Received: 26 March 2021 | Revised: 2 June 2021 | Accepted: 3 July 2021

DOI: 10.1002/ett.4332

SPECIAL ISSUE ARTICLE

WILEY

Intelligent blockchain management for distributed knowledge graphs in IoT 5G environments

Youcef Djenouri[1] | **Gautam Srivastava**[2] | **Asma Belhadi**[3] | **Jerry Chun-Wei Lin**[4]

[1]SINTEF Digital, Oslo, Norway

[2]Brandon University, Brandon, Canada

[3]Kristiania University College, Oslo, Norway

[4]Western Norway University of Applied Sciences, Bergen, Norway

Correspondence
Jerry Chun-Wei Lin, Western Norway University of Applied Sciences, Bergen, Norway.
Email: jerrylin@ieee.org

Abstract

This article introduces a new problem of distributed knowledge graph, in IoT 5G setting. We developed an end-to-end solution for solving such problem by exploring the blockchain management and intelligent method for producing the better matching of the concepts and relations of the set of knowledge graphs. The concepts and the relations of the knowledge graphs are divided into several components, each of which contains similar concepts and relations. Instead of exploring the whole concepts and the relations of the knowledge graphs, only the representative of these components is compared during the matching process. The framework has outperformed state-of-the-art knowledge graph matching algorithms using different scenarios as input in the experiments. In addition, to confirm the usability of our suggested framework, an in-depth experimental analysis has been done; the results are very promising in both runtime and accuracy.

1 | INTRODUCTION

IoT (the Internet of Things) is expeditiously rapidly arising, and the current extent of data and information shift this technology to handle knowledge instead, in particular in 5G environment, where the transactions became more and more larger and bigger.[1-3] IoT 5G networks foster new smart devices and applications as never seen before. Industrial Internet of Things (IoTs) and smart agriculture are few examples of the huge potential number of IoT 5G network applications that are offered to our society. For instance, in the context of health monitoring, Kavitha et al[4] developed an intelligent IoT system which provides different functionalities such as data preprocessing, context-aware, and decision-making process to handle medical data in IoT settings. Smart sensors offered by IoT 5G environment technologies result the creation of large volumes of data varied in time and space. For instance, smart cities are growing rapidly as they aim to deal with over 2.5 billion citizens with a multitude of smart devices by 2050. Making sense of the city interaction is vital to avoid both internal and external conflicts in IT governments.[5,6] Although the current IoT 5G technologies handle with knowledge graphs,[7-9] it has, however, two main issues of these observed solutions as:

1. Missing of common standardization of heterogeneous knowledge graphs of the different sites in the IoT 5G network environments.[10]
2. A known issue of knowledge graphs deals with privacy preservation as well as security issues in IoT 5G network environments.[11]

To deal with the first issue, knowledge graph matching is needed. It is the process of finding the mappings between two knowledge graphs represented in different contexts. It can be applied to several real-life problems. Bellini et al[12] introduced a system for the management of large-volume of concepts and relations from heterogeneous sources. Qui et al[13] developed a semantic graph-based method by incorporating semantic graph structure information and context information that can be used to identify the nontaxonomic relationships in IoT environments. Le et al[14] developed a unified intelligent solution to manage billions of concepts. It also enriches millions of triples for linking to a graph in real-time per hour. All these solutions only deal with two knowledge graphs and did not consider the matching of several knowledge graphs in a real time. Therefore, new approaches to solve the distributed knowledge graphs matching are primordial.

To deal with the second issue, a new security strategy is needed. Blockchain management is an accurate tool for providing high secure platform in IoT 5G environment.[15] Revanesh et al[16] developed a hybrid deep learning framework based on metaheuristic, and blockchain technology for a reliable trustworthy routine scheme in wireless sensor networks. The disseminated routing info in the network is handled by blockchain technology, in which the optimal routing is determined using the Salp swarm intelligence solution. The convolution neural network is also integrated to learn the different variation among the nodes of the network. Dai et al[17] implemented an RL (reinforcement learning) based on blockchain for security in next-gen networks, both wireless and other. Their invented system was shown to maximize the utility of the system, and was also able to accurately cache data sharing across many types of networks. Weng et al[18] invented DeepChain, a framework that possesses a DL (deep learning) based framework that is able to solve known federated learning issues. In DeepChain, the learners may behave maliciously while updating parameters. To solve this issue, they propose a value driven incentive approach using blockchain to motivate nodes to behave correctly. Liu et al[19] handle industrial IoT based issues using blockchain. They adopt an RL approach to give a way to evaluate IIoT systems in terms of latency, privacy, security, scalability, and decentralization. The existing blockchain management solutions do not deal with the different interactions of the knowledge graphs in IoT 5G environment. This research explores the blockchain technology for better secure the distributed knowledge graphs matching process.

Through our own literature search, we have seen that our work here is the first study to explore the distributed knowledge graph matching problem in the IoT 5G environment. In addition, it developed an end-to-end framework which explore both the security issue, and the matching performances. Our contributions are noted clearly as follows:

1. We introduce a new problem, called distributed knowledge graph matching problem. Each IoT node in the set of knowledge graphs is represented by the set of concepts and the set of relations while each relation represents the correlation between two concepts.
2. We further develop a blockchain management strategy for handling the concepts and the relations of the knowledge graphs in a secure way, which is based on Etherum service to store, manage, the concepts and the relations across the different sites in IoT 5G networks. It provides a secure mechanism for concepts and relations sharing.
3. We present a new approach for knowledge graphs matching. It adopts decomposition method to split the concepts and the relations of the knowledge graphs into similar groups. Instead of exploring the whole concepts and relations of the knowledge graphs, only the representatives of the groups are exploited for better and accurate matching. In addition, matching similar concepts and relations instead of homogeneous knowledge graphs gives better accuracy in the final matching results.
4. Extensive experiments were carried out to validate the applicability of the proposed framework. Two well-known knowledge graph matching are used. The first data are called Kensho Derived Wikimedia, with more than 75 million items. The second one is CORD-19 knowledge graph with more than 66 000 concepts and 133 relations. The results reveal that both the suggested framework outperformed the state-of-the-art knowledge graphs matching solutions in terms of runtime, and the quality of returned solutions.

We organize things as follows. Works on the knowledge graph matching problem, and blockchain management are discussed in Section 2. Section 3 presents the proposed framework for distributed knowledge graph matching problem. A performance evaluation of the proposed framework is given in Section 4. Finally, Section 5 draws the conclusions and future perspectives of the knowledge graph matching problem.

2 | RELATED WORK

This research work is based on two main topics: knowledge graph matching and blockchain management.

