This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2023.3344516

Knowledge Inference over Web 3.0 for Intelligent Fault Diagnosis in Industrial Internet of Things

Yuanfang Chi, *Student Member, IEEE*, Haihan Duan, *Member, IEEE*, Wei Cai, *Senior Member, IEEE*, Z. Jane Wang, *Fellow, IEEE*, and Victor C. M. Leung, *Life Fellow, IEEE*

***Abstract*—Collaboration through knowledge sharing is critical for the success of intelligent fault diagnosis in a complex Industrial Internet of Things (IIoT) system that comprises various interconnected subsystems. However, since the subsystems of an IIoT system may be owned and operated by different stakeholders, sharing fault diagnosis knowledge while preserving data security and privacy is challenging. While decentralized data exchange has been proposed for cyber-physical systems and digital twins based on the Web 3.0 paradigm, decentralized knowledge sharing in knowledge-based intelligent fault diagnosis is less investigated. To address this research gap, we propose a Web 3.0 application for collaborative knowledge-based intelligent fault diagnosis using blockchain-empowered decentralized knowledge inference (BDKI). Our proposed mechanism enables workers to self-evaluate their ability to contribute to the knowledge inference with their local knowledge graphs. The knowledge-sharing requestor can then choose a worker with the best evaluation result and initiate collaborative training. To demonstrate the efficiency and effectiveness of BDKI, we evaluate it using well-known datasets. Results show that BDKI delivers a favorable inference model with higher overall accuracy and less training effort compared to inference models trained using conventional knowledge inference with random training sequences.**

***Index Terms*—Industrial Internet of Things, Fault Diagnosis, Decentralized Knowledge Inference, Web 3.0**

I. Introduction

Through the integration of computing, communication, and control, the Industrial Internet of Things (IIoT) establishes connections among machines, computers, and people, enabling intelligent industrial operations with heightened automation. This integration has ushered in a new era of safe, collaborative, robust, and efficient production across various industries. However, serious safety incidents might happen due to machine malfunctions, mistaken actions, or cyber-attacks. Consequently, ensuring the reliability of IIoT systems is imperative to safeguard industrial processes and maintain operational integrity. Among the various strategies employed to enhance IIoT system reliability, fault diagnosis has emerged as a focal point of research and innovation, drawing considerable attention from both academia and industry. Meanwhile, the advent of big data and machine learning has given impetus to the era of intelligent fault diagnosis. For example, using neural networks to differentiate faulty conditions from normal conditions and automatically spot early signs of equipment failure allows maintenance decisions to be optimized over different time horizons, such as weeks or months, to ensure timely and cost-efficient part procurement and/or maintenance personnel assignments. Therefore, intelligent fault diagnosis provides much safer and more efficient approaches to enhance the reliability of IIoT systems.

A key factor in the success of intelligent fault diagnosis is the comprehensive cognition of the overall system status and data completeness. Since complex IIoT systems have evolved to encompass decentralized and spatially distributed but interconnected subsystems, a failure observed in one subsystem may have underlying relations with other failures observed in another subsystem. Thus, fault detection and countermeasures may need to be taken across different subsystems to ensure the normal operation of the overall system. While each subsystem may have local measurements and local fault diagnosis models specifically designed and implemented to make fault diagnosis or maintenance decisions, the fault diagnosis result of the overall system should take into account of all relevant localized diagnosis results, such as those achieved through a consensus-based algorithm. Furthermore, manually designing simulation cases to capture all types of faults by any subsystem owner alone is a challenging task owing to the complexity and dynamism of the underlying subsystems or knowledge gaps. Thus collaboration through knowledge sharing is beneficial for more comprehensive intelligent fault diagnosis in IIoT systems [1].

However, collaborative efforts have brought about a series of privacy and security challenges, especially when IIoT subsystems are operated by different stakeholders. These challenges may be overcome by requiring the entire system to

This work was supported by the National Nature Science Foundation of China under Grant 62102266, the Guangdong Pearl River Talent Recruitment Program under Grant 2019ZT08X603, the Guangdong Pearl River Talent Plan under Grant 2019JC01X235, the Shenzhen Science and Technology Innovation Commission under Grants R2020A045 and KCXFZ20201221173411032, Shenzhen Science and Technology Program under Grant JCYJ20210324124205016, the Shenzhen Key Lab of Crowd Intelligence Empowered Low-Carbon Energy Network under Grant ZDSYS20220606100601002, and the Canadian Natural Sciences and Engineering Research Council. (*Corresponding authors: Wei Cai and Victor C. M. Leung*).

YuanFang Chi is a Ph.D. student in the Department of Electrical and Computer Engineering at the University of British Columbia, Vancouver, BC, V6T 1Z4, Canada and a visiting Ph.D. student in the College of Computer Science and Software Engineering at Shenzhen University, China (e-mail: yuanchi@ece.ubc.ca).

Haihan Duan and Wei Cai are with the School of Science and Engineering at The Chinese University of Hong Kong, Shenzhen, China (e-mail: haihanduan@link.cuhk.edu.cn, caiwei@cuhk.edu.cn).

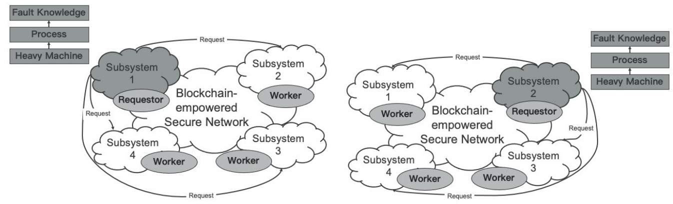
Z. Jane Wang is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: zjanew@ece.ubc.ca).

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

comply with the decentralized requirements of the Web 3.0 paradigm, which encompasses a series of technologies such as blockchain, consensus algorithms, and smart contracts to ensure that multiple parties participating in a system can collaborate without central control and mutual trust. The Web 3.0 paradigm has been proposed for data exchange among decentralized autonomous organizations to ensure security and privacy in cyber-physical systems and digital twins [2]–[4] residing in IIoT systems. Thus, we envision a decentralized intelligent fault diagnosis over Web 3.0 for IIoT systems. That is, after each subsystem has made its own fault diagnosis decisions using its own fault diagnosis knowledge, it may exchange its fault diagnosis knowledge with others through a transparent medium such as the blockchain, achieving a collaborative fault diagnosis that is more practical for future intelligent fault diagnosis in IIoT systems.

Nevertheless, the lack of interoperability between fault diagnosis techniques of different subsystems in IIoT systems due to heterogeneous data collected from different sensors, equipment, or industrial processes makes collaborative intelligent fault diagnosis challenging. To overcome this challenge, knowledge-based intelligent fault diagnosis has emerged as the most recent research trend, where collected fault diagnosis knowledge can be maintained in a universal knowledge base (KB). In particular, knowledge-based intelligent fault diagnosis that leverages a KB represented by knowledge graphs that is well known for their ability to handle highly heterogeneous data and ensure interoperability has become the state of the art [5], [6]. A knowledge inference method utilizing a distributed knowledge representation learning algorithm that embeds several knowledge graphs into a continuous vector space for knowledge inference was investigated in [7]. Knowledge can be shared by sharing the trained reasoning model that contains the vector space without directly sharing the data. However, in that study, knowledge representation learning was performed on carefully partitioned knowledge graphs, whereas individual fault knowledge graphs are usually constructed and maintained independently by each subsystem in IIoT systems. Thus, conventional distributed knowledge inference algorithms that focus on dataset decomposition and parameter aggregation are insufficient for accomplishing distributed knowledge inference for intelligent fault diagnosis in IIoT systems. The authors in [8] proposed a distributed knowledge inference framework that overcomes this obstacle. The proposed framework uses a centralized coordinator to handle the distributed training, thereby allowing participants to train a reasoning model with their local knowledge graphs continuously without any dataset manipulation. However, the proposed framework still relies on a central control unit, and the trustworthiness of the central coordinator could become a key concern among diverse stakeholders in IIoT systems to adopt the proposed framework. To our best knowledge, studies on decentralized knowledge inference over Web 3.0 are still lacking in the literature.

To bridge the research gap identified above, we extend the aforementioned distributed knowledge inference framework [8] by proposing a Web 3.0 application that incorporates a blockchain-empowered decentralized knowledge inference (BDKI) mechanism for intelligent fault diagnosis in IIoT systems. In our proposal, after one participant has requested a knowledge sharing task as the requestor, other participants may help complete the knowledge sharing task with their local knowledge graphs as workers. Since more than one worker may have valuable knowledge in their local knowledge graphs as illustrated in Fig.1, we formulate the BDKI paradigm as an iterative collaboration process. Specifically, when a knowledge-sharing request is published through the *request matching interface* by a knowledge sharing requestor, a group of workers would evaluate their ability to contribute to the request with a newly proposed task evaluation function. The requestor then chooses one worker with the best evaluation result and initiates a round of training. The requestor may attempt to further improve its reasoning model by picking a worker with the second-best evaluation result and initiating another round of training. The process continues until the requestor is satisfied or all the workers have joined the iterative knowledge inference process once. Since there is no central control in the proposed mechanism, the blockchain-based system ensures openness, scalability, anonymity, security, and reliability for the knowledge sharing of intelligent fault diagnosis in IIoT systems.



