Received July 28, 2019, accepted August 13, 2019, date of publication August 26, 2019, date of current version September 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937353

Deep Learning-Based Reasoning With Multi-Ontology for IoT Applications

JIN LIU[1], XIN ZHANG[1], YUNHUI LI[2], JIN WANG[3,4], AND HYE-JIN KIM[5]

[1]College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
[2]CFETS Information Technology (Shanghai) Co., Ltd., Shanghai, China
[3]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China
[4]School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350108, China
[5]Business Administration Research Institute, Sungshin Women's University, Seoul 02844, South Korea

Corresponding author: Jin Wang (jinwang@csust.edu.cn)

This work was supported by the National Natural Science Foundation of China (61872231, 61701297, 61772454, 61811530332, 61811540410).

ABSTRACT In the era of mobile big data, data driven intelligent Internet of Things (IoT) applications are becoming widespread, and knowledge-based reasoning is one of the essential tasks of these applications. While most knowledge-based reasoning work is conducted with knowledge graph, ontology-based reasoning method can inherently achieve higher level intelligence by leveraging both explicit and tacit knowledge in specific domains, and its performance is determined by precise refinement of the inference rules. However, most ontology-based reasoning work concentrates on semantic reasoning in a single ontology, and fail to utilize association of multiple ontologies in various domains to extend reasoning capacity. This is even the case for the IoT applications where knowledge from multiple domains needs to be utilized. To overcome this issue, we propose a deep learning-based method to associate multiple ontology rule bases, thereby discover new inference rules. In our method, we first use a regression tree model to determine the threshold value for parameters in inference rules that constitute the ontology rule base, avoiding the influence of uncertainty factors on knowledge reasoning results. Then, a two-way GRU (Gated Recurrent Unit) neural network with attention mechanism is used to discover semantic relations among the rule bases of ontologies. Therefore, the association of multiple ontology rule bases is realized, and the rule base of knowledge reasoning is expanded by acquiring some unspecified rules. To the best our knowledge, this work is the first one to leverage deep learning in reasoning with multiple ontologies. In order to verify the effectiveness of our method, we apply it in a real traffic safety monitoring application by relating rule bases of a vehicle ontology and a traffic management ontology, and achieve effective knowledge reasoning.

INDEX TERMS Deep learning, ontology based reasoning, IoT, sensor ontology, data mining.

I. INTRODUCTION

With the advent of mobile big data era, applications based on mobile terminals are more closely integrated with artificial intelligence technology. And the application of Internet of Things enables a variety of mobile terminals to achieve real-time interconnection. With simple devices, people can enjoy convenient and intelligent services provided by cloud computing platform, and the IoT applications have become widespread in both industry and daily life, such as vehicle IoT System in Smart City [1], smart home monitoring and service [2], and planning, construction, and management toward sustainable cities [3], etc. Along with this, a large amount of data is generated from a variety of domains, environments, terminals, and sensors. These data have the characteristics of large scale, variety, rapid production, great value but low density. In order to make better use of the big data, it is necessary to mine useful information from large amounts of data through effective data analyzing. There has been a lot of research work devoted to technology mining large amount of unstructured data such as knowledge graph [4]. Knowledge graph can extract, organize and manage knowledge in a large number of data resources. It can provide users with intelligent services that understand the users' needs, such as understanding semantics of query and providing accurate answers. This involves knowledge reasoning in the knowledge graphs,

The associate editor coordinating the review of this article and approving it for publication was Tie Qiu.

which is a key step in deep analyzing and reasoning of data. In essence, knowledge reasoning is based on existing rules to infer unknown ones or identify wrong ones, it generally includes tasks such as connection prediction, entity prediction, relationship prediction, and attribute prediction.

Entity and relationship in knowledge graph can be regarded as instance of concept and concept correlation in ontology respectively. While the knowledge reasoning in knowledge graphs attracts attentions in recent years, counterpart research on ontology-based reasoning is seldom. As ontology describes abstract concepts in specific domains, ontology-based reasoning represents more abstract knowledge. This will undoubtedly provide tremendous help in obtaining general and axiom knowledge.

In recent years, ontology engineering is used for description and organization of domain-specific knowledge [5]. The research work on ontology mainly includes ontology's construction, life cycle, development, etc. The essential tasks include knowledge acquisition, representation and reasoning. Knowledge representation is the basis of knowledge organization and retrieval, including framework, production rules, predicate logic, and semantic network. The framework cannot represent concepts such as the intersection and collection of knowledge; production rules cannot easily express the hierarchical relationship of concepts; predicate logic (generally first-order logic) means that knowledge cannot separate concepts from instances, so it's hard to derive knowledge from the concepts effectively. Compared with above three methods, semantic network contains a human understandable rule system whose knowledge expressed by XML can be used for reasoning; furthermore, RDF and OWL makes it possible to conduct more complex representation and reasoning.