2.1 | Knowledge graph matching

Matching strategies based on concepts are appropriate for connecting database records. Much research has explored methods for improving the efficiency of knowledge graph matching. Solutions regarding the knowledge graph matching issue mainly is categorized into two groups: i) solutions based on the reduction of the search space by employing computational intelligence, data mining, and machine learning methods and ii) solutions based on high-performance computing while parallel matching is established. This work focuses on the solutions based on the reduction of a search space and approaches in this category are discussed in the following section. Li et al[20] developed a concept regarding matching approach, named VMI. For each concept, it builds two distinct vectors, such as the vector name as well as a virtual document vector. The VMI method is able to reduce similarity measurements by using multiple indexing and candidate selection and operates effectively only in large cases with a limited number of data properties. The best results are obtained when users specify all the corresponding data properties and methods of retrieving values. Therefore, the approach used is based solely on the generic matching concept algorithm. However, some processes are applied to particular areas, that is to say, using simple string comparative methods for names/data characteristics utilizes comprehensive instance information. In the 2009 OAEI competition[*] for small knowledge graph, VMI obtained successful matching. However, with increasing instances, its quality decreases. Wang et al[21] developed an approach that is based on the hypothesis that, two entities of the same real-world object may be matched when they are related to previously matched entities. This technique incorporates multiple lexical matches using a new voting aggregation process and only uses the structural information and the correspondences observed to locate the additional information, which can then primarily be broken down into two stages: the identification of highly accurate seminal correspondences by lexical information and the derivation of additional matching outcomes based on the semantic matching of the previous stage with a structural matching strategy.

Based on the findings of the 2010 OAEI study, this method obtains a reasonable accuracy for certain medium and small knowledge databases. Shao et al[22] presented RiMOM at the OAEI competitions in 2013 and 2016. It introduces an iterative matching framework in which the distinctive information is centered on a blocking technique for minimizing the number of pairs of candidates. As a key to the index of the concepts, it uses predicates, and its distinctive object. Moreover, a weighted, exponential similarity averaging method is used to ensure that the concept matching fits with the high precision. The new blocking approach decreases the computational cost significantly without losing precision and recall. RiMOM achieves 99% accuracy in small and medium knowledge graphs. Alam et al[23] developed an expansion of MERGILO, a method to reconcile knowledge graphs extracted from the text by graph matching and word similarity. Compared with the generic approaches, the results of the extended MERGILO show significant improvement. Rosaci[24] found that graph matching can be used to link various smart agents. The knowledge graph of an agent simulates the actions of an agent, and, if an agent proposes, then any agent in the group will know the relation between itself and another agent. Rosaci[25] then used the hierarchical model to identify semantic associations between web data. The semantic connections represented by metadata are discussed in the context of a collection of network entities. The usefulness of this approach has been demonstrated in well-known web user recommendation systems. The interlinking issue was first addressed as problems of duplication or record linkage by the database community, where Elmagarmid et al[26] based their research on several methods to tackle the problems of heterogeneity in graph matching and proposed a method of handling a set in organized property-segmented documents. To address the matching problem in LOD by using rules taken from the association rule mining technique, Niu et al[27] developed the extended inverse functional property suite (EIFPS) technique, which is considered to be a semi-supervised learning approach. A limited number of current matches *owl:sameAs* are used as seeds and the related rules as criteria for optimizing precision are considered. It presented a graphic metric that measures the likelihood and law of Dempster while integrating confidence values. Then, by presenting the power of resource homogeneity for the e-learning context, Ceron et al[28] presented the LOM framework. To expand and improve the available tools for online learning semantically, the use of the initial associative classifier for ontology matching was then developed and investigated. This model uses a feature-based similarity function that needs historical knowledge as the training set. This method was evaluated and verified at the 2014 OAEI knowledge graph database competition. The results for several larger knowledge graph databases showed 90% precision. Ochieng et al[29] presented an approach that splits a graph into many partitions. Cluster-based similarity aggregation (CSA)[30] is a system integrating varied factors (i.e., five measures, a string-similarity calculation, and a WordNet-based similarity measure) to derive the alignment of ontology concepts. Algergawy et al[31] then proposed a large-scale ontology matching clustering approach. The main concept is to divide the schema graph by using context-driven structural node similarities into clusters. The Vector Space Model

[*]http://oaei.ontologymatching.org/.

(VSM) is also defined after the partitioning of each ontology to discover similar clusters and generate the same concepts. Belhadi et al[32] proposed the genetic feature selection for ontology matching (GFSOM), a hybrid solution for improving the ontology matching process. The relevant properties of the ontology are first selected using the feature selection process. The genetic algorithm is then performed in order to explore the alignment space between two ontologies. Belhadi et al[33] proposed a pattern mining for ontology matching (PMOM), which is a data mining based solution for the ontology matching. The set of frequent patterns of both ontologies are first discovered, instead of exploring the whole set of properties, only these relevant patterns are checked to find the best alignment.

2.2 | Blockchain management

Dai et al[17] created an RL (reinforcement learning) architecture using blockchain to secure next-gen networks, both wired and wireless. Their novel system was shown to maximize utility as well as cache data sharing accurately across the entire network. Weng et al[18] invented DeepChain, a novel framework which can be defined as a distributed DL (deep learning) system that can be used for solving FL (federated learning) issues. In their novel system, learners are known to behave in an incorrect manner when parameter updates are taking place. Their system is based on having incentives that are value-driven incentive in a blockchain based system that mandated participants to hopefully behave in correct manners. Liu et al[19] handle blockchain enabled IIoT (the Industrial Internet of Things) problems and make use of an RL (reinforcement learning) based approach that can give a clear mechanism for the evaluation of IIoT (the Industrial Internet of Things) systems maintaining security, privacy, trust, scalability, latency, and decentralization. Qiu et al[34] dealt with optimization problems, and a Q-learning approach to be able to solve and describe access selection, view change, as well as resource allocation in blockchain systems. Liu et al[35] implemented a reinforcement learning blockchain-enabled approach that could create a safe and secure environment and can maximize collection of data in IIoT systems. Dai et al[36] handled the offloading problem online using a Markov decision tree. Their system integrates RL, blockchain mining, and the well know Genetic Algorithm to be able to maximize offloading performance long term. Chai et al[37] implemented a FL (federated learning) hierarchical system that can be used for knowledge sharing in vehicles. In a similar area, Lu et al[38] implied that an asynchronous, blockchain-based FL solution could handle security issues in the Internet of Vehicles (IoV). Youyang et al[39] created a novel FL based strategy that uses blockchain. Their system enables learning at a local level for terminal devices through the exchange with the global learning based model using blockchain technology. Furthermore, their system was also able to allow autonomous ML (machine learning) for sustaining a global model without the need for a centralized authority. Luo et al[40] discussed a blockchain-based IoT technology that can synchronize the local views in between many different SDN (software-defined network) controllers and is able to achieve an accurate consensus in the global view. Their novel approach was able to reduce computational resources while also able to consider hidden features for controllers as well as resource constraints in the environment simultaneously. Abbas et al[41] looked at authentication of distributed medical patients in hospital-based networks using blockchain. We have seen ample work in the emerging telecommunications field recently that uses blockchain technology and artificial intelligence based solutions in 5G IoT environments in many different fields.[42-44]

2.3 | Discussion

As seen in the above short review of literature, current knowledge graph matching solutions have good results on small-scale databases (ie, many small and medium concepts) in terms of runtime and the solutions of the quality. It also focuses on discovering matching inside the knowledge graph, and does not deal with distributed knowledge graph matching problem, which is vital in IoT 5G environments. It also neglects security problems, as for the existing blockchain learning technologies; they are not dedicated to semantic knowledge graph matching. In this work, we present an end-to-end solution based on intelligent blockchain management for matching knowledge graphs in distributed IoT 5G environments.

