

RESEARCH

Open Access



Defect knowledge graph construction and application in multi-cloud IoT

Wenqing Yang¹, Xiaochao Li^{2*}, Peng Wang¹, Jun Hou³, Qianmu Li² and Nan Zhang¹

Abstract

As the State Grid Multi-cloud IoT platform grows and improves, an increasing number of IoT applications generate massive amounts of data every day. To meet the demands of intelligent management of State Grid equipment, we proposed a scheme for constructing the defect knowledge graph of power equipment based on multi-cloud. The scheme is based on the State Grid Multi-cloud IoT architecture and adheres to the design specifications of the State Grid SG-EA technical architecture. This scheme employs ontology design based on a fusion algorithm and proposes a knowledge graph reasoning method named GRULR based on logic rules to achieve a consistent and shareable model. The model can be deployed on multiple clouds independently, increasing the system's flexibility, robustness, and security. The GRULR method is designed with two independent components, Reasoning Evaluator and Rule Miner, that can be deployed in different clouds to adapt to the State Grid Multi-cloud IoT architecture. By sharing high-quality rules across multiple clouds, this method can avoid vendor locking and perform iterative updates. Finally, the experiment demonstrates that the GRULR method performs well in large-scale knowledge graphs and can complete the reasoning task of the defect knowledge graph efficiently.

Keywords: Multi-cloud IoT, Defect knowledge graph, Ontology design, Knowledge graph reasoning

Introduction

With the rapid development of Internet of Things (IoT) technology and the increasing popularity of IoT devices, there are more and more computing-intensive IoT applications [1–3]. However, due to the limited hardware and software resources of IoT devices, it is difficult to meet the requirements of localized data processing [4, 5]. Cloud computing is a popular solution to this problem. It can provide resources in a low-cost and flexible way [6, 7]. However, relying on a single cloud to provide all resources and services for IoT users is difficult [8]. The multi-cloud architecture has gained more attention and application than the single cloud architecture due to its greater flexibility, robustness, and avoidance of service

resource locking [9–13]. Therefore, State Grid provides services for IoT users through multi-cloud.

Every day, the State Grid IoT devices produce a considerable volume of heterogeneous data [14]. Along with business information and equipment monitoring data, these data also include basic equipment information [15]. Currently, the operation and maintenance staff mostly fill out and upload the equipment defect inspection data to the cloud platform after fixing the site's flaws. Manual entry will result in omitted and incomplete data [16, 17]. Additionally, there are numerous different types of devices that are extensively used. Dealing with these heterogeneous data in a large-scale, real-time manner is still quite challenging [18, 19]. We provide a scheme to construct an equipment defect knowledge graph in order to address the aforementioned issues and intelligently manage the equipment in the multi-cloud scenario of the State Grid IoT [20]. The scheme adheres to the State Grid SG-EA technical architectural design specification and is based on the State Grid Multi-cloud Architecture

*Correspondence: lixiaochao@njjust.edu.cn

² School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

Full list of author information is available at the end of the article

of the IoT. The multi-layer technology system design is implemented using dynamic software technology based on components and the consistent and shareable data model.

Ontology design, knowledge extraction, knowledge fusion, and knowledge reasoning are all included in our KG construction approach [21, 22]. The KG of equipment defects, which is based on the integration of various business system equipment databases, expert experience databases, defect data, defect reports, standards, specifications, rules, and regulations, etc., provides intelligent and individualized power equipment defect knowledge services for professional equipment management [23, 24]. Offer a smart multi-round question and answer (Q &A) service for equipment problems for the queries entered by users. The Q &A's major focus is on knowledge Q &A related to common defect business scenarios, such as equipment defect cases, defect categorization and judgment bases, topology information of the equipment grid, defect treatment procedures, etc. Our proposed scheme is based on natural language processing technology, which is superior to existing knowledge graph construction methods in that it can intelligently process heterogeneous data in a multi-cloud scenario. To increase the effectiveness of data processing in a multi-cloud scenario, the construction process utilizes a fully automated construction method without manual involvement (Fig. 1).

A logic rule-based KG reasoning method called GRULR is suggested in order to efficiently harness the value of the defect knowledge graph. This approach uses maximum likelihood estimation to update the network parameters of the Gate Recursive Unit (GRU) and posterior probability to identify high-quality logic rules. For iterative training, the Expectation Maximization (EM) technique is employed. Actually, there are many ways to learn logic rules from the KG [25]. Most traditional methods, such as path sorting algorithm

[26], ProPPR [27], and Markov logic network [28], enumerate the relationship paths on the graph as candidate logic rules and then learn the weight of each candidate logic rule according to the algorithm to evaluate the rule quality. Recently, some researchers have put forward some methods based on neurologic programming, which can learn logic rules and their weights in a differentiable way at the same time [29, 30]. Although these methods are effective for prediction empirically, their search space is exponential, so it is difficult to identify high-quality logic rules. In addition, some studies describe the learning problem of logic rules as a continuous decision-making process and use reinforcement learning [31] to search for logic rules, which greatly reduces the complexity of searching. However, due to the large action space and low returns in the learning process, the effect of extracting logical rules by these methods is still not satisfactory.

In summary, our contributions are as follows:

- We examined the power equipment defect knowledge, enhanced and refined the architecture of the ontology, and provided a knowledge graph construction strategy appropriate for the multi-cloud State Grid IoT scenario.
- We proposed a flexible, reliable, and effective KG reasoning approach GRULR, which is ideal for multi-cloud State Grid IoT applications.
- We created and executed the power equipment defect KG application in the context of the multi-cloud State Grid IoT scenario, which offers suggestions for the use and innovation of KG in the power industry.

Table 1 explains the abbreviations in the figure and the symbols in the formula, making it easier to read this paper.