Knowledge reasoning can combine ontology and rules to express explicit and implied knowledge in a domain. Its performance depends largely on whether the semantic relationship in the ontology knowledge base can be accurately analyzed, the implied knowledge is extracted, the inference rules are extracted, and the formal description is made using appropriate ontology rule description language. In addition, the semantic reasoning implemented by combining a single ontology and rules is largely limited by the construction of the ontology itself. If multiple ontology associations are realized, and the relationships between two ontologies are analyzed and new semantic relationships are mined, multiple ontologies' rule bases can be associated and new rules based on multiple ontologies can be inferred. For example, there are many inference rules to manage traffic in the traffic management ontology, while there are also many rules to ensure safe driving in the vehicle ontology. If rule bases of these two ontologies can be correlated and unspecified new rules can be discovered, smarter and safer IoT based traffic management can be achieved.

In summary, most knowledge reasoning work so far in the literature is generally about knowledge graph completion or cleaning tasks, thereby cannot adequately express concepts of intersection and collection in abstract knowledge. In addition, the knowledge reasoning work based on ontology is even less, let alone reasoning with multiple ontologies. Therefore, in this article, we propose a multiple ontology-based reasoning method by making use of deep learning. This method first determines values of a number of parameters in ontology rule base, reducing workload of manual parameters settings, and avoiding impact of uncertain factors such as unreasonable empirical parameters setting for knowledge reasoning. Then an RNN is used to discover the semantic relations between the ontologies, so as to realize the association of rule bases of multiple ontologies, thus discover unknown rules and expand rule base for knowledge reasoning.

The remaining of this article is structured as follows. Section II introduces related work of ontology mapping, rule base construction, and knowledge reasoning. Section III presents a novel method of constructing rule base by deep learning and a novel method of knowledge inference with multiple ontologies. Section IV introduces an IoT case for determining the threshold value of parameters for inference rules by the proposed regression tree-based model and an IoT case of semantic reasoning in the traffic safety management. Finally, in section V, some conclusion is drawn.

II. RELATED WORK

In big data era, human generate 2.2 EB data per day on average, and 90% of the total global data is created in the past few years. Big data comes from the pervasive mobile devices, social media tools, and the IoT applications. The IoT is a confluence of a number of different fields merging together to create the promise of connected smart devices [6]. Together with cloud computing and artificial intelligence technology, typical application fields include smart product management, sustainable urban environment, emergency response [7], smart home automation [8], intelligent Transportation System (ITS) [9] and so on. The essence of these application is how to effectively process big data that is generated in the IoT. To explore the value of big data, it is necessary to effectively analyze them, and a core issue in data analysis is how to dig unknown truth from data. This is called knowledge discovery which integrates resource integration, relationship extraction [10], knowledge discovery and information push [11], and realizes the reconstruction of knowledge value [12]. Currently, a large amount of data exists in the form of unstructured text, semi-structured of webpages and structured data of various IT systems [13]. Knowledge graph technology has emerged in order to process these data efficiently. Knowledge graphs can be used to aggregate information, data and link relationships in big data into knowledge by forming a structured semantic knowledge base, thereby making information resources easily evaluated. The missing parts of the semantic knowledge base can be complemented by knowledge reasoning. For example, knowledge graph stores a large number of triples [14], and known triples (*X, birthPlace, Y*) in DBpedia [15] can largely infer missing triples (*X, nationality, Y*).

For the concepts of knowledge reasoning, academic community has given various similar definitions. Tari [16] defines knowledge reasoning as the acquisition of new knowledge from existing knowledge based on specific rules and constraints. Known knowledge is usually expressed in the form of feature-value pairs [17]. Reasoning method is divided into deductive reasoning, inductive reasoning and default reasoning according to new judgment. Deductive reasoning covers widely used methods such as natural deduction, resolution principle, and performance calculation [18], [19]. Case-based reasoning infers new problems by using or adapting solutions to old problems [20]. Traditional knowledge reasoning methods are mainly based on logical and rule-based reasoning, and then gradually developed as the basic general reasoning methods.

The traditional rule-based reasoning method mainly draws on simple rules or statistical features to make reasoning with knowledge graph. Inside the YAGO knowledge graph, an inference engine, Spass-YAGO, is used to enrich the knowledge graph content [21]. Spass-YAGO abstracts the triples in YAGO to the equivalent rule class. The chain superposition is used to calculate the transitivity of the relationship. The superposition process can be iterated arbitrarily. Through these rules, the YAGO extension is completed. Wang *et al.* [22], [23] proposed a first-order probabilistic language model ProPPR (programming with personalized PageRank) for knowledge reasoning on knowledge graphs. Cohen [24] proposed TensorLog, which uses a differentiable process for reasoning. Jang *et al.* [25] proposed a model-based approach to assess the quality of knowledge graph triples. This method performs data pattern analysis directly in the knowledge graph, based on the assumption that the more frequent patterns are more reliable and the selection rate is higher.