3 | IBM-DKG: INTELLIGENT BLOCKCHAIN MANAGEMENT FOR DISTRIBUTED KNOWLEDGE GRAPH MATCHING

The proposed IBM-DKG shown in Figure 1 (intelligent blockchain management for distributed knowledge graph matching) employs both blockchain management and decomposition for distributed knowledge graph matching. The process

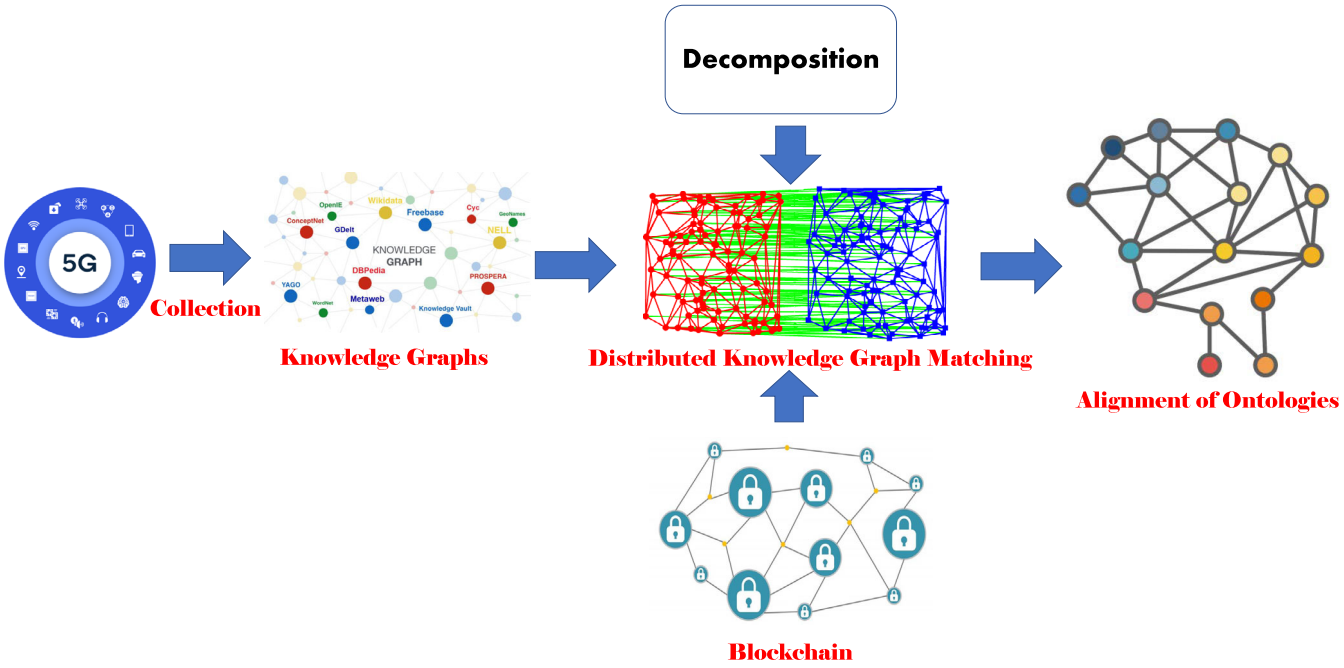

FIGURE 1 IBM-DKG framework

starts by transforming the knowledge graphs into a set of groups of similar concepts and relations. Instead of exploring the whole knowledge graphs, only the representative groups are checked for retrieving the best matching of each pair of the knowledge graphs. The output of the IBM-DGK will be the set of alignments of each pair of the knowledge graphs in the system. The detailed explanation of the proposed framework is presented in the following section.

3.1 | Distributed knowledge graph matching

The aim of this part is to build a distributed knowledge graph matching system. We consider a set of knowledge graphs, where each knowledge graph is represented by a set of concepts C, and set of relations R. Each relation relies semantically two adjacent concepts in the graph. Distributed knowledge graph matching problem aims to determine the shared concepts from of the set of knowledge graphs in distributed settings. In this section, we will present a new approach for accurately solve the distributed knowledge graph matching problem. The knowledge graphs are divided into several dependent groups, each of which contains highly correlated concepts. The concepts of each group are intelligently explored in order to deduce the shared concepts. The decomposition step is considered as pre-processing, where the matching could be applied several times in the same set of derived groups. Each group of concepts shared maximum number of relations. Instead of running the matching on the whole concepts and relations of each knowledge graph, the concepts of each group are explored, where the similarity among the representative of the groups is determined. The set of knowledge graphs $\mathcal{K}$, the set of concepts C is considered as input, and the best matching as $\mathcal{M}^*$. The set of groups $\mathcal{G}$, and the representative of each group G_i is noted g_i. The first step is to generate the representative of the groups in $\mathcal{G}$. The first loop step is to scan all the concepts for each knowledge graph in $\mathcal{K}$. We determine the distance between each group representative and each concept in K_i. The lowest distance value between the concept c and all the groups representative in g is returned. The concept c is assigned to the cluster with the lowest distance value. All groups representative are updated and kept in the set g'. This process is repeated until a stabilization among groups is observed. The final groups of each knowledge graph are stored in matrix, which is called $\mathcal{MK}$. Each element $\mathcal{MK}[i][j]$ is the distance between the g_j and the ith concept of the jth group.

After the decomposition step, the groups of the knowledge graphs are explored in order to find the best matching $\mathcal{M}^*$ of each pair of knowledge graphs. Instead of comparing all concepts, only the representatives of the groups are checked and compared. If the similarity between the g_i^1 and g_i^2, the two representatives groups of the knowledge graphs K_1 and K_2, respectively, is greater than a given threshold, then the concepts of the groups G_i of both knowledge graphs

are concatenated to the matching results of the knowledge graphs K_1 and K_2. This process is repeated for all groups of the knowledge graphs K_1, and K_2. The complexity of the proposed solution for distributed knowledge graph matching depends on the number of concepts $|C|$, the number of relations $|R|$, the number of groups $|G|$, the number of knowledge graphs $|K|$, and the number of possible matching p. In the designed model, the decomposition step thus needs $O(|C| \times |R| \times |K|)$. This process is performed only once for the set K whatever the number of matching to be established. Only similar groups are exploited during the matching process. This requires $O\left(\frac{|C|\times|R|\times|K|}{|G|}\right)$. The total cost of the proposed solution for establishing p matching is $O\left(|C| \times |M| \times |K| + p \times \frac{|C|\times|R|}{|G|}\right)$, which is significantly lower than the baseline solutions that require $O(|C| \times |R| \times |K| \times p)$.