(a) (b)

Fig. 1. Collaborative knowledge sharing among participants

We conducted empirical studies to prove that the proposed BDKI mechanism can deliver a reasoning model that outperforms most reasoning models trained by the distributed reasoning method with random training sequences proposed in [8]. Specifically, we use a fault knowledge graph of a real industrial process, namely the Tennessee Eastman (TE) process, to show that the proposed mechanism can work with practical IIoT systems. Additionally, we use FB15K-237 and WN18RR datasets to show that the proposed mechanism can scale up to large-scale knowledge graphs. Furthermore, we use the WN18RR dataset to show that the proposed mechanism can deliver a reasoning model that has a high overall accuracy with less training effort. In summary, the contributions of this work are as follows:

- To the best of our knowledge, this is the first study to propose a BDKI mechanism for intelligent fault diagnosis in IIoT systems.
- We present a novel Web 3.0 application for decentralized knowledge-based intelligent fault diagnosis, with a task evaluation function that provides usable and practical references for the workers to estimate their possible contributions, enabling the requestor to select a worker for collaborative training according to the evaluation results. This application can be be practically realized for modern IIoT systems in general, especially those with

diverse stakeholders;

- With empirical study, we show that the proposed BDKI mechanism is efficient and effective. Therefore, the proposed Web 3.0 application is both useful and beneficial for distributed IIoT systems.

The rest of this paper is organized as follows. Related works are reviewed in Section II. The proposed model and methodology are described in Section III. Evaluation results are presented in Section IV.

II. Related Work

A. Knowledge-based Intelligent Fault Diagnosis in IIoT Systems

Measured signals (e.g., vibration, noise, or pressure) of mechanical components can be analyzed and compared to prior knowledge obtained from healthy systems to identify faulty symptoms [9]. Also, the characteristics of a system can be learned through machine learning algorithms. For example, authors in [10] constructed a classification and regression tree to find the deciding thresholds of the features to diagnose faults in a variable refrigerant flow system. Moreover, studies have shown that different maintenance requirements need to be analyzed from different kinds of signals, and maintenance actions could be determined with different maintenance expectations, such as weeks or months before predicted failure. For instance, a multiple classifier approach is proposed in [11] to identify integral type faults from machine failures due to wear and tear effects of usage and stress on equipment parts. Different maintenance management results are assigned to different classifiers, such as SVM or K-nearest neighbor, to identify maintenance requirements and minimize expected costs. Intelligent fault diagnosis is found to be a cost-effective and compelling approach to ensure the reliability of IIoT systems.

However, machine learning algorithms need to be carefully designed for different fault types of different components. To overcome this obstacle and accomplish fault diagnosis at the system level, knowledge-based intelligent fault diagnosis has become the state of the art for IIoT systems [1]. Whereas data-driven approaches such as machine learning approaches can detect and locate component failures, knowledge-based intelligent fault diagnosis is particularly well suited for complex or multi-element systems/processes for which detailed mathematical models are not available. Typically, a knowledge-based intelligent fault diagnosis system consists of a KB with observations and knowledge embedded in experiences. Knowledge such as the root-cause investigation and the fault recovery process can be maintained in the KB for efficient decision-making at the system level. Meanwhile, an inference engine in the knowledge-based intelligent fault diagnosis applies reasoning methods to the known facts in order to help reveal any unknown or indirect relation between the system behavior and a faulty state of the system.

To ensure interoperability of the knowledge-based intelligent fault diagnosis, ontologies conceptualize the domain knowledge with its properties and relations by defining the classes of objects with nouns. For example, the authors of [12] defined four classes in the fault diagnosis ontology model for loaders: *FaultMode* with two subclasses, namely *FaultCause* and *FaultEffect*; *FaultEquipment*, indicating the location of faults; *FaultMaintenance*, describing the fault repair methods; and *Parameters*, expressing the data collected by sensors. In addition, real observations regarding the individual causes and symptoms of, and maintenance actions in response to, a fault can be added to create a knowledge graph, which is a new type of knowledge representation [13]. A knowledge question-and-answer system for fault diagnosis based on knowledge graphs was established in [14]. As complex IIoT systems have evolved to encompass decentralized and spatially distributed but interconnected subsystems, how to share data among various stakeholders has become a concern. Utilizing the decentralization technique of the Web 3.0 paradigm, security, anonymity, scalability, and reliability can be ensured for data exchange among decentralized autonomous organizations. Researchers have investigated distributed knowledge-based intelligent fault diagnosis with independent knowledge graphs constructed and maintained in each subsystem of the IIoT system [8]. But central control is still required in their work. In this work, we focus on decentralized knowledge-based intelligent fault diagnosis over Web 3.0 with distributed knowledge graphs, which is more practical than previous approaches.

B. Distributed Knowledge Inference

The effectiveness of knowledge-based intelligent fault diagnosis using knowledge graphs depends on the completeness and correctness of the knowledge graphs. The inference engine in the knowledge-based intelligent fault diagnosis adopts reasoning methods to infer new conclusions and derive new relations among entities in order to enrich the knowledge graphs. Ontologies and their object properties can be expressed with the resource description framework (RDF) schema. Then, a knowledge graph is constructed with RDF triples. TransE, an algorithm that translates entities and relations to low-dimensional expressions in the embedding space, was originally proposed in [15]. Furthermore, chains of reasoning can be expressed by paths in the graph. Thus, knowledge graphs can be analyzed as graphs with graph topology algorithms. A reasoning method with knowledge inference was proposed in [16]. It uses reinforcement learning (RL) with pre-trained embeddings to predict if a head entity and a given tail entity have a relation. The path searching problem was formalized in [17] as a partially observed Markov decision process using RL to predict the tail entity given the head entity and the relation.

Meanwhile, distributed knowledge inference has been studied to tackle the scalability, performance, and KB isolation issues. In [18], the translation of embedded expressions of knowledge graphs was transformed into distributed ones to resolve the efficiency issue. In [7], a distributed translation of embedding learning was proposed to further improve the approach proposed in [18] by carefully designing the partition of the edges and vertices of the knowledge graph. Nevertheless, distributed knowledge inference based on local knowledge graphs maintained by different participants did not attract much research attention until a distributed path-based reasoning algorithm was proposed in [8]. However, their work involves a central coordinator that coordinates the collaborative training process, which introduces a single point of failure and privacy problems. In this work, we focus on decentralized knowledge inference that is more secure and privacy-preserving.

III. System Model and Methodology

In this section, we introduce the proposed BDKI mechanism for knowledge-based intelligent fault diagnosis with a knowledge graph in IIoT systems.

A. System Overview

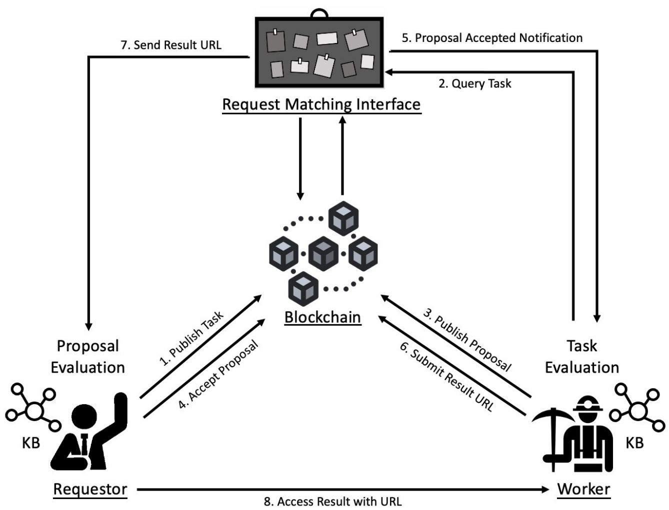


Fig. 2. The Workflow of Proposed Mechanism

The overall system architecture of our Web 3.0 application is shown in Fig. 2. we propose to incorporate a blockchain-empowered *request matching interface* implemented as a smart contract for requestors and workers in the system to publish, browse, and match knowledge requests among the participants. Consider an M-participant IIoT system, each participant maintains a reasoning model that embeds its local knowledge graph and tries to complete the reasoning model with a BDKI mechanism. A BDKI process starts when a participant m requests assistance from other participants to complete its reasoning model that embeds the local knowledge graph. As a task requestor, the participant m publishes a task through the blockchain-empowered *request matching interface*. The other participants can browse published tasks through the *request matching interface* and evaluate a task by evaluating their abilities to complete the requestor's reasoning model with a task evaluation protocol. Then, they will submit their task evaluation results in a proposal through the *request matching interface* as workers. The requestor will choose which proposal to accept. Then, the *request matching interface* will notify the worker whose proposal is accepted by the requestor. Upon receiving the notification, the worker will start performing the knowledge inference task with its local knowledge graph. Finally, the requestor will retrieve the result with the URL submitted through the *request matching interface*, which concludes a round of BDKI training. If the requestor wants to know if the other participants can help complete its reasoning model further, it will pick another worker from the submitted proposals and start a new round of training until it is satisfied.