Ontology is a formal description of a model which comprise of shared concepts [26]. As a semantic basis, ontology is widely used in information retrieval [37], artificial intelligence, semantic network, software engineering, natural language processing, and knowledge management. In order to meet needs from both industry and academia, a variety of general ontology library systems (such as WordNet, DBpedia, Cyc, HowNet, Frame Ontology, DublinCore, etc.) and a large number of domain ontology library systems have been developed. However, with the widespread development, two problems aroused. First, ontology is usually developed for a specific domain, such as biology, finance, sensor, news, etc. Second, ontology of same domain may have different models and construction methods. Therefore, if these rule bases of ontologies are combined to construct a larger one to represent knowledge in a greater domain, the conceptual structure will become much more complex, and the inference rules are more difficult to set up, which largely limits the capacity of ontology-based intelligent applications.

How to realize the association between heterogeneous ontology and the use of ontology for knowledge reasoning are the basis of various intelligent applications. Since OWL reasoner evaluation (ORE) was completed in 2012, researchers have proposed various reasoners such as UKSTv [27], TrOWL [28], Chainsaw [29], jcel [30], MOre [31], ELepHant [32], ELK [33], HermiT8 [34], PAGOdA [35] and so on. Among them, ontology mapping is the basis for solving the heterogeneity of ontology, and it has been applied to many fields, such as space context awareness, database integration, and discovery and combination of Web services. Ontology mapping refers to finding a correspondence in a multi-ontology entity [36], and is an effective way to solve knowledge sharing and reuse of heterogeneous ontologies in semantic webs, which solve the exchange of complex information [37]. The method of ontology mapping is divided into the four categories: 1) Statistical-based ontology mapping: a statistical approach is used in the mapping process. Prasad *et al.* [38] proposed a method based on Bayesian, while Doan *et al.* [39] proposed a method based on probability distribution in the mapping process. 2) Rule-based ontology mapping: the way in which the heuristic rules are given by domain experts during the mapping process. The mapping method proposed by Ehrig *et al.* [40] is based on heuristic rules. The method first denotes the heuristic rules by domain experts and calculates the similarity of each pair of entities to obtain the calculated results.3) Ontology mapping based on machine learning: the method of machine learning is used during the process of ontology mapping. Gruber [41] proposed an ontology-based classification method using the decision tree classifier method for multi-source classification of nature conservation areas.4) The ontology mapping method based on the ontology concept feature: This kind of method mainly calculates the similarity from the different aspects of the concept name, the instance of the concept, the attribute of the concept and the structure of the ontology. The multi-strategy system that emerges in an endless stream brings a variety of mapping options. However, when two or more ontologies appear, how to correctly select the appropriate mapping method according to their characteristics is difficult. In response to this problem, a number of strategic combination selection methods have been proposed at this stage. Current methods for weight distribution include: methods based on conflict sets [42], methods based on analytic hierarchy [43], methods based on triangular fuzzy numbers [44], methods based on entropy weight decision making [45].

For the ontology based knowledge reasoning, the construction of ontology rule base is the essential part. The rule base can guide the expert system to derive new facts or conclusions from known facts. The construction of the rule base generally uses the semantic web rule language (SWRL). Research on ontology based knowledge reasoning can be generally divided into symbol-based reasoning and statistical-based reasoning. In artificial intelligence, symbol-based reasoning is generally based on classical logic (first-order predicate logic or propositional logic) or classical logic variation (such as default logic). Some researchers have proposed a series of logical languages for concept descriptions, collectively referred to as description logic. It eventually became the logical basis for the W3C-recommended Web Ontology Language OWL. With the large-scale growth of data in recent years, the description logic inference engine has been challenged in the real world development. In recent years, researchers have begun to consider the parallelism of description logic and RDFS to improve the efficiency and scalability of reasoning, and have achieved many results. Goodman *et al.* [46] used the high-performance computing platform Cray XMT to implement large-scale RDFS ontology reasoning. However, for platforms with limited computing resources, optimization of memory usage has become an inevitable problem. Motik *et al.* [47] work to convert RDFS and the more expressive OWL RL into Datalog programs equivalently, and then use the parallel optimization technology in Datalog to solve the memory usage problem. In [48], the authors attempted to improve the reasoning efficiency of OWL RL by using a hybrid approach of parallel and serial. Kazakov *et al.* [49] proposed a method for implementing OWL EL classification using multi-threading technology and implementing the inference engine ELK. Oren *et al.* [50] was the first to attempt to implement RDF data reasoning using Peer-To-Peer's distributed framework. The experimental results show that using distributed technology can accomplish many large data volume inference tasks that cannot be completed in a single machine environment. The inference system WebPIE [51] proposed by Urbani et al. show that it can complete the inference of tens of billions of RDF triples on a large cluster. Based on this, they also proposed the OWL RL query algorithm based on MapReduce [52]. The reasoning algorithm by using MapReduce to implement OWL EL ontology is proposed in [53]. Experiments show that MapReduce technology can also solve large-scale OWL EL ontology reasoning. The reasoning technique of OWL EL is further extended, so that reasoning can be completed in multiple parallel computing platforms [54].