For instance, consider two knowledge graphs $K_1 = \{R'_1, R'_2, \ldots, R'_{30}\}$ and $K_2 = \{R_1, R_2, \ldots, R_{30}\}$ of the same concepts $C = \{C_1, C_2, \ldots, C_{60}\}$. Each relation describes two different concepts in C. The first step aims at extracting the set of relations $R$and R' and grouping them into several subsets. The matching process is then performed to derive an alignment among the two knowledge graphs. The reference alignment represents the set of the common relations among two knowledge graphs. Thus, the optimal matching between K_1 and K_2 is, for example, $R_1 = R'_{12}$, $R_3 = R'_{15}$, and $R_{10} = R'_{26}$.

3.2 | Security

The main objective of this part is to secure the proposed framework using blockchain technology. Ethereum is used as a service to store, and manage the data across the different sites in a safe way. It provides a secure mechanism for data sharing by creating a highly guarded blockchain system. The blocks are first created, where each block contains the representative groups of the knowledge graph derived by each site. Once the block of the given site is calculated, a hash is determined, in order to avoid updating of the representative groups of the knowledge graph of the site. The hash is used here to protect the representative groups of the knowledge graph of each site, and easily identify unexpected changes by the hackers. A proof of work strategy is also integrated for avoiding automatically detecting the hashes. All sites read the proof of work, and agree about the entities that are able to create new blocks. The smart contract are also delivered to the sites. The smart contracts are cached in the different blocks of the sites, which can be used to automatically exchange hidden information among sites. All sites accept request authentication for data exchanging. Each site enrolls agreement with the certificate authority and saved its public and private keys in hidden space. The encryption system is also needed to be used for ensuring the privacy of data transportation across the different sites of the proposed system. The certificate authority permanently verified both the data source and destination. If the data are sent or receive from nonlegitimate site, a transaction is rejected, and a report is made, where the IP address of the detected site is stored in a block list. Once the certificate authority checks the validity of the transaction, both signature and the encrypted data are given back to the designated site which will be delivered to the blockchain system as saved inquiry from that site.

Algorithm 1 presents the formal description of the IBM-DKG framework. The set of relations R^i of each knowledge graph K_i is considered as input, and the best alignment $\mathcal{A}$ as output. The set of clusters is represented by $\mathcal{G}$, and the set of centroids is stated as g. The first step is to randomly initialize the centroids using the function *InitializeCenters()*. The first loop aims to scan all the set of relations. The function *Distance*(e, g_1) calculates the distance between the relation and the first centroid g_1. Consider e = {(Name, Joe), (age, 26), and (type, man)} and the centroid is set as g_1 =(26, man, USA), *Distance*(e, g_1) aims to calculate the intersection of values, which is set to 2. The next loop finds the smallest distance between the relation e and all the centroids in g, where it conserves the range r. Each relation e is assigned to the cluster r, which represents the minimum distance using the function *AddElement()*. Afterward, the centers are updated and kept in the set g'. If g_{new} is equal to the previous center in g, then the decomposition process is then terminated; otherwise, the same process is repeated until g_{new} and g become the same. It scans the set of centroids G^i, G^j of the two knowledge graphs K_i and K_j, and the minimum distance between two centroids with the function *Distance*($g^i_{l_1}$, $g^j_{l_2}$). The minimum distance is selected and the two clusters are added to the list of the alignment clusters *list* using the function *AddClusters*(). It scans again the whole relations of the two aligned clusters. Here, p and q are represented as the two selected clusters, and scans all the relations e_1 and e_2 for both clusters p and q, and the minimum distance is computed. For the set of aligned $\mathcal{A}$, the alignment results of the clusters p and q are then added and denoted as $\mathcal{A}_{p,q}$. This process is repeated for all the clusters in *list*. The privacy of the whole process is ensured by the standard and generic blockchain technology, which is not stated here since it is not the major problem to be solved in this article.

Algorithm 1. IBM-DKG: Intelligent blockchain management for distributed knowledge graph matching

Input : $C^i = \{C_1^i, C_2^i ... C_{c_i}^i\}$: the set of c_i concepts of the knowledge graph K_i.$R^i = \{R_1^i, R_2^i ... R_{n_i}^i\}$: the set of n_i relations of the knowledge graph K_i.
Output: $\mathcal{A}$: Alignment set.

```
InitializeCenters(g^i)
for each relation e ∈ R^i do
    dis ← Distance(e,g_1^i)
    r ← 1
    for j=2 to k do
        d ← Distance(e,g_j^i)
        if d < dis then
            dis ← d
            r ← j
        end if
    end for
    AddElement(e,c_r^i)
end for
repeat
    change ← false
    g_new^i ← UpdateCenter (g,G^i)
    if g^i != g_new^i then
        change ← true
    end if
until change == false
list ← ∅
for p = 1 to k_i do
    min ← Distance ( g_p^i, g_1^j)
    index ← 1
    for q = 2 to k_j do
        d ← Distance(g_p^i, g_q^j)
        if d < min then
            min ← d
            index ← j
        end if
    end for
    list ← list ∪ AddClusters( c_p^i, c_r^j)
end for
A ← ∅
for each (p, q) ∈ list do
    min ← n_i × n_j
    for each instance (e_1, e_2) ∈ (G_p^i, G_q^j) do
        d ← Distance(e_1, e_2)
        if d ≤ min then
            min ← d
            A_{p,q} ← (e_1, e_2)
        end if
    end for
    A ← A ∪ A_{p,q}
end forreturn A
```