The proposed mechanism utilizes the blockchain for message exchange, which enables a scalable and open system. Furthermore, it leverages a distributed knowledge inference based on path-based reasoning and RL proposed in [8]. Furthermore, there are two major problems that must be addressed: 1) how should workers evaluate if they can help complete the requestor's reasoning model? 2) how should the requestor choose a worker? In the next section, we will briefly introduce the distributed knowledge inference algorithm and address the aforementioned problems by introducing the task evaluation protocol and proposal evaluation for the requestor in detail.

B. Methodology

As more than one participant (i.e., workers) may have valuable knowledge that could help the requestor to improve the local knowledge graph, after receiving the trained reasoning model from one worker, the requestor may choose another worker and start a new round of training to see if any other participants could help further improve its reasoning model.

For the participant m, we denote the local knowledge graph as $\mathcal{B}_m$, the set of embeddings of entities in $\mathcal{B}_m$ as $\mathcal{E}_m$, and the set of embeddings of relations in $\mathcal{B}_m$ as $\mathcal{R}_m$. Then, knowledge graphs of all participants are denoted by $\mathcal{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_M\}$. More specifically, each local knowledge graph $\mathcal{B}_m$ is composed by a collection of triples $(e_{m,n_1}, r_{m,n}, e_{m,n_2})$ where $e_{m,n_1}, e_{m,n_2} \in \mathcal{E}_m$ respectively denote the embeddings of entities n_1 and n_2 in $\mathcal{B}_m$, and $r_{m,n} \in \mathcal{R}_m$ denotes the embedding of relation n in $\mathcal{B}_m$. The triples in each $\mathcal{B}_m$ are modeled by a directed labeled multigraph $G_m = (V_m, E_m, \mathcal{R}_m)$, where entities in $\mathcal{E}_m$ are modeled by vertices V_m and $r_{m,n}$ is represented as an edge in the graph E_m.

1) Distributed Path-based Reasoning Algorithm: In [8], a distributed knowledge inference framework was proposed with a path-based reasoning algorithm based on RL. After training its reasoning model with a reasoning agent, the proposed framework allows participants to record the entities along the paths with the original query as a handover query HQ_m. By sharing the handover queries with other participants, the distributed reasoning agent can connect links across knowledge graphs $\mathcal{B}$ to address scattered reasoning path problems. It has been proven that participants with small knowledge graphs can benefit significantly from the proposed distributed reasoning framework by initiating training and asking other participants to continue the training using their local knowledge graphs with the handover queries. Thus, in this work, we apply the distributed knowledge inference framework proposed in [8] to our BDKI mechanism. The participant trains its reasoning model with a local knowledge graph and hands over its queries to the other participants in the IIoT system as the requestor, while the others are the workers. Specifically, the reasoning process is a deterministic partially observed Markov decision process. During training, the participant can only observe the

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2023.3344516

head entity e_{m,q_1} and relation $r_{m,q}$ of the query triple, and its current location e_m^t per step t. The answer of the query triple, e_{m,q_2} remains hidden. The observation of the reasoning agent of participant m with $\mathcal{B}_m$ at time step t is then derived as

$$O_m^t = (e_m^t, e_{m,q_1}, r_{q,n}) \tag{1}$$

The state space of the reasoning agent of participant m consists of all valid combinations in $\mathcal{E}_m * \mathcal{E}_m * \mathcal{R}_m * \mathcal{E}_m$. The reasoning agent will choose from the state space at time step t. Thus, the state of the agent at time step t is denoted as

$$S_m^t = (e_m^t, e_{m,q_1}, r_{m,q}, e_{m,q_2}) \tag{2}$$

The set of possible actions of the agent of participant m at time step t that consists of all the outgoing edges of the vertex e_m^t in graph G_m are derived as

$$A_m^t = \{(e_m^t, r, v) \in E_m | r \in R_m, v \in V_m, (e_m^t, r, v) \in G_m\} \tag{3}$$

A path is formed as the reasoning agent transits from one state to the next by selecting an edge and walking to the incident vertex from the current vertex. All the path histories are stored in a long short-term memory (LSTM) based recursive neural network (RNN) network and a two-layer feedforward network that helps the agent to choose from the possible actions A_m^t. The transition function can be formulated as

$$\delta(S_m^t, A_m^t) = (v, e_{m,q_1}, r_{m,q}, e_{m,q_2}), v \in V_m \tag{4}$$

Specifically, the LSTM network stores the sequential histories to encode the path history H_m^{t-1}, the actions an agent has taken A_m^{t-1}, and the observation O_m^t, as H_m^t, where $H_m^t = (H_m^{t-1}, A_m^{t-1}, O_m^t)$. Then, based on the history, the policy network chooses an action a_m^t from a probability distribution over all available actions d_m^t conditioned on the query relation $r_{m,q}$, where

$$d_m^t = softmax(A_m^t(W_{m2}ReLU(W_{m1}[h_m^t; O_m^t; r_{m,q}]))) \tag{5}$$

$$a_m^t \sim Categorical(d_m^t) \tag{6}$$

A reward is given if the answer entity is reached at step T. Given the state at step T is $S_m^T = (e_m^T, e_{m,q_1}, r_{m,q}, e_{m,q_2})$, the reward is calculated as

$$Reward(S_m^T) = \begin{cases} 1, & e_m^T = e_{m,q_2} \\ 0, & \text{else.} \end{cases} \tag{7}$$

Finally, the reward is evenly split among the states on the path within T time steps. The model parameter is trained and updated during back-propagation.

Then, the handover queries, HQ_m, are derived as

$$HQ_m = ([e_m^t]_{t=1}^T, e_{m,q_1}, r_{m,q}, e_{m,q_2}) \tag{8}$$

where $\mathcal{E}_{hq}$ can represent the set of embeddings of entities in HQ_m.

The handover query is sent to the selected worker in the IIoT system. A worker that receives a handover query checks if the handover entities e_{hq} of HQ_m exist in its KB. Using e_i^t to denote the corresponding handover entity found in the $\mathcal{B}_i$ by participant i, with the original query $e_{m,q_1}, r_{m,q}, e_{m,q_2}$, the state space of agent i is,

$$S_i^t = (e_i^t, e_{m,q_1}, r_{m,q}, e_{m,q_2}) \tag{9}$$

Let auc^0 denote the initial overall accuracy that the requestor achieved after training with its local knowledge graph. When a requestor publishes a task, it shares the trained policy network RNN_m, the handover queries HQ_m, and the initial overall accuracy auc^0 with the *request matching interface*. The overall algorithm is summarized in Algorithm 1. Specifically, the time complexity of the Algorithm 1 is $O(n)$, where n is the number of episodes set for the experiment.

Algorithm 1: Participant m trains its own reasoning model as requestor

Given KB_m, construct G
Given training queries $(e_{m1q}, r_{mq}, e_{m2q})$
Function `TrainStep(`A_m^{t-1}, $r_{m,q}$`)`**:**
    $A_m^t = LSTM(A_m^{t-1}, r_{m,q})$
    $d_m^t = softmax(A_m^t(W_{m2}ReLU(W_{m1}[h_m^t; O_m^t; r_{m,q}])))$
    return $a_m^t \sim Categorical(d_m^t)$
Function `Train`**:**
    sample a batch of triples from KB_m
    initialize $A_m^{t-1}$from batch
    initialize $HQ_m = []$
    construct LSTM-based RNN network RNN_m
    for *each episode* **do**
        for $t \leftarrow 0$ *to* T **do**
            a_m^t = `TrainStep(`A_m^{t-1}, $r_{m,q}$`)`
            update RNN_m push this triple to HQ_m
        end
    end
    initiate a knowledge sharing request and upload RNN_m and HQ_m through Request Matching Interface

2) Task Evaluation for Workers: Other participants i perform as workers (i.e., worker i) to evaluate the published task. In this section, we describe the task evaluation protocol for the worker i with $i \in \{1, \ldots, m-1, m+1, \ldots, M\}$.

Each worker i evaluates how confident it is to help the requestor improve its reasoning model with its local knowledge graph with the handover queries of published tasks. Let CI_i denote worker i's confidence indicator with $CI = [CI_i]_{i=1, i \neq m}^M$.

The Iverson bracket of a statement is the indicator function of the set of values for which the statement is true. For entity $e_{i,n} \in \mathcal{E}_i$ in worker i's knowledge graph and $e_{hq} \in \mathcal{E}_{hq}$ in HQ_m, the statement of Iverson bracket $[e_{i,n} = e_{hq}]$ is

$$[e_{i,n} = e_{hq}] = \begin{cases} 1, e_{i,n} = e_{hq} \\ 0, \text{otherwise.} \end{cases} \tag{10}$$

Define the number of equal entities in $\mathcal{E}_i$ and $\mathcal{E}_{hq}$ as

$$C(e_{hq}, i) = \sum_{n=1}^{|\mathcal{E}_i|} [e_{i,n} = e_{hq}] \tag{11}$$

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2023.3344516

where $|\mathcal{E}_i|$ represents the number of entities in $\mathcal{E}_i$.