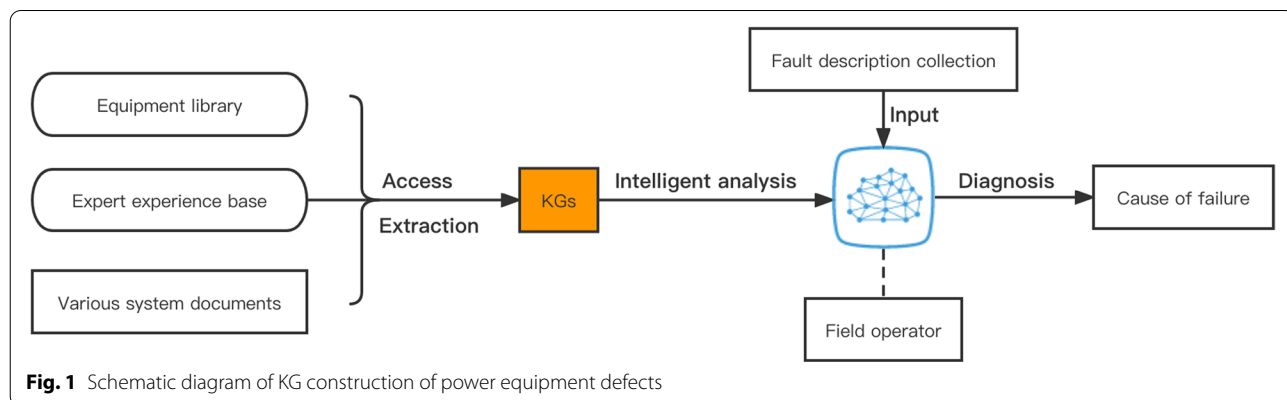


Table 1 Symbols and their representations in the proposed approaches

Symbol	Representation
Q & A	question and answer
DB	database
ES	elasticsearch
OSS	Object Storage Service
MOM	Message Oriented Middleware
ISC	Internet Service Customer
S6000	a network equipment
$r(X_0, X_n)$	the relationship between X_0 and X_n
G	the knowledge graph
P_d	the representation of training data
q	the question to be queried
a	the inference result
p_w	the probability of an answer conditioned defined by Reasoning Evaluator
p_θ	the prior probability of an answer conditioned defined by Rule Miner
R	the representation of rule set
E	the maximum likelihood estimation
w_{rule}	the weight of the rule
$score_w$	the score of candidate answer in set A
exp	the representation of exponential function in softmax
log	the representation of log function
H	the score of rules' quality
t	the representation of tail entity of knowledge triple
A	the set of candidate answers
MD	the representation of polynomial distribution
N	the sampling number parameter in polynomial distribution
GRU	the representation of Gate Recurrent Unit
f	the representation of explicit function mapping used in GRU network
K	the number of high-quality rules to output
$\max \mathcal{O}$	the representation of maximizing the loglikelihood of the ruleset

Related Work

The construction of KG needs to be applied to various information processing technologies. Knowledge extraction, which extracts knowledge from various data sources and stores it in the KG, is the foundation of building a large-scale KG [32, 33]. Knowledge fusion can solve the heterogeneity of different KGs. Through knowledge fusion, the heterogeneous KGs of different data sources can be connected and operated with each other, thus improving the quality of KGs. Knowledge calculation is the main output capacity of KGs, and knowledge reasoning is one of the most important capabilities [10, 34]. Knowledge extraction has gone through three stages: manual rule writing, traditional machine learning, and deep learning, which are mainly divided into named entity identification and relation extraction [14]. Knowledge fusion eliminates the ambiguity of the processes, and integrates heterogeneous

and diverse knowledge from different data sources in the same framework, thus realizing the fusion of data, information, and other perspectives [18, 35]. The core of knowledge integration is the generation of mapping. At present, knowledge fusion techniques can be divided into ontology fusion and data fusion. Our KG construction method combines the characteristics of the State Grid IoT cloudy environment and the requirements for bandwidth and computing resources, and can well complete the task of building the domain KG.

The work of this paper is related to the existing KG reasoning work based on logic rules. In early methods, the relationship path between entities was used as candidate logic rules to enumerate, and the quality was evaluated by learning the weight of the rules [36, 37]. Some recent methods learn logic rules and weights in a differentiable way at the same time. These methods are mainly based on neural networks and transform the feature distribution of input data from the original space to another feature

space through nonlinear transformation, and automatically learn the feature representation [9, 25]. The method in this paper is similar to these methods in thought, and its purpose is to learn logic rules and weights. The main innovation of this method is to divide the model into two parts: Rule Miner and Reasoning Evaluator. Rules and weights are studied separately. The search efficiency is high, it is suitable for the cloud deployment of IoT, and it also has a good reasoning effect.

Tensor decomposition is a process of decomposing high-dimensional arrays into several low-dimensional matrices. Using tensor decomposition to learn the embedding of entities and relations has always been a hot research direction, but these methods often only learn some simple logic rules [38–40]. Compared with this method, this method can discover more complicated logic rules.

KG Construction

System scheme

As shown in Fig. 2, the architecture of the power equipment defect KG system is divided into four layers, including the data layer, support layer, service layer, and application layer from bottom to top. The data layer is based on the State Grid multi-cloud platform, including graph database, third-party relational database, cache database, object storage services, etc. The supporting layer realizes the basic functions of KG construction,

such as knowledge computing, entity alignment, knowledge modeling, knowledge storage, knowledge error correction, and so on. The service layer provides a graphic analysis model, system log, query service, rule setting, document management, etc. The application layer mainly provides external services, such as knowledge Q &A, error analysis, defect intelligent retrieval, and so on, based on a power equipment defect KG.

The data layer is abstracted on the State Grid multi-cloud platform, and various database data sources are integrated to provide services to the upper layer. This design improves the system’s flexibility and security. It strengthens deployment and expansion in multi-cloud scenarios.

As shown in Fig. 3, the technical architecture of this scheme is based on the State Grid multi-cloud architecture of the IoT and follows the design specification of the State Grid SG-EA technical architecture [40]. This scheme makes use of component and dynamic software technology, along with a consistent and shareable data model. The visualization layer, application layer, service layer, security layer, data layer, and so on are all part of the multi-layer technology system. Through the application integration of the basic support platform, the scheme realizes cooperation and integration of the interface components of the State Grid’s work business in the enterprise, as well as reuse to meet business requirements.