Statistical-based reasoning generally refers to relational machine learning methods. The purpose of this method is to learn the relationship between instances. There is plenty of work in this area. Nickel *et al.* [55] presented a relational potential feature model called a bilinear model that considers the interaction of potential features to learn about potential entity relationships. Drumond *et al.* [56] applied a pairwise tensor decomposition model to learn the potential relationships in the knowledge graph. The translation model [57] uniformly maps entities and relationships into low-dimensional vector space, and obtains potential triad relationships by examining and comparing entity vector pairs with similar potential features in vector space. The Holographic Embedding (HolE) model [58] uses a combination of circular correlation calculation triads and uses circular convolution to recover representations of entities and relationships from the combined representation. Still other methods predict the existence of a possible edge by observing the characteristics of the edges of the triples. Typical methods are methods based on Inductive Logic (ILP) [59], methods based on Association Rules Mining (ARM) [60] and path ranking [61]. SDType [62] uses the statistical distribution of attributes connected by triple subject or predicate to predict the type of instance. This method can be used in the knowledge graph of any single data source, but it cannot do type inference across data sets. Both Tipalo [63] and LHD [64] use DBpedia data's unique abstract data to extract instance types using specific patterns. Such methods rely on textual data of a specific structure and cannot be extended to other knowledge bases.

III. METHOD

A. DETERMINE THRESHOLD VALUE OF PARAMETERS FOR INFERENCE RULES

As mentioned before, knowledge is represented as rules in ontology, and unknown rules can be achieved through ontology rules reasoning. The inference rules are usually predefined and lack the adaptability to different application scenarios. To solve this issue, variable parameters for each rule in the rule base needs to be used. However, these parameters are traditionally determined manually. In order to realize appropriate and accurate ontology-based reasoning for domain specific applications, we propose to use machine learning method to determine the threshold value of parameters for inference rules. To prevent overfitting objective functions in the IoT big data application scenarios where missing data is common and parallel processing is needed, in the proposed method, adaptive data analysis is used to determine appropriate parameters' value by utilizing a scalable end-to-end tree boosting method (XGBoost) to process a large amount of sensor data collected in real time IoT applications. This method can determine appropriate parameters adaptively and dynamically, and eliminate human interference to better extract the characteristics of the data and display it in the form of a tree.

In our approach, sensor data needs to be labeled first. The corresponding data tag represents the trigger state of the semantic inference rule in a multi-dimensional form. The trigger state is uniformly parameterized according to the results of all the established semantic inference rules. Based on a large number of constructed sensor datasets, a regression tree model as shown in Fig 1 is trained. After determining the parameters of rule inference, we can get a more accurate trigger state of each semantic inference rule. In addition, SPARQL is used to reason and query, and specific semantic related resources can be retrieved by using SPARQL without affecting the existing data model.

To better illustrate our approach, we abstractly represent each specific semantic inference rule as r_i, and define $\boldsymbol{\tau} = \{r_1, r_2, \ldots, r_i\}$ as a collection of all semantic inference rules. The trigger status of each rule includes an initial state, a transition state, and a state transition condition. The state transition condition is a set of parameters that need to be obtained to determine the trigger state of the inference rule. We denote the set of parameters B_j of each semantic inference rule as $\boldsymbol{\kappa} = \{B_1, B_2, \ldots, B_j\}$. In our method, we construct the

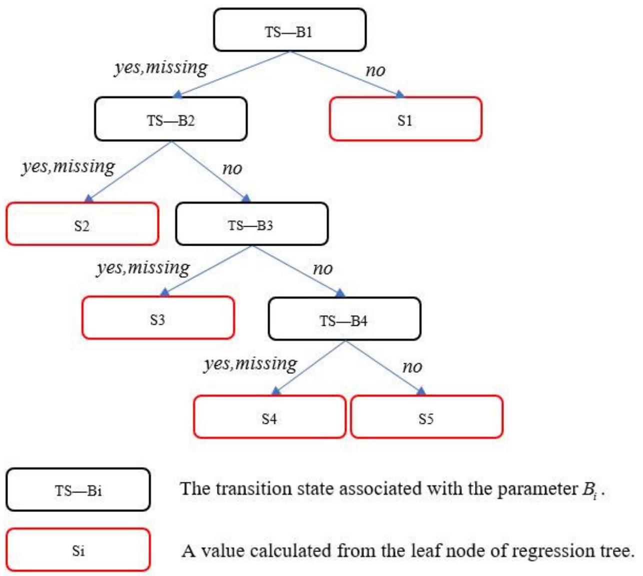

FIGURE 1. Schematic diagram of a regression tree structure for determining rule parameters.

corresponding a regression tree model for each κ to determine each parameter B_j in the set.

We use the same strategy to construct the regression tree for each semantic inference rule. In XGBoost, the definition T represents the number of leaf node of the regression tree, each f_k corresponds to an independent tree structure q and leaf weights ω [65], $loss()$ represents the loss function that measures the difference between the label y_i and the prediction $\hat{y}_i$. $L(\phi)$ represents the objective function, that can be described as the following:

$$L(\phi) = \sum_i loss(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$
$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda \left\|\omega\right\|^2 \tag{1}$$

The Ω is used to penalizes the complexity of the model for avoiding over-fitting. γ and λ are the variable parameters.