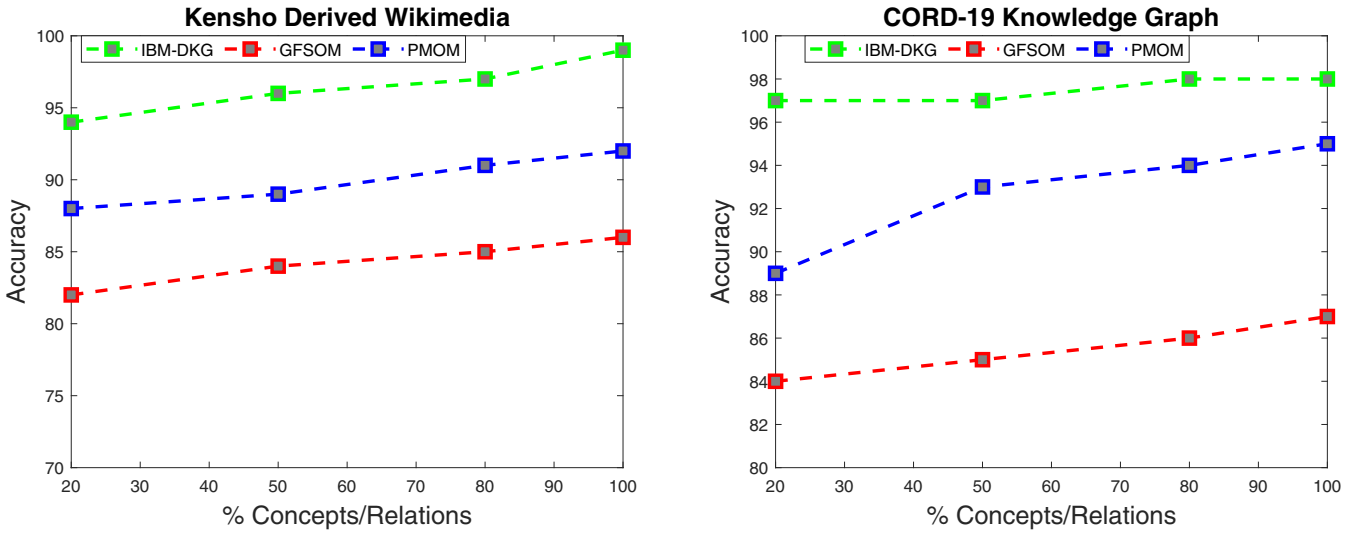

FIGURE 2 Accuracy comparison of the IBM-DKG and the state-of-the-art knowledge graph matching solutions by varying the percentage of concepts and relations

4 | PERFORMANCE EVALUATION

Extensive experiments were conducted on well-known knowledge graphs to validate the usefulness of proposed IBM-DKG framework. The experiments were carried out on a desktop with an Intel *i7* processor and 16 GB of main memory. Python language was used for all the implemented algorithms. We used two different datasets:

1. Kensho Derived Wikimedia Dataset[†]: It represents the Wikipedia, the free knowledge base. It is almost 20 years old and recently added its six millionth article in English. The dataset was created in 2012 but has been growing rapidly and currently contains more than 75 million items.
2. CORD-19 Knowledge Graph[‡]: It is comprised of 50 752 gene nodes, 10 781 disease nodes, 5738 chemical nodes, and 535 organism nodes. These nodes are connected by 133 relation types including Gene–Chemical–Interaction Relationships, Chemical–Disease Associations, Gene–Disease Associations, Chemical–GO Enrichment Associations, and Chemical–Pathway Enrichment Associations.

We split these two datasets into different knowledge graphs in order to simulate the mechanism of distributed knowledge graphs matching proposed in this article. Two baseline algorithms have been compared with, which represents the state-of-the-art knowledge graph matching algorithms. The first algorithm called GFSOM[32] which received the best paper award in the international conference on genetic and evolutionary computing. The second algorithm called PMOM[33] which is recently published in European conference on advances in databases and information systems. The runtime is calculated by seconds, and the accuracy is determined by computing the percentage of the corrected matched.

4.1 | Accuracy

Figures 2 and 3 present the accuracy of the IBM-DKG on *Kensho Derived Wikimedia database* and *CORD-19 Knowledge Graph* compared with GFSOM and PMOM. In Figure 2, we fix the number of knowledge graphs to 20, and we varied the percentage of concepts, and relations used on each knowledge graph from 20% to 100%. However, in Figure 3, we fix the percentage of concepts and relations used on each knowledge graph to 100%, and we varied the number of knowledge graphs from 2 to 20. The results reveal that IBM-DKG outperforms the two baseline algorithms in terms of accuracy, determined by the number of corrected matching. Indeed, the accuracy of IBM-DKG exceeds 97% of corrected matching;

[†]https://www.kaggle.com/kenshoresearch/kensho-derived-wikimedia-data.
[‡]https://www.kaggle.com/yitongtseo/cord19-named-entities.

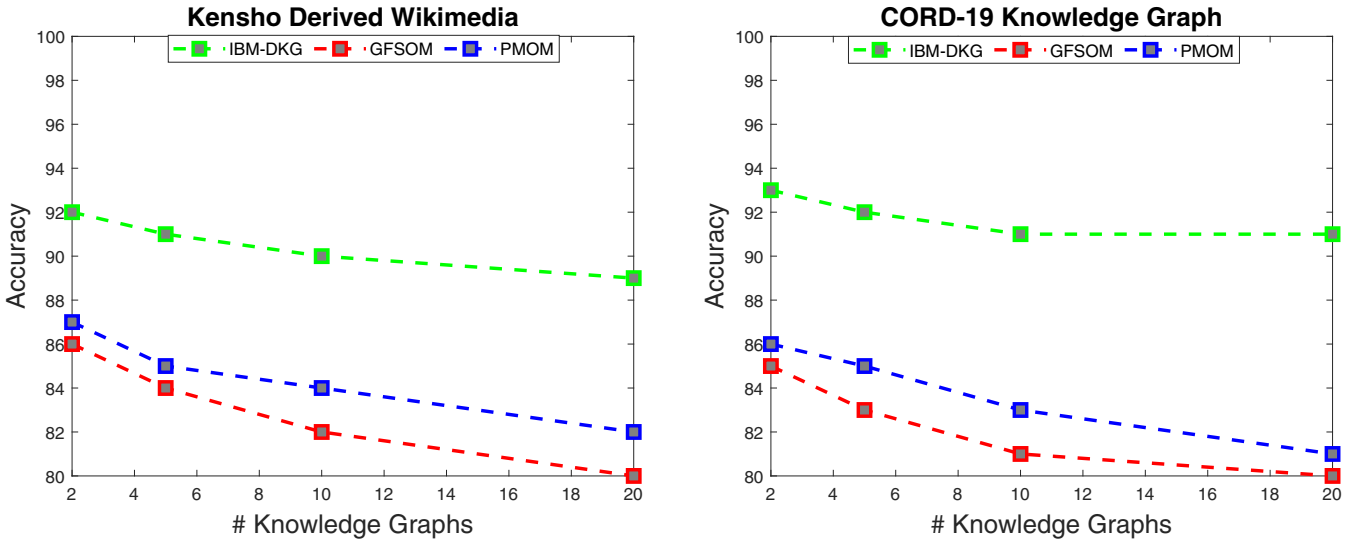

FIGURE 3 Accuracy comparison of the IBM-DKG and the state-of-the-art knowledge graph matching solutions by varying the number of knowledge graphs

however, PMOM accuracy goes below 94%, and the GFSOM accuracy goes below 86%. This is explained by the fact that GFSOM and PMOM are approximate-based solutions. GFSOM is based on stochastic based process, which can refer to genetic algorithm, and PMOM is based on the relevant patterns discovered in each ontology, which are highly dependent to the minimum support value. However, IBM-DKG uses efficient strategies to match the knowledge graphs. It benefits from the decomposition step by dividing the concepts and relations of the knowledge graphs into small components, each of which contain few but highly correlated concepts and relations. It also provides an efficient blockchain management for ensuring the safely sharing of the different representatives of the groups of the knowledge graphs.