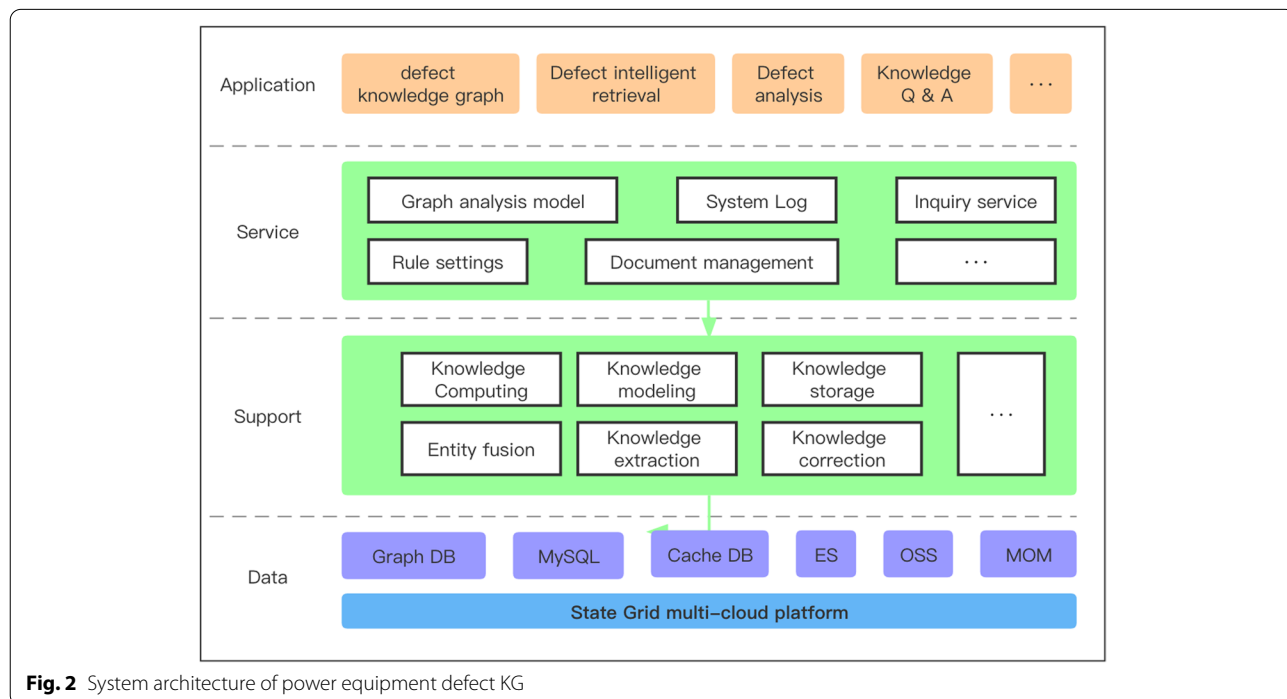


Fig. 2 System architecture of power equipment defect KG

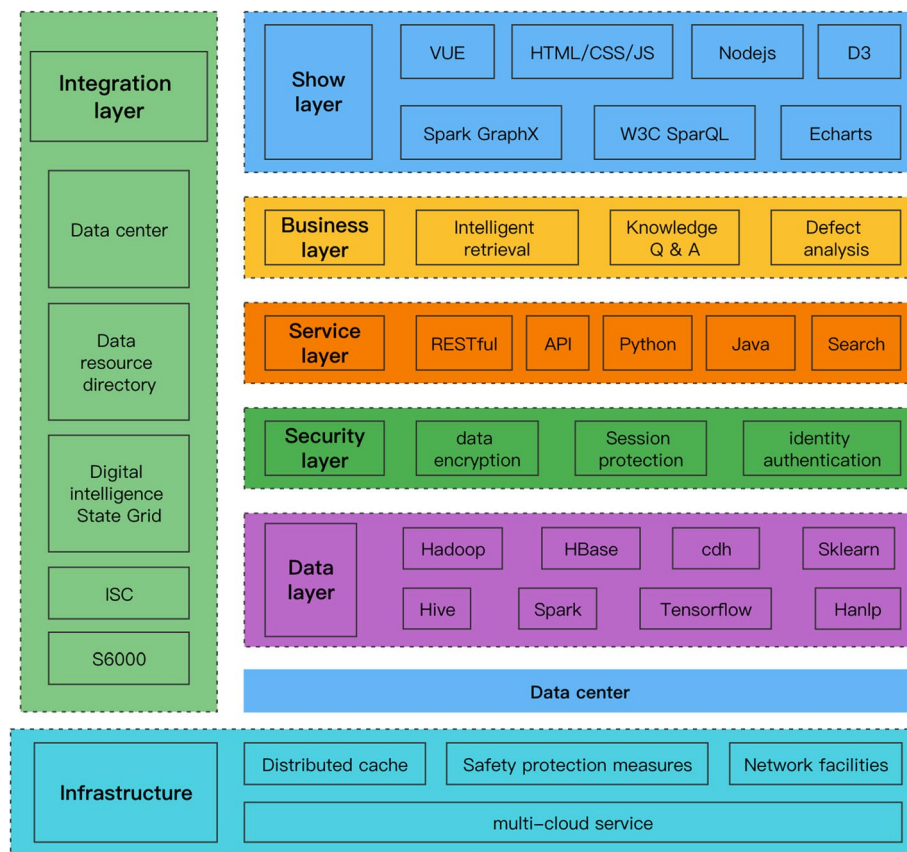


Fig. 3 Technical framework of power equipment defect KG

Ontology design

Ontology mainly describes domain knowledge, provides strong support for common understanding, determines commonly recognized vocabulary in the domain, and formally gives a clear definition of vocabulary and the relationship between vocabulary [29]. That is to say, ontology is the abstract expression of knowledge and the top-level structure for describing knowledge. Ontology design refers to the establishment of the data model of the KG, that is, the concept, attribute, and relationship among concepts of the ontology model. Generally, knowledge modeling includes the methods based on industry standardization models and the “top-down” method gradually refining from the top concept; As well as a “bottom-up” method based on entity induction to form bottom-level concepts and gradually abstract upwards. The ontology design of the equipment defect KG adopts the method of combining both. The top-down ontology design requires expert intervention in the early stages, and the bottom-up ontology design is updated using natural language processing technology in the later stages. In comparison to other ontology design schemes, our scheme combines

the benefits of both design patterns and significantly improves system efficiency.

At present, the results of ontology design are based on business data such as defect records, defect standard library, defect analysis, common code table, and documents such as standards, guides, and reports. In the system, 25 concepts, including defects, defect properties, equipment, components, parts, power stations/lines, manufacturers, people, etc., were constructed, and 131 relationships were defined. Currently, the system has 10493 entities and 152600 attributes. As shown in Fig. 4, taking equipment defects as the core, referring to the basic ontology design concepts of “people, events, objects, rules”, and combining the equipment defect business and model, the equipment defect ontology is constructed with the main vertex concepts such as unit, equipment, defect standard, defect description, defect treatment, defect elimination and acceptance, and family defects [41].