The path-based reasoning algorithm can be regarded as a path-searching problem based on a path ranking algorithm (PRA). In this context, knowledge graphs are analyzed as graphs utilizing graph topology algorithms, where subjects and objects are treated as vertices, and predicates represent the paths linking these vertices. PRA facilitates a random walk on the graph, collecting paths starting from the head entity h and concluding at the specific entity t within predefined lengths. During collaborative training, it is imperative to locate the handover entity e_{hq} within the worker's local knowledge base. Failure to do so leaves the reasoning agent stuck in a state where no next vertex can be found, resulting in no earned reward. To mitigate this, it is crucial for the requestor to select a worker with more overlapping entities, allowing the worker to identify additional handover entities e_{hq} in its knowledge base. This approach enables the worker to construct a path by connecting more links across knowledge bases. Essentially, the reasoning agent's training goal is to ascertain whether the end entity e_{m,q_2} can be found while traversing paths within the knowledge graph. In the realm of graph search methods, depth-first search and breadth-first search are well-known techniques. Unlike other path-searching methods, such as the Dijkstra shortest path algorithm, A* algorithm, or Yen's algorithm, which prioritize finding the shortest path, the depth-first search and breadth-first search methods are particularly suited for our scenario. Therefore, in this study, we incorporate these methods into the task evaluation framework for workers.

Defined as the depth-first search, the confidence indicator of worker i is

$$CI_i = \frac{C(e_{hq}, i)}{|HQ_m|}. \tag{12}$$

By contrast, a breadth-first search ensures that new knowledge is learned first. For the breadth-first search, the confidence indicator of worker i is

$$CI_i = 1 - \frac{C(e_{hq}, i)}{|HQ_m|}. \tag{13}$$

During training, the requestor compiles handover queries and distributes them to other participants for task evaluation. Subsequently, the requestor selects a participant as the worker for the next round of training based on confidence indicators. Once chosen, the participant initiates training the reasoning model using the distributed reasoning algorithm. Because all workers utilize the identical set of handover queries, there is no requirement for them to reassess the queries after each round. This uniformity in the training process enhances efficiency and consistency across evaluations. The overall algorithm is summarized in Algorithm 2. The time complexity of the function *CalculateConfidentIndicator* is $O(n)$, where n is the number of handover triples. Similarly, the time complexity of the training process is $O(n)$, where n is the number of episodes set for the experiment. In this work, we assume all workers are honorable and that they evaluate truly with their local knowledge graph and submit the result without any manipulation.

Algorithm 2: Participant i evaluates the task and train the model if selected as worker

Function `CalculateConfidentIndicator(`HQ`)`:
- initialize $C(e_{hq}, i)$
- initialize local KB_i
- **for** *each* e_{hq} *in* HQ **do**
  - **if** e_{hq} *in* KB_i **then**
    - $C(e_{hq}, i)$++
  - **end**
- **end**
- **if** *DFS* **then**
  - $CI_i = \frac{C(e_{hq},i)}{|HQ_m|}$
- **end**
- **if** *BFS* **then**
  - $CI_i = 1 - \frac{C(e_{hq},i)}{|HQ_m|}$
- **end**
- share CI_i through the Request Matching Interface

Function `TrainStep(`A_i^{t-1}, $r_{m,q}$`)`:
- $A_i^t = LSTM(A_i^{t-1}, r_{m,q})$
- $d_i^t = softmax(A_i^t(W_{i2}ReLU(W_{i1}[h_i^t; O_i^t; r_{m,q}])))$
- return $a_i^t \sim Categorical(d_i^t)$

Function `Train(`HQ, RNN`)`:
- sample a batch of triples from KB_i
- initialize $A_i^{t-1}$from batch
- initialize RNN_i with RNN
- **for** *each episode* **do**
  - **for** $t \leftarrow$ *0 to T* **do**
    - a_i^t = `TrainStep(`A_i^{t-1}, $r_{m,q}$`)`
    - update RNN_i
  - **end**
- **end**
- upload RNN_i through Request Matching Interface

Function `DecentralizedTraining()`:
- browse Request Matching Interface for new requests;
- retrieve HQ of the request from Request Matching Interface;
- `CalculateConfidentIndicator(`HQ`)`
- if selected as the worker of the task, retrieve RNN from Request Matching Interface
- `Train(`HQ, RNN`)`

3) Proposal Evaluation for Requestor: After a requestor trains its reasoning model with its local knowledge graph, it will initiate a new collaborative training if it is not satisfied with the initial training result. To obtain a reasoning model that has higher overall accuracy with less training effort, a rational requestor will choose the worker with the best evaluation result first. Then, if the requestor still wants to see if other participants could help improve its reasoning model further, it will choose the worker with the second-best evaluation result. That is, the requestor will choose the worker with a greedy approach according to their evaluation results. Let

$$PO = \{PO_1, PO_2, ..., PO_m\} \tag{14}$$

be the set of proposals received from workers. Let $PO_{s(n)}$ be the nth order statistics of the proposals,

$$PO_{s(n)} = max\{PO_{s1}, PO_{s2}, ..., PO_{sn}\}, \tag{15}$$

where $PO_{(s1)}$ is the proposal with the highest confident indicator, $PO^{k}_{(s2)}$ is the proposal with the second highest confident indicator, and $PO_{(sn)}$ is the proposal with the lowest confident indicator. Then, for a new round k, the requestor will choose the worker with the proposal $PO_{s(k)}$.

The proposed BDKI methodology utilizes a task evaluation function for participants to evaluate their possible contributions to the knowledge inference task. Then, the requestor will choose a participant as the worker to initiate a round of training according to their evaluation results. In such a manner, participants are aligned with the ability to contribute to the knowledge inference task and could join the iterative training process in a specific order. Furthermore, the iterative training process stops as the requestor is satisfied with the training result or all participants have trained the requestor's model, which mimics the behavior of the real world. Thus, the proposed methodology is rational.

IV. Evaluations

In this section, we describe our evaluations of the proposed mechanism. First, we use a fault knowledge graph constructed from the TE process [19] dataset to show that the proposed mechanism is feasible for IIoT systems. Then, we use large open-source datasets, namely FB15K-237[1] and WN18RR[2], to show that the proposed method is feasible with large-scale datasets. The results indicate that compared to the conventional distributed knowledge inference methodology with random training sequences proposed in [8], the proposed BDKI mechanism allows participants to join the collaborative training in a specific order that delivers satisfactory training results. Furthermore, we use the open-source dataset WN18RR to show that the proposed mechanism can produce a reasoning model with higher overall accuracy and less training effort; hence, it is beneficial and adoptable for participants in IIoT systems.

A. Datasets

The TE process describes a real industrial process that includes five processing units: a reactor, a condenser, a recycle compressor, a vapor/liquid separator, and a product stripper. Eight chemical components, A–H, undergo a chemical process dominated by the processing unit. Further, 20 fault types and 54 system properties (12 manipulated input variables and 42 measured output variables) of the process are measured using IoT devices; e.g., flow rates, pressure, temperatures, and levels. The TE process dataset is widely used as a benchmark for evaluating process diagnosis methods [20]. In this work, we use the fault knowledge graph constructed from the TE process dataset in [8] to demonstrate the feasibility of the proposed mechanism for IIoT systems.

Furthermore, two well-known open-source knowledge graphs are used in our evaluation: FB15K-237, a subset of Freebase introduced in [15]; WN18RR, a subset of WordNet introduced in [21]. Table I summarizes the number of triples, relations, and entities of each dataset.

TABLE I
DATASETS

Dataset	# Triple	# Entity	# Relation
TE process	1095	112	32
FB15K-237	544230	14541	474
WN18RR	173670	40559	22

B. Experiment Setup

We used Python[3] to implement the functionalities of the requestor, worker, and *request matching interface* in the requestor, worker, and *request matching interface* objects, respectively. Also, we implemented our distributed reasoner for the requestor and worker using Tensorflow[4]. Specifically, we implemented the reasoner using the policy network model illustrated in Fig. 3, which is adapted from [8]. The implemented models and functionalities are deployed on a cloud instance that has 16 core CPU and 32G memory. During each experiment, a requestor, several workers, and a *request matching interface* were run separately and independently to realize the decentralization of Web 3.0 applications. Then, we evaluated our BDKI methodology using the TE process fault knowledge graph and the open-source datasets FB15K-237 and WN18RR. The datasets were split into sub-KBs for each participant. Then, we closely followed the experimental setup in [8] and set the training parameter T, i.e., the number of steps in which the correct answer is reached, to 3. We presented the accuracy of the reasoning model trained by our BDKI with HITS@1, 3, 5, 10, 20, and mean reciprocal rank (MRR), the performance metrics used in knowledge inference tasks, as well as in [8]. Specifically, Hit@N measures the fraction of correct answers that rank in the top N of the returned possible responses to queries. MRR evaluates the multiplicative inverse of the rank of the first correct answer to queries. Furthermore, we assumed all participants are collaborative participants and they will evaluate their ability to contribute to the decentralized training truthfully.