4.2 | Computational time

Figures 4 and 5 present the runtime of the IBM-DKG on *Kensho Derived Wikimedia database*, and *CORD-19 Knowledge Graph*, compared with GFSOM and PMOM. By varying with the number of concepts, relations, and the knowledge graphs used as input, IBM-DKG outperforms the two baseline algorithms in terms of runtime. Indeed, the computational time of IBM-DKG does not exceed 45 ms; however, PMOM runtime reaches 60 ms, and the GFSOM runtime exceeds 48 ms. This is explained by the fact that IBM-DKG only explores the representative of the groups of the knowledge graphs, whereas the GFSOM uses high number of concepts and relations in the matching process, and PMOM studied the different correlations among the concepts and the relations of each knowledge graph which is high time consuming.

4.3 | Discussions

From our extensive experiments dealing with distributed knowledge graph matching problem, some perspectives remain to be studied:

1. *Preprocessing of knowledge graphs*: In order to increase the distributed knowledge graph matching performances, the data should accurately preprocessed. The preprocessing step should include different directions, for instance, removing outliers and noises from the knowledge graphs, missing of outlier detection from the knowledge graphs in the advanced solutions for outlier detection.[45-48] Adaptation of such methods allows the process of distributed the knowledge graph matching more robust and accurate. One way to adapt such methods for dealing knowledge graphs, is to develop operators dedicated to the knowledge graphs such as the local reachability distance among concepts and relations, and the set nearest neighbors of the knowledge graphs. Another interesting preprocessing step is feature

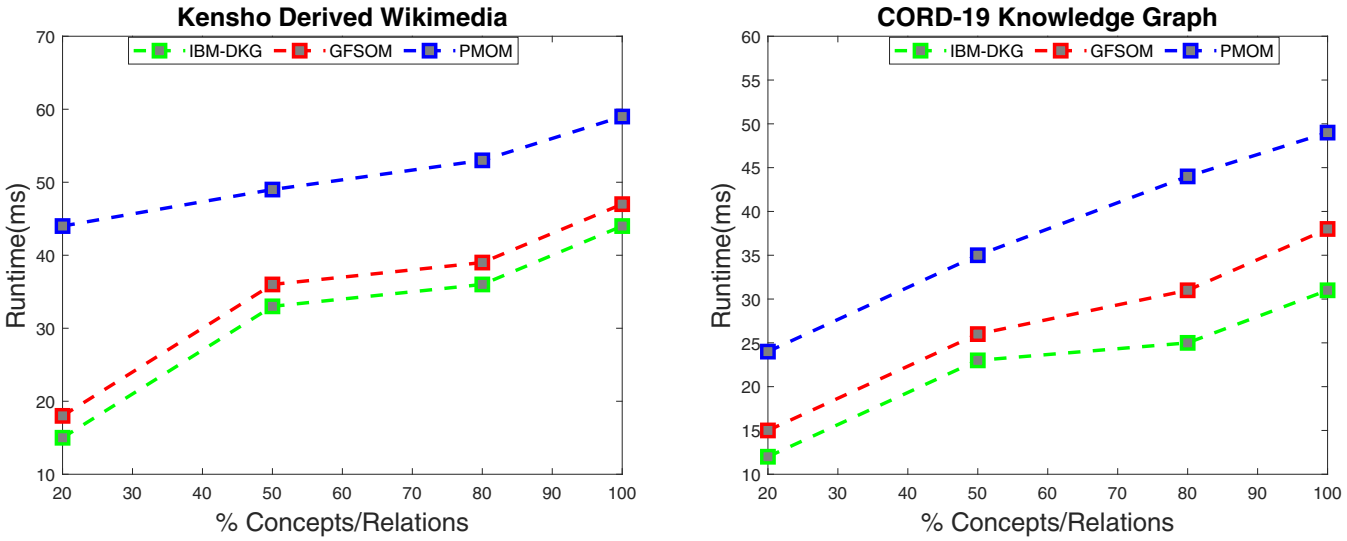

FIGURE 4 Runtime comparison of the IBM-DKG and the state-of-the-art knowledge graph matching solutions by varying the percentage of concepts and relations

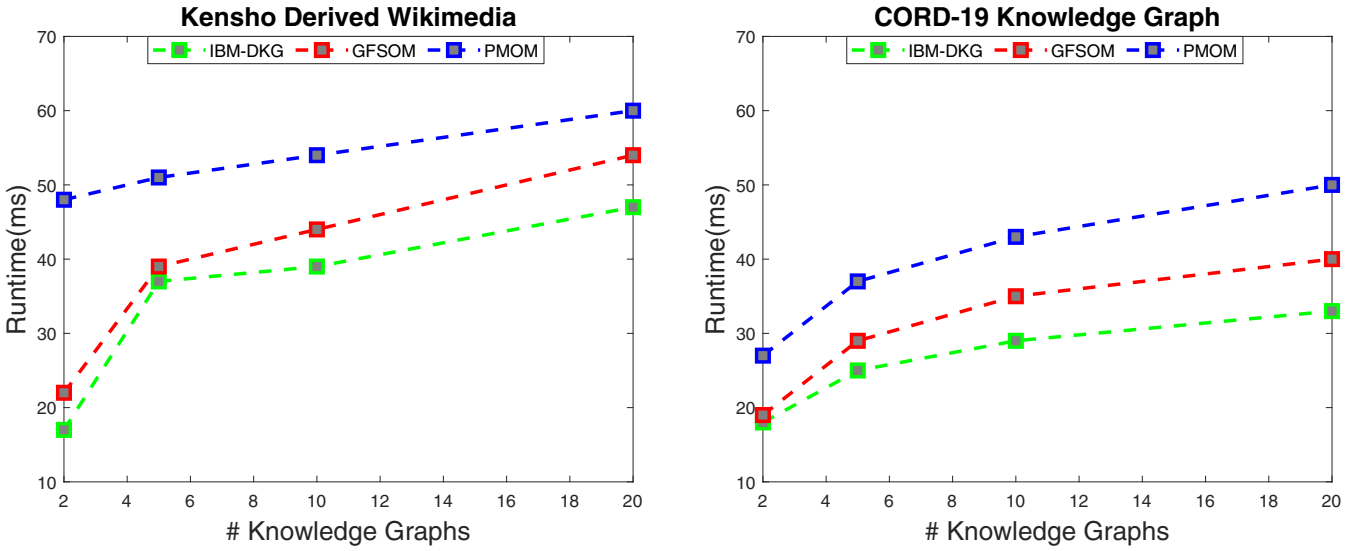

FIGURE 5 Runtime comparison of the IBM-DKG and the state-of-the-art knowledge graph matching solutions by varying the number of knowledge graphs

selection.[49] The idea is to apply the feature selection methods in order to select the relevant concepts and the relations for participating in solving the distributed knowledge graph matching problem. This selection allows to considerably reduce the dimensions of the problem by removing the irrelevant concepts and relations before the matching process.