Knowledge extraction and fusion

As shown in Fig. 5, in the process of building the defect KG, it is necessary to clean and mark all kinds



Fig. 4 Example of ontology design for power equipment defect KG

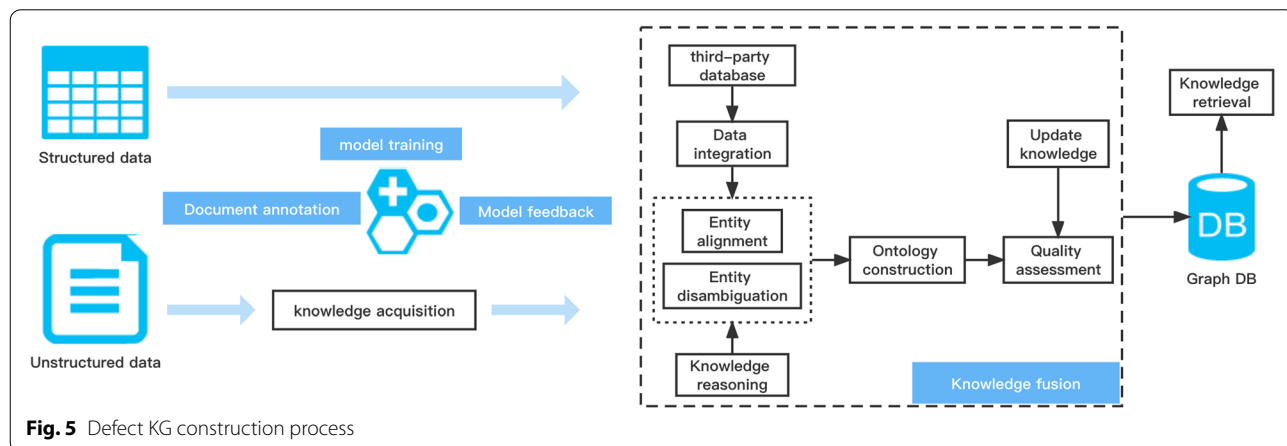


Fig. 5 Defect KG construction process

of equipment defect data collected and visited. The goal of data cleaning is to filter and delete duplicate and redundant data, supplement missing data, and correct or delete wrong data. The goal of data annotation is to turn the original data into usable data for the algorithm. According to the type, it can be generally divided into text, image, audio, and video annotation. Based on the data from different sources and structures such as equipment defect data, standard specifications, system guidelines, defect reports, etc. The equipment map object, equipment event definition, and equipment history information are formed by using various methods such as information extraction, D2R conversion, graph mapping, etc.

Knowledge fusion mainly includes the fusion of the device data pattern layer (concept, the relationship between concepts and attributes of concept) and the device data layer. Data layers are the key to the fusion of equipment KGs. The data of equipment defect KGs comes from multiple business systems and offline documents, so it is necessary to realize the efficient real-time integration of massive equipment defect data from different sources and forms. Knowledge retrieval or knowledge reasoning is mainly used for the completion and quality inspection of KG, that is, using the existing facts or relationships to infer new facts or relationships. The methods mainly include graph mining calculation, ontology reasoning, and rule-based reasoning.

KG reasoning

This section introduces a knowledge graph reasoning method that probabilistically formalizes knowledge graph reasoning. The definition of the probability model we use is derived from RNNLogic [39]. The theory behind the model was proved in the original paper. To accommodate the unique requirements of the multi-cloud State Grid IoT scenario, we revised the probability model. Reasoning Evaluator and Rule Miner are designed as two independent models that may be deployed separately to adapt to the state grid multi-cloud architecture. The system can be made flexible and dependable while avoiding supplier locking by performing iterative updates by sharing high-quality rules among multi-cloud.

In this paper, logic rules are expressed in the form of first-order logical conjunction operators, namely $\forall \{X_i\}_{i=0}^n r(X_0, X_n) \leftarrow r_1(X_0, X_1) \wedge r_2(X_1, X_2) \wedge \dots \wedge r_n(X_{n-1}, X_n)$. This representation can not only facilitate the processing of anti rules $\neg r$ and inverse rules r^{-1} but also displays the structure of the KG. Model the KG reasoning, and represent the training data as $P_d(G, q, a)$. This paper refers to the definition of the probability distribution model defined by RNNLogic [39]:

$$p_{w,\theta}(a | G, q) = \sum_R p_w(a | G, q, R) p_\theta(R | q) = \mathbf{E}_{p_\theta(R|q)} [p_w(a | G, q, R)] \quad (1)$$

where G is the KG, the set of knowledge triples (h, r, t) , q is the question to be queried, and a is the inference result. The purpose is to infer the possible answer based on a given R and q to establish a probability distribution model $p(a | G, q)$. The Rule Miner defines a prior p_θ on latent rules R conditioned on a query q . At the same time, the Reasoning Evaluator gives a probability p_w on an answer conditioned on the latent rule R , the query q , and the KG G .

The Rule Miner and Reasoning Evaluator are jointly trained by maximizing the likelihood of the sample, and the objective function is defined as follows:

$$\max_{\theta, w} \mathcal{O}(\theta, w) = \mathbf{E}_{(G, q, a) \sim p_d} [\log p_{w,\theta}(a | G, q)] \quad (2)$$

Reasoning Evaluator Algorithm

The Reasoning Evaluator is defined as $p_w(a | G, q, R)$ which constructs a Markov logic network [27] according to the logic rules and predicts the answer a by reasoning through a group of logic rules R on the KG G . For each query question, $q = (h, r, ?)$, different paths on the KG G can be found by combining these rules, inferring differentiable candidate answers. Any answer deduced from these logical rules constitutes set A . For each answer a in the set, the score can be calculated by the following formula:

$$\text{score}_w(a) = \sum_{\text{rule} \in R} w_{\text{rule}} \quad (3)$$

To simplify the calculation, this paper sets the path's weight to 1, so it does not participate in the analysis. w_{rule} is the weight of the rule. Each rule in the set will have a contribution score to the reasoning result, and the score is accumulated to represent the answer's score. Calculate the possibility of each entity as the answer through *softmax*:

$$p_w(a = t | G, q, R) = \frac{\exp(\text{score}_w(a))}{\sum_{a' \in A} \exp(\text{score}_w(a'))} \quad (4)$$

Candidate entities with higher scores can be considered as reasoning answers. In addition, the high-quality rules need to be identified from those given rules, and the following formula can calculate a score H :

$$H(\text{rule}) = \text{score}_w(t | \text{rule}) + \log \text{GRU}_\theta(\text{rule} | q) \quad (5)$$

In the above formula, $\log \text{GRU}_\theta(\text{rule} | q)$ is the prior probability of the Rule Miner, and the rules with higher H

scores are given to the Rule Miner as high-quality rules. The posterior probability combines the knowledge of the Rule Miner and the Reasoning Evaluator. Therefore, the high-quality logical rules can be discovered using a posteriori probability sampling. Calculate the probability distribution of those rules through *softmax*:

$$p_{rule} = \frac{\exp(H(rule))}{\sum_{rule' \in R} \exp(H(rule'))} \quad (6)$$

Rule Miner Algorithm

For the Rule Miner p_θ , this paper focuses on the relation r in the KG instead of the entity. To answer a query $q(h, r, ?)$, these logical rules can be expressed as the following sequence relationship $[r_q, r_i, \dots, r_e]$, where r_q is the query relationship, r_e represents the end of the rule, and r_i is the reasoning logic rules. Therefore, The Rule Miner generates logical rules based on the GRU network for reasoning to answer queries. To calculate the generation probability of rules, the distribution of generation rule set R is defined as the following polynomial distribution:

$$p_\theta(R | q) = MD(R | N, GRU_\theta(q)) \quad (7)$$

Where MD represents the polynomial distribution, N represents the size of the set R , and $GRU_\theta(q)$ stands for the distribution over a group of rules for query q .