In Euclidean space, Eq.(1) cannot be optimized using traditional optimization methods, so f_t is added to minimize the following objective $L^{(t)}$ which represents the objective function at the t-th iteration.

$$L^{(t)} = \sum_{i=1}^{n} loss(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{2}$$

Second-order approximation can be used to quickly optimize the objective in the general [66]. So Eq.(2) can be converted to Eq.(3) as the following:

$$L^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$
$$\text{where } g_i = \partial_{\hat{y}(t-1)} loss(y_i, \hat{y}^{(t-1)})$$
$$\text{and } h_i = \partial^2_{\hat{y}(t-1)} loss(y_i, \hat{y}^{(t-1)}) \tag{3}$$

removing the constant terms, the Eq.(3) can be rewritten by expanding Ω as:

$$\tilde{L}^{(t)} = \sum_{j=1}^{T} [(\sum_{i \in S_j} g_i)\omega_j + \frac{1}{2}(\sum_{i \in S_j} h_i + \lambda)\omega_j^2] + \gamma T \tag{4}$$

The $S_j = \{i | q(x_i) = j\}$ is defined as the instance set of leaf j. For a fixed structure $q(x)$,the optimal weight ω_j^* of leaf j can be computed as following [65]:

$$\omega_j^* = -\frac{\sum_{i \in S_j} g_i}{\sum_{i \in S_j} h_i + \lambda} + \gamma T \tag{5}$$

The Eq.(4) can be rewritten by ω_j^* as the following, which can be used to measure the quality of a tree structure q:

$$\tilde{L}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i \in S_j} g_i)^2}{\sum_{i \in S_j} h_i + \lambda} + \gamma T \tag{6}$$

Normally it is impossible to enumerate all the possible tree structures q. So a greedy algorithm Eq.(7) is used in XGboost to evaluate the split candidates [65].(S_L and S_R are the instance sets of left and right nodes after the split, and $S = S_L \cup S_R$)

$$L_{split} = \frac{1}{2}[\frac{(\sum_{i \in S_L} g_i)^2}{\sum_{i \in S_L} h_i + \lambda} + \frac{(\sum_{i \in S_R} g_i)^2}{\sum_{i \in S_R} h_i + \lambda} - \frac{(\sum_{i \in S} g_i)^2}{\sum_{i \in S_R} h_i + \lambda}] - \gamma \tag{7}$$

Then we can build a regression tree model using XGBoost whose threshold of the split can be used as the reference value for the threshold of inference rules.

Algorithm 1 Determining Threshold Value of Parameters

Input: S, instance of IoT dataset
Input: f, feature dimension
gain < -0
$G \leftarrow \sum_{i \in S} g_i$, $H \leftarrow \sum_{i \in S} h_i$,
For $k = 1$ to m do
  $G_L \leftarrow 0$, $H_L \leftarrow 0$
  For j in *sorted(S, by* x_{jk}) do
    $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
   //Greedy Algorithm for Split Finding
   score $\leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  End
End
Output: Split and default directions with max score.

B. CONDUCT REASONING WITH MULTI-ONTOLOGY

To realize versatile knowledge reasoning, solely determining parameters of rules is not enough, finding new rules is more important in figuring out unknown knowledge. Although algorithms in the literature were proposed to infer new rules, they are all limited to the scope of one single ontology, thereby not applicable in multiple ontologies scenarios. If the rule bases of multiple ontologies are associated according to the similarity relationship between the two concepts in different ontologies, mapping of multiple ontology can be achieved. Then the knowledge reasoning based on multiple ontologies can leverage more rules. These new rules represent the knowledge that hides in multiple domain ontologies, breaking through the limitations of the original standalone ontology, and bring more inference ability to various practical IoT applications.

It is assumed that the ontology is represented by a quad (*concept, relationship, instance set, rule*), where the *relationship* represents the relations between two concepts or concepts and attributes in the ontology. After the multi-ontology is related by the similarity relationship between two concepts, the new knowledge can be expressed as the inferred new relationship between two ontologies and the corresponding concept or attribute of the new relationship. We can represent knowledge in the form of triples (*concept, relationship, concept*).

Then the set of relationships in the ontology can use the above-mentioned knowledge triples to construct the semantic network contained in the ontology. This network structure is essentially a directed graph with path tags. Inspired by this idea, we view knowledge as a sequence and put it into two-way GRU model with attention mechanism (both of them can discover the potential relationship between sequences). In our method, a deep learning method (As shown in Fig 2) is used to construct a new larger semantic network based on the similarity of nodes in these isolated semantic networks, thus knowledge reasoning based on two ontologies can be achieved.