2. *Explainable matching*: Distributed knowledge graph matching solutions might derive various results from the same set of knowledge graphs. Indeed, different matching may be produced from the same concepts and relation. These results depend to many factors such as the algorithm used during the matching process, and the score function employing for computing the matching score. The problem is to make an explanation of the results in order to decide the better matching to the end-user. A crowd-sourcing may be one way to address this issue, where different knowledge graph matching approaches should work together to identify the best matching. Agents represented by approaches and programs could find locally the matching and send them to the end-user. Another way to address this challenging issue is to use the explainable AI tools[50] in order to compute the contribution of each concept and relation in the final matching, and then decide which criteria should be targeted in order to find the best matching to the end-user.

3. *Missing of ground truth*: Missing of the ground truth is a common problem in evaluating knowledge graph matching algorithms, in particular, for real scenarios, such as IoT applications. As challenges for future research regarding the quality assessment of the knowledge graph matching results, the following issues and research questions remain to be addressed:

   - Defining useful, publicly available benchmark data for IoT problems is beneficial for analyzing the distributed knowledge graph matching algorithms.
   - It would be very useful to identify the meaningful criteria for an internal evaluation of knowledge graph matching. One way to address this challenging issue is to provide unified ranking-function scores to rank the matching. These functions should be independent of the whole process for identifying the best matching.

5 | CONCLUSIONS

This article developed an end-to-end framework for dealing with the distributed knowledge graph matching problem in IoT 5G networks. The framework used both artificial intelligence and blockchain management for accurately find the shared concepts and relations from the set of knowledge graphs. The concepts and the relations of the knowledge graphs are decomposed into similar groups. Instead of exploring the whole concepts, and the relations of the knowledge graphs, only the representative of the groups are compared during the matching process. To certify the validity of the proposed framework, intensive experiments have been carried; the results are very promising in both runtime and accuracy. In addition, the framework outperforms the existing matching algorithms by varying the number of concepts, the number of relations, and the number of knowledge graphs. As future perspective, we plan to investigate in preprocessing step of the knowledge graphs by filtering and/or removing noises, and/or exploring the feature selection process in order to only process the set of relevant concepts of the knowledge graphs. Exploring explainable AI for knowledge graph matching is also an interesting topic that will be considered in our future agenda.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in Kensho Derived Wikimedia Dataset at https://www.kaggle.com/kenshoresearch/kensho-derived-wikimedia-data. These data were derived from the following resources available in the public domain: Kensho Derived Wikimedia Dataset, https://storage.googleapis.com/kaggle-data-sets/469330/909986/bundle/archive.zip?X-Goog-Algorithm=GOOG4-RSA-SHA256%26X-Goog-Credential=gcp-kaggle-com%40kaggle-161607.iam.gserviceaccount.com%2F20210326%2Fauto%2Fstorage%2Fgoog4_request%26X-Goog-Date=20210326T10.

ORCID

Gautam Srivastava https://orcid.org/0000-0001-9851-4103
Jerry Chun-Wei Lin https://orcid.org/0000-0001-8768-9709

REFERENCES

1. Srivastava G, Lin JCW, Jolfaei A, Li Y, Djenouri Y. Uncertain-driven analytics of sequence data in IoCV environments. *IEEE Trans Intell Transp Syst*. 2020.
2. Shafique K, Khawaja BA, Sabir F, Qazi S, Mustaqim M. Internet of things (IoT) for next-generation smart systems: a review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. *IEEE Access*. 2020;8:23022-23040.
3. Serrano W. The blockchain random neural network for cybersecure IoT and 5G infrastructure in smart cities. *J Netw Comput Appl*. 2021;175:102909.
4. Kavitha D, Ravikumar S. IOT and context-aware learning-based optimal neural network model for real-time health monitoring. *Trans Emerg Telecommun Technol*. 2021;32(1):e4132.
5. Lin JCW, Srivastava G, Zhang Y, Djenouri Y, Aloqaily M. Privacy preserving multi-objective sanitization model in 6G IoT environments. *IEEE Internet Things J*. 2020.
6. Esposito C, Ficco M, Gupta BB. Blockchain-based authentication and authorization for smart city applications. *Inf Process Manag*. 2021;58(2):102468.
7. Liang T, Sheng X, Zhou L, et al. Mobile app recommendation via heterogeneous graph neural network in edge computing. *Appl Soft Comput*. 2021;103:107162.