The process of generating the ruleset R is very specific because the GRU network can output N rules to form R . The GRU network generates the following relation based on the currently given sequence relation and gives the probability for generating the relation. The GRU network used is defined as follows:

$$\begin{aligned} h_0 &= f_1(v_r) \\ h_t &= \mathbf{GRU}(h_{t-1}, f_2([v_r, v_{rt}])) \end{aligned} \quad (8)$$

Where h_0 is the hidden layer of the initial state, h_t is the hidden layer of the t state, f_1 and f_2 are linear transformation functions, v_r is the embedding vector of the head relationship of the query q , v_{rt} is the embedding vector of the relationship with v_r , $[v_r, v_{rt}]$ is vector concatenation operation. The probability of using *softmax* to give the next state relation r_{t+1} is:

$$\mathit{softmax}(f_3(h_{t+1})) \quad (9)$$

Assuming that the relationship set is denoted as R , transform h_{t+1} into $|R|$ dimensional vector by linear transformation f_3 , and then apply the *softmax* function to the $|R|$ dimensional vector to obtain the probability of each relation and generate r_{t+1} .

Finally, K high-quality rules R_k are output by the Reasoning Evaluator as part of the trained data, update the

Rule Miner parameter θ by maximizing the log-likelihood of the ruleset:

$$\max_{\theta} \mathcal{O}(\theta) = \sum_{rule \in R_k} \log GRU_\theta(rule | q) \quad (10)$$

Experiments

Datasets

This paper selects four datasets for experiments, including two small datasets, KINSHIP and UMLS [42], and two large datasets, WN18RR and FB15K-237 [43]. The KINSHIP dataset consists of names and kinships of persons in two families in Australia. The UMLS dataset comes from the biomedical domain, where biomedical concepts (e.g., disease, drug) are entity data, and treatments and diagnoses are relation data. WN18RR is a subset extracted from WordNet. FB15K-237 is a subset drawn from the Freebase subset FB15K. For each dataset, we randomly select 50% of the dataset as the training set, 30% as the test set, and the remaining 20% as the validation set. The statistics of each dataset are shown in Table 2.

Evaluation Criteria

To comprehensively evaluate the effectiveness of our model, we adopt several representative evaluation metrics [28]: HITS@N, Mean Reciprocal Rank (MRR).

- The formula for calculating HITS@N is:

$$HITS@N = \frac{1}{|S|} \sum_{i=1}^{|S|} \mathbf{I}(rank_i \leq N) \quad (11)$$

Where S is the triplet set, $|S|$ is the number of triplet sets, and $rank_i$ refers to the inference prediction rank of the i -th triplet. The HITS@N indicator refers to the average proportion of results ranking less than or equal to N in the inference prediction. The larger the value, the better. In this paper, the value of N is 1, 3, and 10.

- The formula for calculating MRR is:

Table 2 Statistics of each data set

Dataset	Entities	Relations	Triples	Mean degree
KINSHIP	104	25	8544	85.15
UMLS	135	46	5216	38.63
WN18RR	40945	11	86835	2.19
FB15K-237	14505	237	272115	19.74

$$MRR = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{1}{rank_i} \tag{12}$$

MRR is an internationally used mechanism for evaluating search algorithms, the first result matches with a score of 1, the second matches with a score of 1/2, and the nth matches with a score of 1/n. If there is no match, the score is 0. The final score is the sum of all scores. The larger the value, the better, indicating that the inference result conforms to the facts.

Benchmark methods

We compare the proposed model GRULR with several state-of-the-art methods. Benchmark methods are divided into three categories: embedding-based, rule-based logic, and reinforcement learning-based methods.

- Embedding-based representative methods: DistMult, ComplEx, and ConvE. ConvE and ComplEx utilize the method proposed by Dettmers et al. [44].
- Representative methods based on logical rules: NLP and DRUM [45].
- a usual method based on reinforcement learning: MINERVA [31].

First, Table 3 shows the experimental comparison results of our method and other benchmark methods on standard datasets. As shown in Table 3, on the large datasets WN18RR and FB15K-237, all inference ability and accuracy drop significantly, such as embedding-based methods (ComplEx, DistMult, and ConvE) MRR metrics drop by more than 50%. However, the HITS@1,

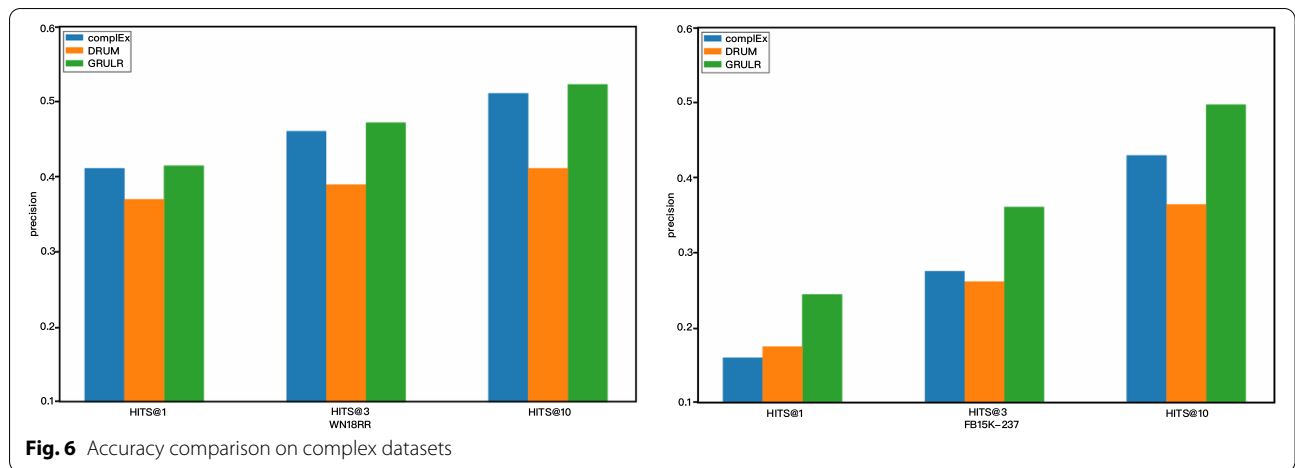
3, 10, and MRR results of the method proposed in this paper are generally higher than other benchmark methods. The MRR metrics of GRULR on the large datasets WN18RR and FB15K-237 reach 0.450 and 0.329, respectively, and the HITS@3 metrics also get 0.471 and 0.360. In conclusion, the model GRULR proposed in this paper outperforms the other six methods on the large datasets WN18RR and FB15K-237, mainly because GRULR can jointly optimize the Rule Miner and Reasoning Evaluator to generate high-quality logical rules for KG reasoning. In addition, the evaluation criteria HITS@N and MRR of the GRULR model are slightly lower than the embedding-based methods (i.e., ComplEx, DistMult, and ConvE) on the small datasets KINSHIP and UMLS. This is mainly because these datasets are not designed to test the model’s ability to learn logical rules. Considering its small scale and simple logic rules, the embedding-based model can achieve high performance. Furthermore, the GRULR model outperforms both rule-based learning methods (NLP, DRUM) and reinforcement learning-based methods (MINERVA) on small and large datasets. Based on the above experimental results, the method proposed in this paper is more suitable for complex KG reasoning on large datasets.