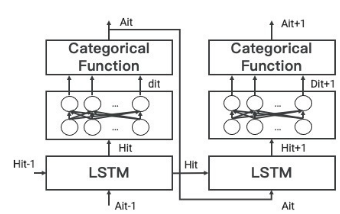


Fig. 3. Network structure. Adapted from [8]

[1]https://developers.google.com/freebase
[2]https://wordnet.princeton.edu/
[3]https://www.python.org/
[4]https://www.tensorflow.org/

C. Numerical Evaluations on TE Process Dataset

The proposed decentralized reasoning methodology allows participants to join the iterative reasoning process in a training sequence that delivers favorable training results. To show that the proposed methodology is feasible for IIoT systems, in this section, we use the TE process fault knowledge graph to demonstrate that the reasoning model trained by the proposed decentralized methodology can achieve higher accuracy that is interpreted with different performance metrics. Specifically, we split the dataset TE Process into four sub-KBs; the numbers of triples and entities involved in each sub-KB are summarized in Table II. We assume that there are four participants (Participant A, B, C, D) in the IIoT system and assign each participant a sub-KB. Then, assuming that all the participants are collaborative participants, we let Participant A initiate the training processes and compare the accuracy of the reasoning model obtained from the training sequence generated by the proposed decentralized reasoning methodology with that of the reasoning models obtained from random training sequences.

TABLE II
DATASET OF EACH PARTICIPANT FOR TE PROCESS DATASET

Dataset	# Triple	# Entity	# Relation
Participant A	271	80	10
Participant B	271	65	2
Participant C	271	79	10
Participant D	271	77	10

Fig. 4 shows the result of Participant A's full iterative training process with Participant B, C, and D trained in different sequences. Specifically, with Participant A as the requestor, Fig. 4 shows the change of different performance metrics as the reasoning model is trained as more rounds of training go on and more workers are involved. As the reasoning model trained after each round of training is a complete model to be used by Participant A in fault diagnosis, we accept the model with the best result as the final trained model of the proposed decentralized reasoning mechanism to show that our mechanism outperforms conventional distributed reasoning methods with random training sequences. Specifically, the best Hit@1 and Hit@3 achieved by the depth-first and breadth-first search sequences are higher than those achieved by training sequences from other combinations of Participant B, C, and D, as shown in Fig. 4(a) and Fig. 4(b). The best Hit@5 achieved by the breadth-first search sequence is higher than that achieved by the other training sequences. However, the best Hit@5 achieved by the depth-first search sequence is slightly lower than that achieved by the training sequence ADCB, while it is higher than that achieved by the other training sequences, as shown in 4(c). For Hit@10 shown in 4(d), the result of the depth-first search sequence is higher than that achieved by the other training sequences. Meanwhile, the result of the breadth-first search sequence is equivalent to the result of the training sequence ADCB and higher than that of the other training sequences. For Hit@20 shown in 4(e), the result of the depth-first search sequence is equivalent to that of the training sequence of ADCB but higher than that of the other training sequences. Finally, for MRR shown in 4(f), the result of the

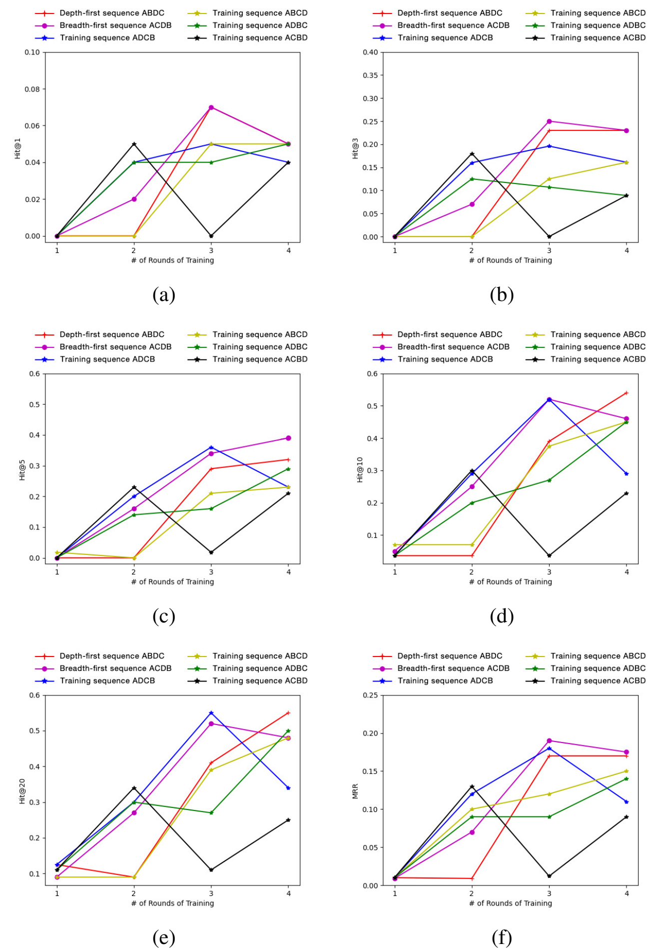


Fig. 4. Comparison of the reasoning models with different performance metrics: (a) Hit@1; (b) Hit@3; (c) Hit@5; (d) Hit@10; (e) Hit@20; (f) MRR

breadth-first search sequence is higher than that of the other training sequences, while the result of the depth-first search sequence is equivalent to that of the training sequence ADCB and higher than that of the other training sequences. For the TE process dataset, the reasoning model obtained by the proposed decentralized reasoning mechanism outperforms the models obtained by the conventional distributed reasoning method with a random training sequence. In addition, the breadth-first search sequence outperforms the depth-first search sequence for the TE process dataset.

D. Numerical Evaluations on Open-Source Datasets

To show that the proposed methodology is feasible for large-scale knowledge graphs, we use the open-source knowledge graphs FB15K-237 and WN18RR to demonstrate that the proposed BDKI methodology delivers a reasoning model with high overall accuracy compared to most reasoning models trained by conventional distributed knowledge inference methods with random training sequences. Here, we split the open-source knowledge graphs into eight sub-KBs; the numbers of triples and entities involved in each sub-KB are summarized in Table III and Table IV. The FB15K-237 dataset contains a total of 14541 entities, as shown in Table I. The smallest sub-KB of the FB15K-237 dataset contains 9079 entities, i.e., 62.4% of the total number of entities in this dataset. The remaining sub-KBs of the FB15K-237 dataset contain entities that account for 80% to 89.6% of the total number of entities in this dataset. Thus, the number of intersections of the entities among the sub-KBs may be large. Meanwhile, the WN18RR dataset has more entities but fewer relations. The WN18RR dataset contains a total of 40559 entities, as shown in Table I. The smallest sub-KB of the WN18RR dataset contains 12029 entities, i.e., 29.7% of the total number of entities in this dataset. The remaining sub-KBs of the WN18RR dataset contain entities that account for 54% to 71% of the total number of entities in this dataset. Hence, the number of intersections of the entities among the sub-KBs may be small.

TABLE III
DATASET OF EACH PARTICIPANT FOR FB15K-237

Dataset	# Triple	# Entity	# Relation
Participant A	27024	9079	470
Participant B	62028	11777	474
Participant C	83429	12639	474
Participant D	101209	13035	474
Participant E	101610	13016	474
Participant F	83031	12684	474
Participant G	60501	11789	474
Participant H	25393	8880	474

FB15K-237 The MRR evaluates the multiplicative inverse of the rank of the first correct answer to the queries. As it is critical for a fault diagnosis system to produce the most relevant fault diagnosis knowledge (e.g., the fault root causes), we evaluate the performance of our BDKI using MRR as the overall accuracy measure in this section. Fig. 5 compares the MRR of reasoning models trained by the depth-first search sequence and the breadth-first search sequence with those of reasoning models trained by 20 random training sequences for the FB15K-237 dataset. The result shows that the breadth-first search sequence outperforms the depth-first search sequence on the FB15K-237 dataset. Furthermore, the breadth-first search sequence outperforms most of the random training sequences. Specifically, for Participant A, the MRR of the reasoning model trained by the breadth-first search sequence is higher than the MRR of 18 models,i.e., 90% of the reasoning models trained by random training sequences. The MRR of the reasoning model trained by the depth-first search sequence is higher than the MRR of 4 models, i.e., 20% of the reasoning models trained by random training sequences. Among the 18 reasoning models, the MRR of the reasoning model trained by the breadth-first search sequence is 8% to 10% higher than that of 4 (22.2%) reasoning models trained by random training sequences, and 4% to 5% higher than that of 10 (55.6%) reasoning models trained by random training sequences. For Participants B and C, the MRR of the reasoning model trained by the breadth-first search sequence is higher than that of 12 and 13 models, respectively, i.e., 60% and 65% of the reasoning models trained by random training sequences. For Participants D and E, 17 models, i.e., 85% of the reasoning models trained by random training sequences, are outperformed by the reasoning model trained by the breadth-first search sequence. The MRR of 6 (35.3%) reasoning models trained by random training sequences are exceeded by the MRR of the reasoning model trained by the breadth-first search sequence by around 5% to 7% for Participant D, where 7 (41.2%) of those reasoning models are outperformed by around 3% to 5% for Agent E. Furthermore, the results of the breadth-first search sequence for Participants F, G, and H follow a similar trend. All 20, i.e., 100% of the reasoning models, trained by random training sequences are outperformed by the model trained by the breadth-first search sequence. The proportion exceeded varies from 2% to 5%. In summary, the breadth-first search of the proposed BDKI mechanism achieves favorable training results compared to the generalized distributed knowledge inference mechanism with random training sequences for the FB15K-237 dataset. It is as expected since the number of overlapping entities among the sub-KBs is large.