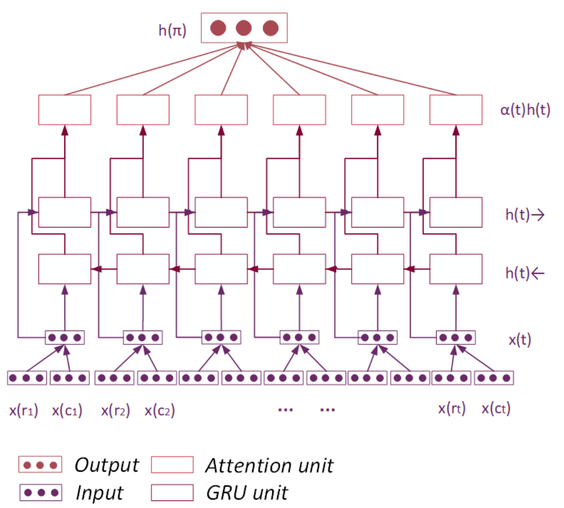

FIGURE 2. **Schematic diagram of the two-way GRU neural network with attention mechanism.**

We denote the concept pairs in these semantic networks as (C_s, C_e), where C_s denotes the starting node in the directed graph and C_e denotes the terminating node in the directed graph. Set S represents all paths between two concepts that exist from C_s to C_e. Where the path is represented as $\pi = \{C_{s(0)}, r_1, C_1, r_2, C_2, \ldots r_k, C_{e(k)}\} \in S$. r represents the specific relationship that exists on the path. The length of the path is k, $len(\pi) = k$. In addition, according to the mapping relationship between two ontologies, C_i' and C_j' $(i, j \in [0, k])$ having similar relationships with C_i and C_j are randomly replaced with C_i and C_j in the corresponding proportions in the set S according to their corresponding degrees of similarity (normalized to the range of [0, 1]). The intersection of the constructed path set and the set S is represented as a set S'.

For the relationship r_t $(t \in [1, k])$ that exists on the path, we use $x_{r_t} \in R^d$ to represent the vector form of the relationship. For the concept C_t $(t \in [1, k])$ that exists on the path, we use $x_{C_t} \in R^m$ to represent the vector form of the concept. Then we use the network structure of the two-way GRU to process the vectors of these representations. The corresponding neural network input vector at the t position is expressed as:

$$x_t = w_{dd} x_{r_t} + w_{md} x_{C_t} \tag{8}$$

where $w_{dd} \in R^{d\times d}$, $w_{md} \in R^{m\times d}$ is the weighting parameter that the neural network needs to determine.

Then the calculation of the two-way GRU neural network at the t position can be expressed as:

$$\vec{h}_t = GRU\left(x_t, \vec{h}_{t-1}\right) \tag{9}$$

$$\overleftarrow{h}_t = GRU\left(x_t, \overleftarrow{h}_{t+1}\right) \tag{10}$$

The calculations in each unit of the GRU (As shown in Fig 3) are defined as follows:

$$z_t = \sigma\left(w_z \cdot [h_{t-1}, x_t]\right) \tag{11}$$

$$r_t = \sigma\left(w_r \cdot [h_{t-1}, x_t]\right) \tag{12}$$

$$\tilde{h}_t = \tanh\left(w \cdot [r_t * h_{t-1}, x_t]\right) \tag{13}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{14}$$

where σ represents the sigmoid activation function, h_t represents the hidden layer of the corresponding GRU, and w_z, w_r

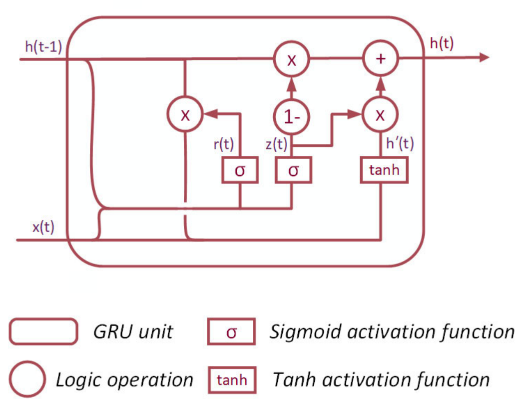

FIGURE 3. **Schematic diagram of each URG unit in the model.**

and w represent the parameters that the neural network needs to train. Through this RNN unit structure using the update mechanism z_t and the reset mechanism r_t, the correlation between the path and the position t closest to the location can be paid more attention. That is, the forgetting mechanism selectively forgets the relationship far away from the current position, thereby avoiding the long-term dependence on the far input, which will result in poor performance of the neural network.

We use all the hidden layers h_t in the two-way GRU structure to construct the neural network output result h_π on the corresponding path π. But in this process, if each hidden layer h_t is given same weight, a more realistic relationship on the path cannot be effectively extracted.

To make different relationships in the path have different effects on final output, our approach introduces an attention mechanism to select the appropriate weight for each hidden layer.