As shown in Fig. 6, we compare the HITS@N metric changes of three different methods on the large datasets WN18RR and FB15K-237. In Fig. 6, the proposed models GRULR, DRUM, and ComplEx all increase as N increases. Moreover, the growth rate of GRULR and ComplEx on the WN18RR dataset is relatively fast. In conclusion, the model GRULR proposed in this paper outperforms the other two methods (DRUM and

Table 3 Inference results on common datasets

Dataset	Metric	ComplEx	DistMult	ConvE	NeuralLP	DRUM	MINERVA	GRULR
KINSHIP	HITS@1	0.754	0.808	0.697	0.475	0.183	0.605	0.568
	HITS@3	0.910	0.942	0.886	0.707	0.378	0.812	0.824
	HITS@10	0.980	0.979	0.974	0.912	0.675	0.924	0.912
	MRR	0.838	0.878	0.797	0.619	0.335	0.720	0.715
UMLS	HITS@1	0.823	0.916	0.894	0.643	0.358	0.728	0.730
	HITS@3	0.962	0.967	0.964	0.869	0.699	0.900	0.893
	HITS@10	0.995	0.992	0.992	0.962	0.854	0.968	0.963
	MRR	0.894	0.944	0.933	0.778	0.548	0.825	0.814
WN18RR	HITS@1	0.410	0.390	0.400	0.368	0.369	0.413	0.412
	HITS@3	0.460	0.440	0.440	0.386	0.388	0.456	0.471
	HITS@10	0.510	0.490	0.520	0.408	0.410	0.513	0.522
	MRR	0.440	0.430	0.430	0.381	0.382	0.448	0.450
FB15K-237	HITS@1	0.158	0.155	0.237	0.173	0.174	0.217	0.245
	HITS@3	0.275	0.263	0.356	0.259	0.261	0.329	0.360
	HITS@10	0.428	0.419	0.501	0.361	0.364	0.456	0.497
	MRR	0.247	0.241	0.325	0.237	0.238	0.293	0.329

Bold numbers are the results obtained by the best method in the table



CompIEx) on the large datasets WN18RR and FB15K-237, mainly because GRULR can generate high-quality logic rules to ensure the accuracy of inference results. The above experimental results further demonstrate that the method proposed in this paper is more suitable for complex KG reasoning on large datasets.

Table 4 shows some of the logic rules generated by the GRULR method on the FB15K-237 dataset in the multi-cloud IoT scenario. It can be seen that these rules are meaningful and diverse, and some of them are complex multi-hop reasoning. Experiments show that our method can generate high-quality logic rules. The high-quality logic rules are logic rules based on human natural understanding. For example, if actor X was born in the U state and the U state belongs to the country Y, it

is easy to get the nationality of actor X as Y. As shown in Table 4, the logical inference is represented by symbols: ActorNationality(X, Y) \leftarrow BornIn(X, U) \wedge StateContain⁻¹(U, Y). These meaningful logical rules are found through our model, so we conclude that GRULR can generate high-quality logical rules. We analyze the length of the rule paths that can be learned by the three types of techniques and finds that the size of the rule paths known by the GRULR method is more extended, which is more effective for KG reasoning in the multi-cloud IoT scenario and can effectively mine potential knowledge data.

In Fig. 7, we used four RTX 3090 GPUs to simulate the training of the FB15k-237 dataset under the MEC scenario, completed five iterations of training within

Table 4 Some rules found by GRULR on the FB15K-237 dataset

AppearsInRankedLists(X, Y) \leftarrow ArwardOf(X, Y)

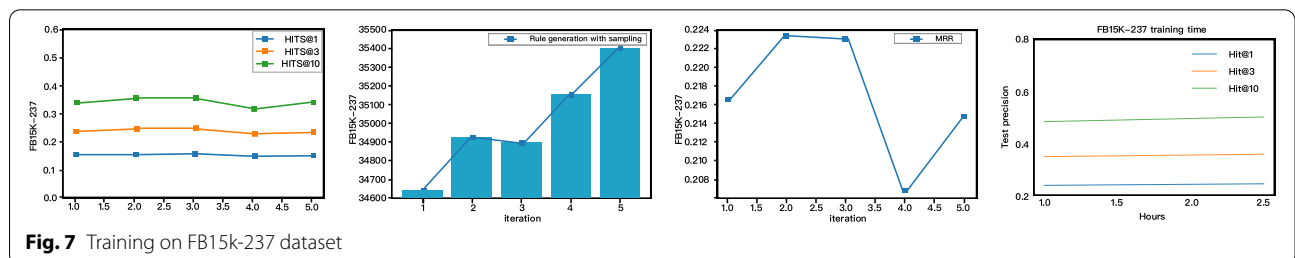
AppearsInRankedLists(X, Y) \leftarrow ProgramCreatorBy(X, U) \wedge DirectBy(U, V) \wedge ArwardOf(V, Y)

ActorNationality(X, Y) \leftarrow BornIn(X, U) \wedge StateContain⁻¹(U, Y)

ActorNationality(X, Y) \leftarrow PlaceOfBirth(X, U) \wedge TownInCity(U, V) \wedge CityInState(V, W) \wedge ActorOf(W, K) \wedge FilmReleaseRegionIn(K, L) \wedge LanguageOf(L, Y)

TvProgramLanguage(X, Y) \leftarrow TvCountryOfOrigin(X, U) \wedge CountryOfficialLanguage(U, Y)

TvProgramLanguage(X, Y) \leftarrow ActorOf(X, U) \wedge FilmReleasIn(U, V) \wedge CountryOfficialLanguage(V, Y)



two hours, and achieved good results. In fact, due to the high efficiency of our training method on large KGs, the initial iteration can achieve better KG reasoning results. As shown in Fig. 7, because newly discovered high-quality logic rules are added after each iteration, the overall prediction effect of GRULR is slowly improving. At the same time, we also noticed that the number of rules sampled in the third iteration decreased, resulting in a decrease in thrust performance. According to the above experimental analysis, we know that the number of rules sampled significantly impacts the reasoning effect, so how to optimize and generate more high-quality logic rules will be the focus of our future research.