TABLE IV
DATASET OF EACH PARTICIPANT FOR WN18RR

Dataset	# Triple	# Entity	# Relation
Participant A	8368	12029	11
Participant B	19344	21875	11
Participant C	26226	26015	11
Participant D	32189	29033	11
Participant E	32520	29127	22
Participant F	26879	26412	11
Participant G	19878	22074	11
Participant H	8260	11979	11

WN18RR The depth-first search should be more effective for the proposed BDKI mechanism for the WN18RR dataset since the number of overlapping entities among the sub-KBs is small. Fig. 6 compares the MRR of the reasoning models trained by the depth-first search sequence and the breadth-first search sequence with those of 20 reasoning models trained by random training sequences for the WN18RR dataset. The result shows that the depth-first search performs better on the WN18RR dataset compared to the FB15K-237 dataset, as expected. Specifically, the reasoning model trained by the depth-first search sequence outperforms 19 models, i.e., 95% of the reasoning models trained by random training sequences, for Participant A. Furthermore, 1 (5.26%) of the reasoning models trained by random training sequences is outperformed by the reasoning model trained by the depth-first search sequence by 35%. In addition, 6 (31.58%) of the reasoning models trained by random training sequences are outperformed by the reasoning model trained by the depth-first search sequence by 10% to 30%. Meanwhile, for Participant A, the MRR of the reasoning model trained by the breadth-first search sequence exceeds that of only 10 models, i.e., 50% of the reasoning models trained by random training sequences. The reasoning models trained by the breadth-first search sequence and the depth-first search sequence perform better on Participant B's sub-KB compared to Participant A's sub-KB. The MRR of the reasoning models trained by the depth-first search sequence and the breadth-first search sequence is higher than the MRR of 17 models, i.e., 85% of the reasoning models trained by random training sequences. Nevertheless, the reasoning model

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2023.3344516

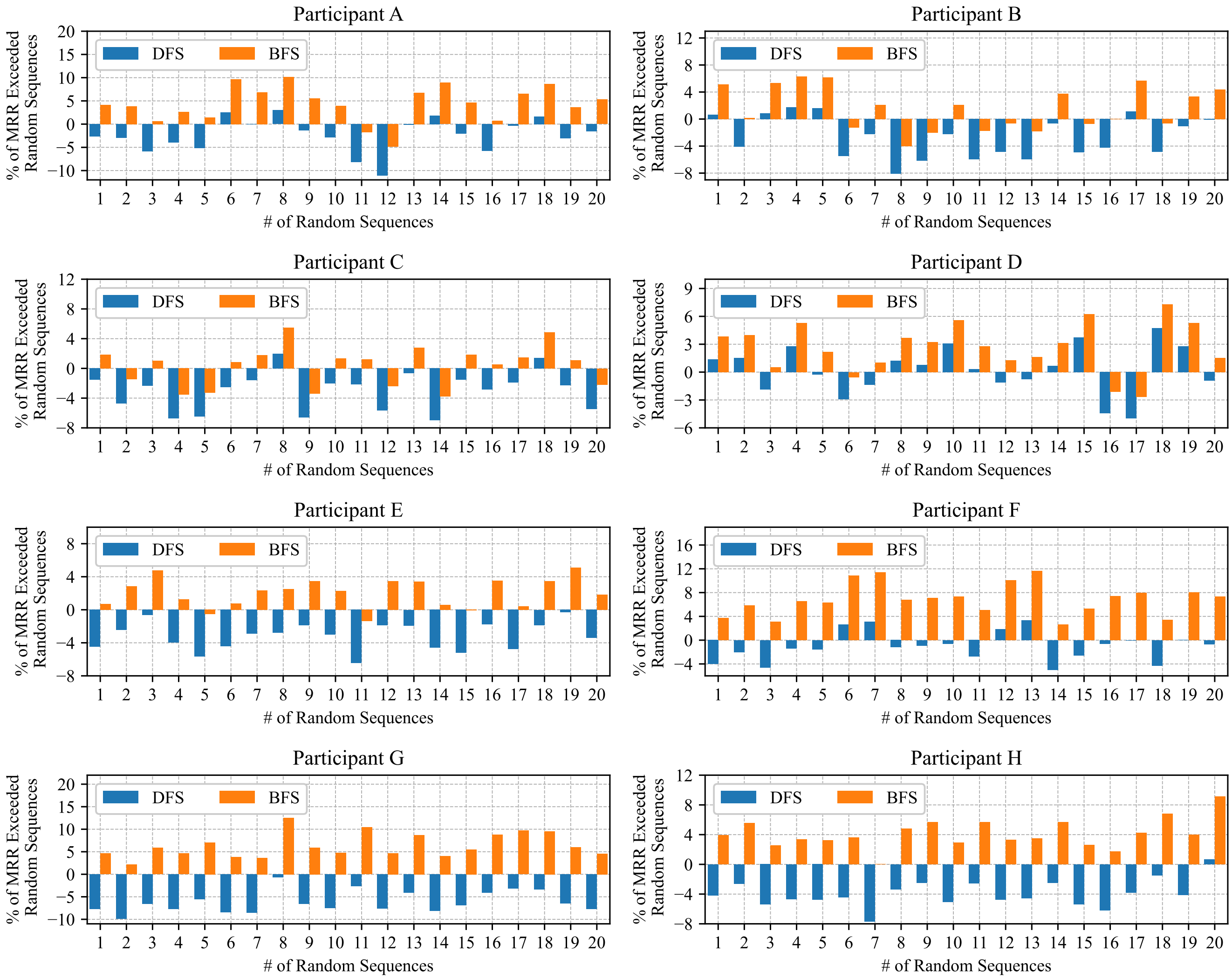


Fig. 5. Comparison of MRR between the proposed mechanism and random training sequences of FB15K-237 dataset

trained by the depth-first search sequence outperforms the reasoning model trained by the breadth-first search sequence in terms of the MRR. The MRR of the reasoning model trained by the depth-first search sequence exceeds that of 13 models, i.e., 65% of the reasoning models trained by random training sequences, for Participants C and D. For Participant E, the MRR of the reasoning model trained by the depth-first search sequence exceeds the MRR of all 20 models, i.e., 100% of the reasoning models trained by random training sequences. In addition, for Participant G, 19 and 18 models, i.e., 95% and 90% of the reasoning models trained by random training sequences, are outperformed by the reasoning model trained by the depth-first search sequence and the breadth-first search sequence, respectively. Finally, for Participant H, 19 models, i.e., 95% of the reasoning models trained by random training sequences, are outperformed by the model trained by the depth-first search sequence. By contrast, only 12 models, i.e., 60% of the reasoning models trained by random training sequences, are outperformed by the model trained by the breadth-first search sequence, by 1% to 10%. In general, the evaluation result confirms that the proposed mechanism delivers reasoning models with favorable overall accuracy. Moreover, the depth-first search sequence outperforms the breadth-first search sequence on knowledge graphs with fewer overlapping entities such as the WN18RR dataset.

E. Performance Evaluation of Proposed BDKI Mechanism

Once the requestor receives a satisfactory reasoning model, it can stop initiating a new round of training with a new worker and end the iterative training process, which reduces the cost of training. Thus, an efficient mechanism should be able to produce a reasoning model that has high overall accuracy and fewer workers and less training involved. In this section, we use the dataset WN18RR to further demonstrate the efficiency of the proposed mechanism. We use the same experimental setup and the same set of sub-KBs described in Table IV. According to our findings, the depth-first search of the proposed BDKI mechanism is more suitable for dataset WN18RR. Therefore, in this section, we use the depth-first search method to show that the requestor can obtain a reasoning model with a