For the hidden layer h_t, the corresponding weight α_t of the application attention mechanism is defined as:

$$\mu_t = \tanh\left(w_t h_t + b_t\right) \tag{15}$$

$$\alpha_t = \frac{\exp\left(\mu_t^T \mu_w\right)}{\sum_t \exp\left(\mu_t^T \mu_w\right)} \tag{16}$$

Among them, w_t and b_t are the weights and offsets in the training parameters for the corresponding neural network, and μ_w are the weight parameters that need to be trained. Therefore, all hidden layer h_t are weighted and summed to calculate the output of the neural network:

$$h_\pi = \sum_t \alpha_t h_t \tag{17}$$

The specific relationship r derived from the neural network output is then defined as:

$$P\left(r|C_s, C_e\right) = \sigma\left(\frac{1}{k}\sum_{i=1}^{k} h_\pi\right), \quad \pi \in S', \; k \in [1, N] \tag{18}$$

We derive the specific relationship of neural network inference based on the first k paths of the results of all the paths (the number is N) in the set S'.

Suppose that the set of all relationships is represented as $R_r = \{r_1, r_2, \cdots, r_n\}$ (the specific relationship that is inferred is $r \in R$). The optimizer function in the neural network training process is expressed as:

$$L\left(\theta, \Delta_R^+, \Delta_R^-\right) = -\frac{1}{M}\sum_{C_s, C_e, r \in \Delta_R^+} \log P\left(r|C_s, C_e\right) + \sum_{\hat{C}_s, \hat{C}_e, \hat{r} \in \Delta_R^-} \log\left(1 - P\left(\hat{r}|\hat{C}_s, \hat{C}_e\right)\right) \tag{19}$$

Δ_R^+, Δ_R^- represent the sets of all positive and negative samples in set R, respectively. θ represents the parameters that the neural network needs to trained in our method.

According to our proposed method, the semantic network for multiple ontology can be constructed, and the specific relationship r, that is, the new knowledge, is inferred for the given concept pair (C_s, C_e).

By constructing a semantic network with recent found knowledge from multiple ontologies, and utilizing the inference rules in each ontology, rule reasoning between two ontologies can be achieved. This will provide richer information and more efficient reasoning for knowledge-based applications. For example, combining a sensor ontology with a domain ontology for shipping industry can discover more reasonable shipping strategy such as a rapid emergency response mechanism triggering by a particular sensor data.

IV. EXPERIMENTAL RESULTS

IoT based intelligent devices and applications have been widely deployed in various fields [67] . Among them, traffic safety which involves vehicle safety, vehicle monitoring, and intelligent traffic control, has always been a concern in daily lives. With the continuous development of driverless cars, traffic safety conditions are increasingly being combined with vehicle driving conditions. By reducing some complicated judgments and operations of the driver, the intelligent decision-making system can more effectively avoid traffic accidents and ensure traffic safety.

Large amount of sensor data from vehicle and urban road monitoring systems has been accumulated. However, with popular data analysis mechanism which comprises of predefined rules, it is difficult to iterate every situation that may occur in real executions. Thus, a more effective way is to extend the rule base with learning method as introduced in section III. This can provide richer information for the IoT system while better understanding for the application scenario.

A. DETERMINE THRESHOLD VALUE OF PARAMETERS FOR INFERENCE RULES

We use the XGBoost based model which was introduced in section III to determine the threshold value of parameters for inference rules to the dataset of traffic safety management.

The definition of the vehicle driving status information is shown in Table 1:

TABLE 1. The vehicle driving status information.

Operating part information	Description
Seat belt status	1-Buckle 0-Unbuttoned
Driver's head state	1-Look straight ahead 0-Other
Parking brake status	1-Pull up 0-Lay down
Door status	1-Open 0-Shut down
Vehicle speed status	Vehicle speed value
Driver safety status	1-safty 0-other

With sensor data, we constract a inference rule with parameters $\{B_1, B_2, \ldots, B_5\}$ to determine whether the driver is

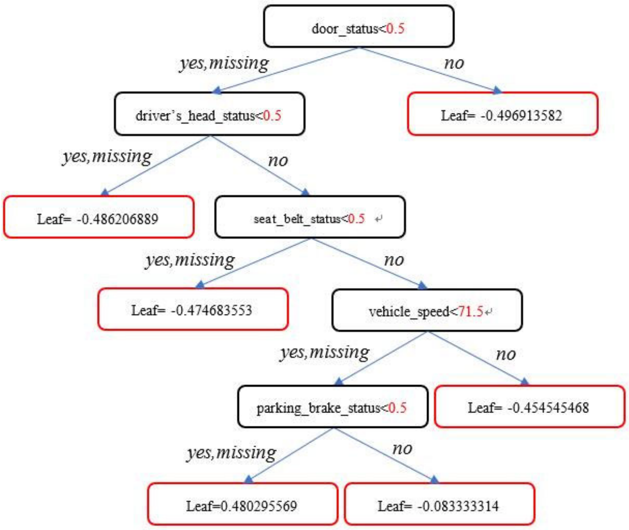

FIGURE 4. A decision tree built by XGBoost on the dataset of vehicle driving status information.

driving safely. Here, the construction of rule is build by semantic web rule language (SWRL). The *?d*, *?z*, *?pe*, *?p*, *?s* respectively represent instances of the concept: *Car Door Value, Car Speed Value, Driver's head state Value, Parking brake Value and Seat belt Value*. The *swrlb:less Than or Equeal* is the built-in atom of SWRL, which represents the instance value is less than or equal to the parameter value.