In the field of electric power, the GRULR method is applied to cognitive reasoning and intelligent knowledge service in multi-cloud scenarios. In the application of electric power knowledge question answering, the accuracy of knowledge association reasoning increases by 18.3%, and 23.1% improves the combined search performance; in the application of equipment knowledge base, the accuracy of knowledge association increases by 15.4%, and the efficiency of robot response increases by 21.2%, reducing manual pressure in business scenarios and promoting the reasoning, analysis, and application of knowledge. Increase the business system's intelligence and practicality.

Conclusion

This paper proposes a power equipment defect KG construction scheme. This scheme realizes intelligent management of the State Grid IoT's big data and reduces the artificial burden. Simultaneously, a logic-based KG reasoning method GRULR is proposed. The experimental results demonstrate that this method can perform KG reasoning in a multi-cloud environment. With the increasing data scale of the State Grid IoT, the main research direction in the future will be how to perform efficient KG reasoning on super large data sets.

Acknowledgements

Not applicable.

Authors' contributions

Wenqing Yang and Peng Wang completed the construction of the knowledge graph, Xiaochao Li wrote the main manuscript text, Jun Hou and Nan Zhang prepared figures and tables and Qianmu Li approved the final paper to submit. All authors reviewed the manuscript.

Funding

This research was supported by the Fund Project of Research on Basic Capability of Multi-modal Cognitive Graph(NO.524608210192). This research was also supported by the 2022 Jiangsu Province Major Project of Philosophy and Social Science Research in Colleges and Universities(2022SJDZS011), "Research on the Construction of Ideological and Political Selective Compulsory Courses in Higher Vocational Colleges", the Research Project of Nanjing Vocational University of Industry Technology(2020SKYJ03), the Fundamental Research Fund for the Central Universities(30920041112), the

Think Tank Special Project of Nanjing University of Science and Technology (NO.2022TB101).

Availability of data and materials

Not applicable.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹State Grid Electric Power Research Institute, NARI Group Co., Ltd, Nanjing, China. ²School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. ³School of Social Science, Nanjing Vocational University of Industry Technology, Nanjing, China.

Received: 29 July 2022 Accepted: 18 September 2022

Published online: 01 October 2022

References

- Huang J, Tong Z, Feng Z (2022) Geographical POI recommendation for Internet of Things: A federated learning approach using matrix factorization. *Int J Commun Syst* 1:e5161. <https://doi.org/10.1002/dac.5161>
- Xu J, Li D, Gu W, Chen Y (2022) UAV-assisted task offloading for IoT in smart buildings and environment via deep reinforcement learning. *Build Environ* 222(1):109218. <https://doi.org/10.1016/j.buildenv.2022.109218>
- Zhou X, Liang W, Li W, Yan K, Shimizu S, Wang KIK (2021) Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System. *IEEE Internet Things J* 9(12):9310–9319. <https://doi.org/10.1109/JIOT.2021.3130434>
- Gu R, Chen Y, Liu S, Dai H, Chen G, Zhang K et al (2021) Liquid: Intelligent Resource Estimation and Network-Efficient Scheduling for Deep Learning Jobs on Distributed GPU Clusters. *IEEE Trans Parallel Distrib Syst* 33(11):2808–2820. <https://doi.org/10.1109/TPDS.2021.3138825>
- Dai H, Wang X, Lin X, Gu R, Shi S, Liu Y et al (2021) Placing wireless chargers with limited mobility. *IEEE Trans Mob Comput* 1:1. <https://doi.org/10.1109/TMC.2021.3136967>
- Xue X, Wang S, Zhang L, Feng Z, Guo Y (2019) Social learning evolution (SLE): computational experiment-based modeling framework of social manufacturing. *IEEE Trans Ind Inform* 15(6):3343–3355
- Zhou D, Xue X, Zhou Z (2022) SLE2: The improved Social Learning Evolution Model of Cloud Manufacturing Service Ecosystem. *IEEE Trans Ind Inform* 1:1. <https://doi.org/10.1109/TII.2022.3173053>
- Qi L, Yang Y, Zhou X, Rafique W, Ma J (2022) Fast Anomaly Identification Based on Multiaspect Data Streams for Intelligent Intrusion Detection Toward Secure Industry 4.0. *IEEE Trans Ind Inform* 18(9):6503–6511. <https://doi.org/10.1109/TII.2021.3139363>
- Zhou X, Liang W, Kevin I, Wang K, Yang LT (2021) Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations. *IEEE Trans Comput Soc Syst* 8(1):171–178. <https://doi.org/10.1109/TCSS.2020.2987846>
- Chen Y, Gu W, Li K (2022) Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning. *Int J Commun Syst* 1:e5154. <https://doi.org/10.1002/dac.5154>
- Qi L, Hu C, Zhang X, Khosravi MR, Sharma S, Pang S et al (2021) Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment. *IEEE Trans Ind Inform* 17(6):4159–4167
- Zhou X, Yang X, Ma J, Wang KIK (2021) Energy-Efficient Smart Routing Based on Link Correlation Mining for Wireless Edge Computing in IoT. *IEEE Internet Things J* 9(16):14988–14997. <https://doi.org/10.1109/JIOT.2021.3077937>