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2023.3344516

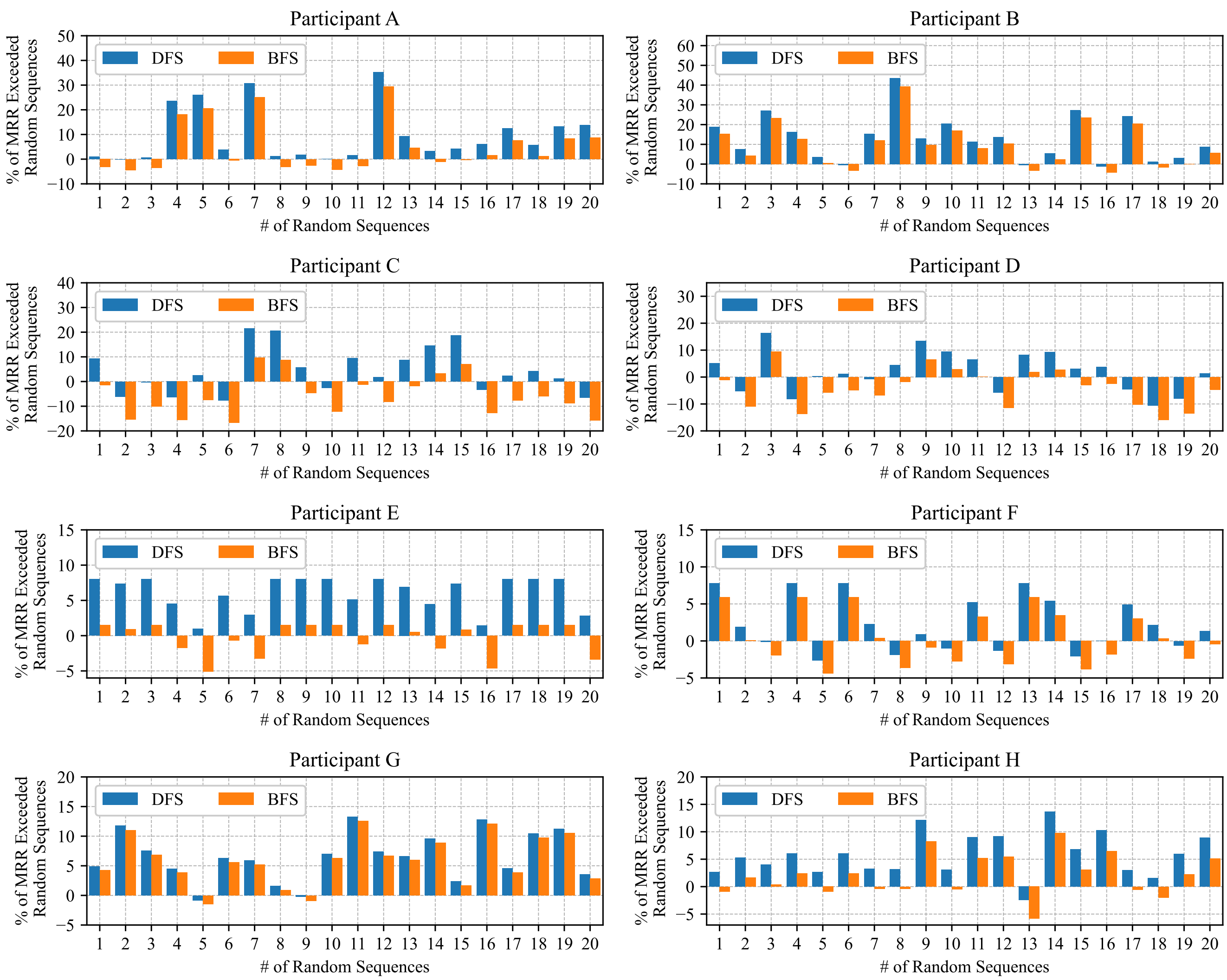


Fig. 6. Comparison of MRR between the proposed mechanism and random training sequences of WN18RR dataset

favorable overall accuracy with less training effort, compared to the general distributed knowledge inference methodology with random training sequences.

For simplicity, we choose Participant A, the participant with the smallest sub-KB, Participant E, the participant with the largest sub-KB, and Participant G, the participant with a mid-sized sub-KB in our evaluation. The result is evaluated using MRR and described as the overall accuracy. The changes in the overall accuracy of the reasoning models as more rounds of training go on and more workers are involved is shown in Fig. 7, where the red dot illustrates the MRR of the reasoning models trained with the proposed BDKI methodology and the blue shallow illustrates the distribution of the MRR of the reasoning models trained with the conventional distributed knowledge inference methodology. Specifically, for Participant A, the best reasoning model with the highest overall accuracy using the proposed BDKI methodology is achieved after one worker has trained. With the same training effort, the reasoning model obtained from the first worker selected using the proposed BDKI is better than 95% of the reasoning models obtained from the first worker of 20 random training sequences. However, the overall accuracy of the model trained by the proposed BDKI methodology decreases after the first worker's training. Thus, a rational requestor should keep the model trained from the first worker. In other words, using the proposed BDKI methodology, the requestor can receive a model with higher overall accuracy than that of 99.2% of the models trained by the conventional distributed knowledge inference methodology with 20 random training sequences, if the iterative training process ends after all the participants have helped on the training and 140 reasoning models have been trained.

For Participant E, the best reasoning model with the highest overall accuracy using the proposed BDKI methodology is achieved after the fifth worker has trained, which is better than all of the 100 models obtained from the first five workers of 20 random training sequences. If the requestor is satisfied with the reasoning model in the early rounds and stops the iterative training process early, it can receive a model with an overall accuracy that is higher than that of 95%, 48.3%, 35%,

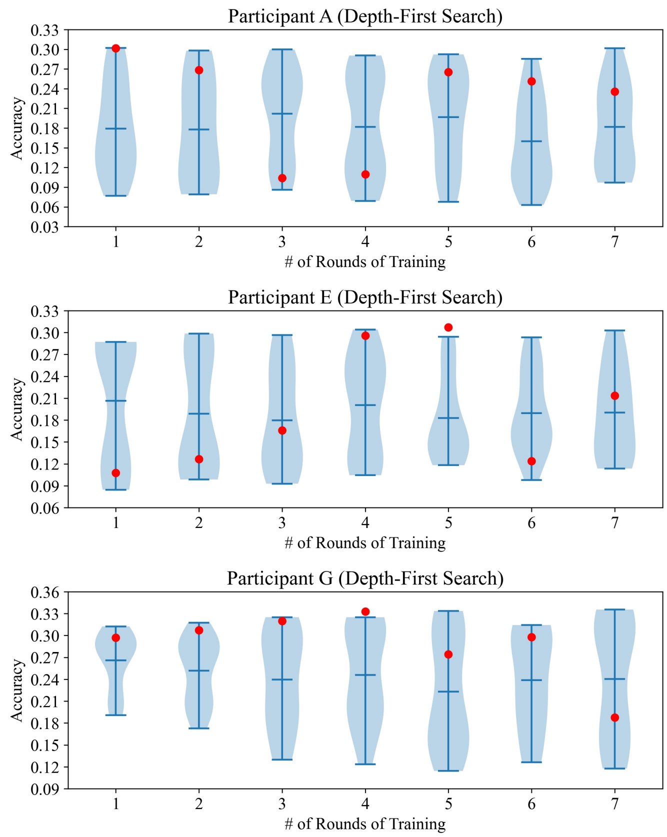


Fig. 7. Performance of the proposed mechanism with dataset WN18RR

and 25% of the models obtained from the fourth, third, the second and first worker of random training sequences, respectively. Similarly, for Participant G, the best reasoning model with the highest overall accuracy using the proposed BDKI methodology is obtained from the fourth worker. Furthermore, the reasoning model obtained from the fourth worker selected using the proposed mechanism is better than all of the 80 reasoning models obtained from the first four workers of 20 random training sequences. If the requestor allows the iterative training process to continue after all the participants have helped with the training, the reasoning model obtained from the fourth worker selected using the proposed mechanism is better than 99%, 99.2%, and 98.6% models obtained from the fifth, sixth, and the last worker of random training sequences, respectively. In addition, if the requestor stops the training process early, it can receive a model with an overall accuracy that is higher than that of 98.3%, 95%, and 75% of the models obtained from the third, second, and first worker of random training sequences, respectively. Fig. 8 summarizes the discussion above. The proposed mechanism can produce a reasoning model that has high overall accuracy and less training effort since the proposed mechanism allows the requestor to choose a worker in a specific order and fewer models need to be trained by random combinations. Thus, it is beneficial and adoptable by the participants for knowledge sharing of intelligent fault diagnosis in IIoT systems. Additionally, from the observation, we find that the proposed mechanism benefits the participants with smaller KB more since a better reasoning model is obtained from early rounds of training with the proposed mechanism.

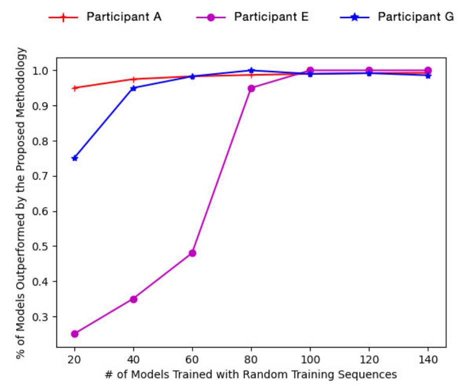


Fig. 8. Efficiency of the proposed mechanism with dataset WN18RR

V. Limitations and Opportunities

The proposed mechanism incorporates a blockchain-powered *request matching interface*, established as a smart contract, to facilitate requestors and workers in publishing, browsing, and matching knowledge requests [22]. Recognizing that the *request matching interface* merely requires decentralization and transparency among participants, we advocate its deployment on a consortium blockchain employing a Practical Byzantine Fault Tolerance (PBFT) consensus model. This strategic choice eradicates the gas fees and latency overhead inherent in public blockchains [23]. In addition, since the requestor chooses workers based on the confidence indicator self-evaluated by workers with no verification methods, the truthfulness of the worker is the key to the success of the proposed mechanism.

