source Software Competition Chair for ACM MM’23, and the Reproducibility Chair for ACM MMSys’23. He was a recipient of six best paper awards. He is a member of ACM.

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Z. Jane Wang (Fellow, IEEE) received the B.Sc. degree from Tsinghua University, Beijing, China, in 1996 and the M.Sc. and Ph.D. degrees from the University of Connecticut, Storrs-Mainsfield, Connecticut, USA, in 2000 and 2002, respectively, all in electrical engineering.

From 2002 to 2004, she was a Research Associate with the University of Maryland, College Park, Washington, D.C., USA. Since 2004, she has been with the ECE Department, The University of British Columbia, Vancouver, BC, Canada, where she is currently a Professor. Her research interests include the broad areas of statistical signal processing and machine learning.

Dr. Wang is a fellow of the Canadian Academy of Engineering.

Victor C. M. Leung (Life Fellow, IEEE) received the B.A.Sc. (Hons.) and the Ph.D. degrees in electrical engineering from The University of British Columbia, Vancouver, BC, Canada, in 1977 and 1982, respectively. He is a Distinguished Professor in computer science and software engineering with Shenzhen University, China. He is an Emeritus Professor in electrical and computer engineering and the Director of the Laboratory for Wireless Networks and Mobile Systems, The University of British Columbia (UBC). His research interests include the broad areas of wireless networks and mobile systems, and he has published widely in these areas.

Prof. Leung is serving on the editorial boards of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and several other journals. He received the 1977 APEBC Gold Medal, 1977–1981 NSERC Postgraduate Scholarships, IEEE Vancouver Section Centennial Award, 2011 UBC Killam Research Prize, 2017 Canadian Award for Telecommunications Research, 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award, and 2018 ACM MSWiM Reginald Fessenden Award. He co-authored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, 2017 IEEE Systems Journal Best Paper Award, 2018 IEEE CSIM Best Journal Paper Award, and 2019 IEEE TCGCC Best Journal Paper Award. He is named in the current Clarivate Analytics list of “Highly Cited Researchers.” He is a fellow of the Royal Society of Canada (Academy of Science), Canadian Academy of Engineering, and Engineering Institute of Canada.

Authorized licensed use limited to: University of Washington Libraries. Downloaded on November 30,2024 at 04:47:03 UTC from IEEE Xplore. Restrictions apply.