13. Huang J, Lv B, Wu Y et al (2022) Dynamic Admission Control and Resource Allocation for Mobile Edge Computing Enabled Small Cell Network. *IEEE Trans Veh Technol* 71(2):1964–1973. <https://doi.org/10.1109/TVT.2021.3133696>
14. Zhou X, Xu X, Liang W, Zeng Z, Yan Z (2021) Deep-Learning-Enhanced Multitarget Detection for End-Edge-Cloud Surveillance in Smart IoT. *IEEE Internet Things J* 8(16):12588–12596. <https://doi.org/10.1109/JIOT.2021.3077449>
15. Xu Y, Yan X, Wu Y, Hu Y, Liang W, Zhang J (2021) Hierarchical bidirectional RNN for safety-enhanced 5G heterogeneous networks. *IEEE Trans Netw Sci Eng* 8(4):2946–2957
16. Gu R, Zhang K, Xu Z, Che Y, Fan B, Hou H, et al (2022) Fluid: Dataset abstraction and elastic acceleration for cloud-native deep learning training jobs. 2022 IEEE 38th International Conference on Data Engineering (ICDE), p 2182–2195. <https://doi.org/10.1109/ICDE53745.2022.00209>
17. Xu Y, Zhang C, Zeng Q, Wang G, Ren J, Zhang Y (2021) Blockchain-enabled accountability mechanism against information leakage in vertical industry services. *IEEE Trans Netw Sci Eng* 8(2):1201–1213
18. Zhou X, Li Y, Liang W (2021) CNN-RNN based intelligent recommendation for online medical pre-diagnosis support. *IEEE/ACM Trans Comput Biol Bioinforma*. 18(3):912–921. <https://doi.org/10.1109/TCBB.2020.2994780>
19. Xu Y, Zeng Q, Wang G, Zhang C, Ren J, Zhang Y (2020) An efficient privacy-enhanced attribute-based access control mechanism. *Concurr Comput Pract Experience* 32(5):1–10
20. Xu X, Liu Q, Luo Y, Peng K, Zhang X, Meng S et al (2019) A computation offloading method over big data for IoT-enabled cloud-edge computing. *Futur Gener Comput Syst* 95:522–533
21. Chen X, Jia S, Xiang Y (2020) A review: Knowledge reasoning over knowledge graph. *Expert Syst Appl* 141:112948
22. Chen Z, Wang Y, Zhao B, Cheng J, Zhao X, Duan Z (2020) Knowledge graph completion: A review. *IEEE Access* 8:192435–192456
23. Dai H, Xu Y, Chen G, Dou W, Tian C, Wu X et al (2022) ROSE: Robustly Safe Charging for Wireless Power Transfer. *IEEE Trans Mob Comput* 21(6):2180–2197
24. Chen Y, Zhao F, Chen X, Wu Y (2022) Efficient Multi-Vehicle Task Offloading for Mobile Edge Computing in 6G Networks. *IEEE Trans Veh Technol* 71(5):4584–4595. <https://doi.org/10.1109/TVT.2021.3133586>
25. Wang M, Qiu L, Wang X (2021) A survey on knowledge graph embeddings for link prediction. *Symmetry* 13(3):485
26. Li Z, Jin X, Li W, Guan S, Guo J, Shen H, et al (2021) Temporal knowledge graph reasoning based on evolutionary representation learning. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p 408–417. <https://doi.org/10.1145/3404835.3462963>
27. Liu Y, Hildebrandt M, Joblin M, Ringsquandl M, Raissouni R, Tresp V (2021) Neural multi-hop reasoning with logical rules on biomedical knowledge graphs. *Eur Semant Web Conf* 1:375–391
28. Ji S, Pan S, Cambria E, Marttinen P, Yu PS (2022) A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans Neural Netw Learn Syst* 33(2):494–514
29. Zha H, Chen Z, Yan X (2022) Inductive relation prediction by BERT. *Proc AAAI Conf Artif Intell* 36(5):5923–5931
30. Xu Y, Zhang C, Wang G, Qin Z, Zeng Q (2021) A blockchain-enabled deduplicatable data auditing mechanism for network storage services. *IEEE Trans Emerg Top Comput* 9(3):1421–1432
31. Das R, Dhuliawala S, Zaheer M, Vilnis L, Durugkar I, Krishnamurthy A, et al (2017) Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851* 1:1
32. Zhang Z, Wang J, Chen J, Ji S, Wu F (2021) Cone: Cone embeddings for multi-hop reasoning over knowledge graphs. *Adv Neural Inf Process Syst* 34:19172–19183
33. Xu Y, Ren J, Zhang Y, Zhang C, Shen B, Zhang Y (2020) Blockchain empowered arbitrable data auditing scheme for network storage as a service. *IEEE Trans Serv Comput* 13(2):289–300
34. Wang B, Meng Q, Wang Z, Wu D, Che W, Wang S, et al (2022) InterHT: Knowledge Graph Embeddings by Interaction between Head and Tail Entities. *arXiv preprint arXiv:2202.04897* 1:1
35. Chen Y, Zhao F, Lu Y, Chen X (2021) Dynamic task offloading for mobile edge computing with hybrid energy supply. *Tsinghua Sci Technol* 10:1. <https://doi.org/10.26599/TST.2021.9010050>
36. Vu T, Nguyen TD, Nguyen DQ, Phung D, et al (2019) A capsule network-based embedding model for knowledge graph completion and search personalization. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, p 2180–2189
37. Chen Y, Liu Z, Zhang Y, Wu Y, Chen X, Zhao L (2021) Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things. *IEEE Trans Ind Inform* 17(7):4925–4934
38. Qi Z, Zhang Z, Chen J, Chen X, Xiang Y, Zhang N, et al (2021) Unsupervised knowledge graph alignment by probabilistic reasoning and semantic embedding. *arXiv preprint arXiv:2105.05596* 1:1
39. Qu M, Chen J, Xhonneux LP et al (2021) Rnnlogic: Learning logic rules for reasoning on knowledge graphs[J]. *arXiv preprint arXiv:2010.04029*, 2020
40. Li K, Zhao J, Hu J et al (2022) Dynamic energy efficient task offloading and resource allocation for noma-enabled iot in smart buildings and environment. *Build Environ* 1:1–1. <https://doi.org/10.1016/j.buildenv.2022.109513>
41. Qi L, Lin W, Zhang X, Dou W, Xu X, Chen J (2022) A Correlation Graph based Approach for Personalized and Compatible Web APIs Recommendation in Mobile APP Development. *IEEE Trans Knowl Data Eng* 1:1. <https://doi.org/10.1109/TKDE.2022.3168611>
42. Sanglerdsinlapachai N, Plangprasopchok A, Ho TB, Nantajeewarawat E (2021) Improving sentiment analysis on clinical narratives by exploiting UMLS semantic types. *Artif Intell Med* 113:102033
43. Bhowmik R, Melo Gd (2020) Explainable link prediction for emerging entities in knowledge graphs. *Int Semant Web Conf* 1:39–55
44. Dettmers T, Minervini P, Stenetorp P, Riedel S (2018) Convolutional 2d knowledge graph embeddings. *Proc AAAI Conf Artif Intell* 32(1):1
45. Rocktäschel T, Riedel S (2017) End-to-end differentiable proving. *Adv Neural Inf Process Syst* 30:1

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)