VI. Conclusion

We have proposed a BDKI mechanism over Web 3.0 for intelligent knowledge-based intelligent fault diagnosis in IIoT systems. To the best of our knowledge, this is the first attempt to introduce a BDKI mechanism for knowledge-based intelligent fault diagnosis using knowledge graphs into IIoT systems. The proposed mechanism allows collaborative workers to self-evaluate their ability to contribute in completing the requestor's reasoning model with their local knowledge graphs with a task evaluation function. Upon receiving the evaluation results, the requestor will choose a worker with the best evaluation results, thereby offering a more practical decentralized knowledge inference for modern IIoT systems. We have experimentally evaluated the proposed mechanism with the TE process, FB15K-237, and WN18RR datasets. The results show that the proposed mechanism can deliver a reasoning model with higher overall accuracy and less training effort compared to conventional distributed knowledge inference with random training sequences. Moreover, we have experimentally verified that for sparse knowledge graphs, the depth-first search method should be used, whereas the breadth-first search method should be used when there is a large overlap of the entities in each local knowledge graph.

REFERENCES

[1] 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 of Things Journal*, vol. 9, no. 15, pp. 12886–12900, 2022.

[2] X. Wang, J. Yang, J. Han, W. Wang, and F.-Y. Wang, "Metaverses and demetaverses: From digital twins in cps to parallel intelligence in cpss," *IEEE Intelligent Systems*, vol. 37, no. 4, pp. 97–102, 2022.

[3] 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 Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3325–3342, 2022.

[4] M. Dietz, B. Putz, and G. Pernul, "A distributed ledger approach to digital twin secure data sharing," in *Data and Applications Security and Privacy XXXIII*, pp. 281–300, 2019.

[5] L. Su, Z. Wang, Y. Ji, and X. Guo, "A survey based on knowledge graph in fault diagnosis, analysis and prediction: key technologies and challenges," in *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pp. 458–462, 2020.

[6] Y. Lu, Y. Wen, X. Li, and B. Liu, "Exploration of the construction and application of knowledge graph in equipment failure," in *Proc. International Conference on Software, Multimedia and Communication Engineering*, Apr. 2017.

[7] D. Song, F. Zhang, M. Lu, S. Yang, and H. Huang, "Dtranse: Distributed translating embedding for knowledge graph," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 10, pp. 2509–2523, 2021.

[8] Y. Chi, Z. J. Wang, and V. C. M. Leung, "Distributed knowledge inference framework for intelligent fault diagnosis in iiot systems," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021.

[9] H. Chen and S. Lu, "Fault diagnosis digital method for power transistors in power converters of switched reluctance motors," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 749–763, Feb. 2013.

[10] G. Li, H. Chen, Y. Hu, J. Wang, Y. Guo, J. Liu, H. Li, R. Huang, H. Lv, and J. Li, "An improved decision tree-based fault diagnosis method for practical variable refrigerant flow system using virtual sensor-based fault indicators," *Applied Thermal Engineering*, vol. 129, pp. 1292–1303, Oct. 2017.

[11] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE transactions on industrial informatics*, vol. 11, no. 3, pp. 812–820, 2014.

[12] F. Xu, X. Liu, W. Chen, C. Zhou, and B. Cao, "Ontology-based method for fault diagnosis of loaders," *Sensors*, vol. 18, no. 3, 2018.

[13] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 141, p. 112948, Mar. 2020.

[14] F. Yun, Z. Feng, B. Li, and Y. Cao, "Research on intelligent fault diagnosis of power acquisition based on knowledge graph," in *Proc. International Conference on Electronic Information Technology and Computer Engineering*, pp. 1737–1740, Oct. 2019.

[15] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Advances in Neural Information Processing Systems*, vol. 26, pp. 2787–2795, Dec. 2013.

[16] W. Xiong, T. Hoang, and W. Y. Wang, "Deeppath: A reinforcement learning method for knowledge graph reasoning," in *Proc. Conference on Empirical Methods in Natural Language Processing*, Sept. 2017.

[17] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A. Smola, and A. McCallum, "Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning," in *Proc. International Conference on Learning Representations*, Dec. 2018.

[18] D. Zhang, M. Li, Y. Jia, Y. Wang, and X. Cheng, "Efficient parallel translating embedding for knowledge graphs," in *Proc. International Conference on Web Intelligence*, Aug. 2017.

[19] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, Mar. 1993.

[20] H. Chen, P. Tiňo, and X. Yao, "Cognitive fault diagnosis in tennessee eastman process using learning in the model space," *Computers & Chemical Engineering*, vol. 67, pp. 33–42, Aug. 2014.

[21] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proc. of the AAAI Conference on Artificial Intelligence*, pp. 1811–1818, Apr. 2018.

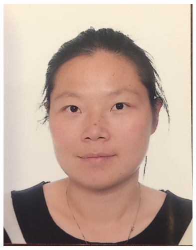
[22] X. Wang, X. Ren, C. Qiu, Z. Xiong, H. Yao, and V. C. M. Leung, "Integrating edge intelligence and blockchain: What, why, and how," *IEEE Communications Surveys Tutorials*, vol. 24, no. 4, pp. 2193–2229, 2022.

[23] S. Fan, H. Zhang, Y. Zeng, and W. Cai, "Hybrid blockchain-based resource trading system for federated learning in edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2252–2264, 2021.

Yuanfang Chi (Student Member, IEEE) received her B.A.Sc. and M.A.Sc. degree from the University of British Columbia (UBC), Vancouver, Canada, in 2012 and 2015, respectively. She is currently a Ph.D. student in the Department of Electrical and Computer Engineering at UBC, and a visiting Ph.D. student in the College of Computer Science and Software Engineering at Shenzhen University, China. Her current research interests include distributed machine learning, knowledge graphs, fault diagnosis, and industrial Internet of Things.

Haihan Duan (Member, IEEE) received his B.Eng. degree in Computer Science and Technology from East China Normal University, Shanghai, China, in 2017, and his M.Eng. degree in Software Engineering from Sichuan University, Chengdu, China, in 2020, and his Ph.D. degree in Computer and Information Engineering from The Chinese University of Hong Kong, Shenzhen, China in 2023. His research interests include multimedia, Web3, metaverse, user-generated content, and human-centered computing. He has served as special session chair for IEEE MMSP'22, and TPC member for top conferences including ACM MM, NOSSDAV, and IEEE ICME.

Wei Cai (Senior Member, IEEE) received a B.Eng. degree in Software Engineering from Xiamen University, China in 2008, an M.S. degree in Electrical Engineering and Computer Science from Seoul National University, Korea, in 2011, and a Ph.D. degree in Electrical and Computer Engineering from The University of British Columbia (UBC), Vancouver, Canada, in 2016. From 2016 to 2018, he was a Postdoctoral Research Fellow at UBC. He is currently an Assistant Professor of Computer Engineering with the School of Science and Engineering at The Chinese University of Hong Kong, Shenzhen. He is serving as the director of the Human-Crypto Society Laboratory, as well as the director of the CUHK(SZ)-White Matrix Joint Metaverse Laboratory. He has co-authored more than 100 journal and conference papers in the areas of distributed/decentralized systems and interactive multimedia. His recent research interests are mainly in the topic of human-centered computing for metaverse, including blockchain, Web3, digital games, social computing, human-computer interaction, and computational art. He is now serving as an associate editor for IEEE Transactions on Computational Social Systems (IEEE TCSS), IEEE Transactions on Cloud Computing (IEEE TCC), ACM Transactions on Multimedia Computing, Communications and Applications (ACM TOMM). He was program co-chair for ACM NOSSDAV'23, open-source software competition chair for ACM MM'23 and reproducibility chair for ACM MMSys'23. He was a recipient of 6 Best Paper Awards. He is a senior member of the IEEE and a member of the ACM.



Z. Jane Wang (Fellow, IEEE) received the B.Sc. degree from Tsinghua University in 1996 and the M.Sc. and Ph.D. degrees from the University of Connecticut in 2000 and 2002, respectively, all in electrical engineering. She has been Research Associate at the University of Maryland, College Park from 2002 to 2004. Since Aug. 2004, she has been with the ECE dept. at the University of British Columbia (UBC), Canada, and she is currently Professor. She is an IEEE Fellow, a Fellow of the Canadian Academy of Engineering (FCAE). Her research interests are in the broad areas of statistical signal processing and machine learning.

Victor C. M. Leung (Life Fellow, IEEE) is a Distinguished Professor of Computer Science and Software Engineering at Shenzhen University, China. He is also an Emeritus Professor of Electrical and Computer Engineering and Director of the Laboratory for Wireless Networks and Mobile Systems at the University of British Columbia (UBC), Canada. His research is in the broad areas of wireless networks and mobile systems, and he has published widely in these areas. Dr. Leung is serving on the editorial boards of the IEEE Transactions on Green Communications and Networking, IEEE Transactions on Cloud Computing, IEEE Transactions on Computational Social Systems, IEEE Access, 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 a Life Fellow of IEEE, and a Fellow of the Royal Society of Canada (Academy of Science), Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the current Clarivate Analytics list of “Highly Cited Researchers”.