8. Al Ridhawi I, Aloqaily M, Boukerche A, Jararweh Y. Enabling intelligent IoCV services at the edge for 5G networks and beyond. *IEEE Trans Intell Transp Syst*. 2021.
9. Qadir Z, Ullah F, Munawar HS, Al-Turjman F. Addressing disasters in smart cities through UAVs path planning and 5G communications: a systematic review. *Comput Commun*. 2021.
10. Xie C, Yu B, Zeng Z, Yang Y, Liu Q. Multi-layer internet of things middleware based on knowledge graph. *IEEE Internet Things J*. 2020.
11. Abu-Salih B. Domain-specific knowledge graphs: a survey. *J Netw Comput Appl*. 2021;185:103076.
12. Bellini P, Benigni M, Billero R, Nesi P, Rauch N. Km4City ontology building vs data harvesting and cleaning for smart-city services. *J Vis Lang Comput*. 2014;25(6):827-839.
13. Qiu J, Chai Y, Liu Y, Gu Z, Li S, Tian Z. Automatic non-taxonomic relation extraction from big data in smart city. *IEEE Access*. 2018;6:74854-74864.
14. Le-Phuoc D, Quoc HNM, Quoc HN, Nhat TT, Hauswirth M. The graph of things: a step towards the live knowledge graph of connected things. *J Web Semant*. 2016;37:25-35.
15. Berdik D, Otoum S, Schmidt N, Porter D, Jararweh Y. A survey on blockchain for information systems management and security. *Inf Process Manag*. 2021;58(1):102397.
16. Revanesh M, Sridhar V. A trusted distributed routing scheme for wireless sensor networks using blockchain and meta-heuristics-based deep learning technique. *Trans Emerg Telecommun Technol*. 2021;e4259.
17. Dai Y, Xu D, Maharjan S, Chen Z, He Q, Zhang Y. Blockchain and deep reinforcement learning empowered intelligent 5G beyond. *IEEE Netw*. 2019;33(3):10-17.
18. Weng J, Weng J, Zhang J, Li M, Zhang Y, Luo W. Deepchain: auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Trans Depend Secure Comput*. 2019.
19. Liu M, Yu FR, Teng Y, Leung VC, Song M. Performance optimization for blockchain-enabled industrial Internet of Things (IIoT) systems: a deep reinforcement learning approach. *IEEE Trans Ind Inform*. 2019;15(6):3559-3570.
20. Li J, Wang Z, Zhang X, Tang J. Large scale instance matching via multiple indexes and candidate selection. *Knowl-Based Syst*. 2013;50:112-120.
21. Wang Z, Li J, Zhao Y, Setchi R, Tang J. A unified approach to matching semantic data on the web. *Knowl-Based Syst*. 2013;39:173-184.
22. Shao C, Hu LM, Li JZ, Wang ZC, Chung T, Xia JB. RiMOM-IM: a novel iterative framework for instance matching. *J Comput Sci Technol*. 2016;31(1):185-197.
23. Alam M, Recupero DR, Mongiovi M, Gangemi A, Ristoski P. Event-based knowledge reconciliation using frame embeddings and frame similarity. *Knowl-Based Syst*. 2017;135:192-203.
24. Rosaci D. CILIOS: connectionist inductive learning and inter-ontology similarities for recommending information agents. *Inf Syst*. 2007;32(6):793-825.
25. Rosaci D. Finding semantic associations in hierarchically structured groups of web data. *Form Asp Comput*. 2015;27(5-6):867-884.
26. Elmagarmid AK, Ipeirotis PG, Verykios VS. Duplicate record detection: a survey. *IEEE Trans Knowl Data Eng*. 2007;19(1):1-16.
27. Niu X, Rong S, Wang H, Yu Y. An effective rule miner for instance matching in a web of data. Paper presented at: Proceedings of the ACM International Conference on Information and Knowledge Management; 2012:1085-1094; ACM, New York, NY.
28. Cerón-Figueroa S, López-Yáñez I, Alhalabi W, et al. Instance-based ontology matching for e-learning material using an associative pattern classifier. *Comput Hum Behav*. 2017;69:218-225.
29. Ochieng P, Kyanda S. A K-way spectral partitioning of an ontology for ontology matching. *Distrib Parallel Databases*. 2018;36:643-673.
30. Tran QV, Ichise R, Ho BQ. Cluster-based similarity aggregation for ontology matching. *Ontol Match*. 2011;814:142-147.
31. Algergawy A, Massmann S, Rahm E. A clustering-based approach for large-scale ontology matching. Paper presented at: Proceedings of the East European Conference on Advances in Databases and Information Systems. Vienna, Austria; 2011:415-428.
32. Belhadi H, Akli-Astouati K, Djenouri Y, Lin JCW, Wu JMT. GFSOM: genetic feature selection for ontology matching. Paper presented at: Proceedings of the International Conference on Genetic and Evolutionary Computing; 2018:655-660; Springer, New York, NY.
33. Belhadi H, Akli-Astouati K, Djenouri Y, Lin JCW. Exploring pattern mining for solving the ontology matching problem. Paper presented at: Proceedings of the European Conference on Advances in Databases and Information Systems; 2019:85-93; Springer, New York, NY.
34. Qiu C, Yu FR, Yao H, Jiang C, Xu F, Zhao C. Blockchain-based software-defined industrial Internet of Things: a dueling deep Q-learning approach. *IEEE Internet Things J*. 2018;6(3):4627-4639.
35. Liu CH, Lin Q, Wen S. Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning. *IEEE Trans Ind Inform*. 2018;15(6):3516-3526.
36. Dai Y, Xu D, Zhang K, Maharjan S, Zhang Y. Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks. *IEEE Trans Veh Technol*. 2020;69(4):4312-4324.
37. Chai H, Leng S, Chen Y, Zhang K. A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles. *IEEE Trans Intell Transp Syst*. 2020.
38. Lu Y, Huang X, Zhang K, Maharjan S, Zhang Y. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Trans Veh Technol*. 2020;69(4):4298-4311.
39. Qu Y, Gao L, Luan TH, et al. Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE Internet Things J*. 2020.
40. Luo J, Qianbin C, Yu R, Lun T. Blockchain-enabled software-defined industrial internet of things with deep reinforcement learning. *IEEE Internet Things J*. 2020.

41. Yazdinejad A, Srivastava G, Parizi RM, Dehghantanha A, Choo KKR, Aledhari M. Decentralized authentication of distributed patients in hospital networks using blockchain. *IEEE J Biomed Health Inform*. 2020;24(8):2146-2156.
42. Li D, Liu W, Deng L, Qin B. Design of multimedia blockchain privacy protection system based on distributed trusted communication. *Trans Emerg Telecommun Technol*. 2021;32(2):e3938.
43. Pohrmen FH, Das RK, Saha G. Blockchain-based security aspects in heterogeneous Internet-of-Things networks: a survey. *Trans Emerg Telecommun Technol*. 2019;30(10):e3741.
44. Yahiatene Y, Rachedi A, Riahla MA, Menacer DE, Nait-Abdesselam F. A blockchain-based framework to secure vehicular social networks. *Trans Emerg Telecommun Technol*. 2019;30(8):e3650.
45. Belhadi A, Djenouri Y, Djenouri D, Michalak T, Lin JCW. Machine learning for identifying group trajectory outliers. *ACM Trans Manag Inf Syst*. 2021;12(2):1-25.
46. Belhadi A, Djenouri Y, Srivastava G, Djenouri D, Lin JCW, Fortino G. Deep learning for pedestrian collective behavior analysis in smart cities: a model of group trajectory outlier detection. *Inf Fusion*. 2021;65:13-20.
47. Belhadi A, Djenouri Y, Djenouri D, Michalak T, Lin JCW. Deep learning versus traditional solutions for group trajectory outliers. *IEEE Trans Cybern*. 2020.
48. Djenouri Y, Belhadi A, Lin JCW, Djenouri D, Cano A. A survey on urban traffic anomalies detection algorithms. *IEEE Access*. 2019;7:12192-12205.
49. Belhadi H, Akli-Astouati K, Djenouri Y, Lin JCW. Data mining-based approach for ontology matching problem. *Appl Intell*. 2020;1-18.
50. Gaur M, Faldu K, Sheth A. Semantics of the black-box: can knowledge graphs help make deep learning systems more interpretable and explainable? *IEEE Internet Comput*. 2021;25(1):51-59.

How to cite this article: Djenouri Y, Srivastava G, Belhadi A, Lin JC-W. Intelligent blockchain management for distributed knowledge graphs in IoT 5G environments. *Trans Emerging Tel Tech*. 2021;e4332. https://doi.org/10.1002/ett.4332