I. *Car Door(?d)∧Car Door Value(?d,*B_1*)∧Car Speed(?z)∧Car Speed Value(?z, ?x)∧swrlb:less Than Or Equal(?x,*B_2*)∧Driver's head state (?pe)∧Driver's head state Value(?pe,*B_3*)∧Parking brake(?p)∧Parking brake Status(?p,*B_4*)∧Seat belt Status(?s)∧Seat belt Status(?s,*B_5*) →The driver safety status(?ds, "safety")*

In order to train the proposed model, we use a six-tuple to represent data containing the necessary information, *Seat belt status, Driver's head state, Parking brake status, Door status, Vehicle speed status and Driver safety status.*

We run the proposed model and build a decision tree to determine the threshold value of parameters of inference rules, which is shown in Fig 4.

From this decision tree, we can determine the specific parameter values automatically. Then we set the parameters according the decision tree and rewrite rule I as following:

II. *Car Door(?d)∧Car Door Value(?d, 0)∧Car Speed(?z)∧Car Speed Value(?z, ?x)∧swrlb:less Than Or Equal(?x, 71.5)∧Driver's head state (?pe)∧Driver's head state Value(?pe, 1)∧Parking brake(?p)∧Parking brake Status(?p, 0)∧Seat belt Status(?s)∧Seat belt Status(?s, 1)→The driver safety status(?ds, "safety")*

As shown by the experimental result, XGBoost based model is efficient and effective in determining threshold value of parameters for inference rules. To some extent, our method is more effective in predicting value of continuous variables than manual setting, such as the vehicle speed.

TABLE 2. The vehicle operation.

Operation information	Description
Vehicle direction status	Direction offset angle degree
Clutch pedal status	1-Step on 0-Not stepping on
Brake pedal status	1-Step on 0-Not stepping on
Accelerator pedal status	1-Step on 0-Not stepping on
Parking brake status	1-Pull up 0-Lay down
Door status	1-Open 0-Shut down
Seat belt status	1-Buckle 0-Unbuttoned
Gear status	Gear value: 0/1/2/3/4
Engine speed status	Engine speed value
Vehicle speed status	Vehicle speed value
Light switch status (left turn, right turn, low beam, high beam, position light, front fog light, rear fog light, hazard warning flash)	1-Open 0-Shut down
Wiper switch status	1-Open 0-Shut down

TABLE 3. The vehicle information.

Vehicle Information	Description
Vehicle driving state	Start, go straight, stop, turn left, turn right, turn around, change to fast lane, change to slow lane, overtake, pull over.
Vehicle mileage	Current mileage.

TABLE 4. The driver information.

Driver Information	Description
Head state	Look straight ahead, observe the left rear view mirror, observe the right rear view mirror, observe the front of the traffic, look down at the file, look down, etc.
Hand status	Hold the steering wheel with both hands, hold the steering wheel with one hand, leave the steering wheel with both hands, etc.

B. MULTI-ONTOLOGY BASED REASONING BY TWO-WAY GRU WITH ATTENTION MECHANISM

We apply our method to the real-world scene of traffic safety management. By constructing the vehicle ontology and the traffic management ontology, we can effectively infer various traffic safety situations and make decisions for the driver quickly.

The definition of the vehicle operation is shown in Table 2:

The definition of the vehicle information is shown in Table 3:

The definition of the driver information is shown in Table 4:

The definition of the location information of the vehicle is shown in Table 5:

The above information can be obtained in real time through devices on the vehicle, such as sensors, car GPS devices, video camera devices, network devices, and the like. In the vehicle ontology we built, the concept defined by the in-vehicle device is

TABLE 5. The location information of the vehicle.

Location information	Description
Vehicle lane	Lane number.
Vehicle driving or parking area	Cross traffics, school areas, crosswalk areas, bus stop areas, etc.
Driving or parking section	Mountainous areas, culverts, tunnels, curved traffics, steep slopes, expressways, ordinary traffic sections, etc.
Traffic speed limit value	Maximum speed of current traffic segment limit.
The solid line value of the vehicle from the center of the traffic (double yellow line, single yellow line, white solid line)	Unit: cm
Vehicle distance from the right edge of the traffic	Unit: cm

From Device = **From Sensors**∪**From GPS**∪**From Camera**∪**Run Time**.

a) **From Sensors**: The real-time data of the vehicle operating parts and the steering wheel are generally acquired using sensors. **From Sensors** = {steering wheel angle state, clutch pedaling depth, foot brake status, accelerator pedaling depth, hand brake status, door status, seat belt status, gear status, vehicle speed, engine speed, lighting status, vehicle jitter status, steering wheel status,…}.

b) **From GPS**: Vehicle location information collected in real time is generally realized by using a GPS receiving terminal.

c) **From Camera**: The driver's head motion information and hand motion information collected in real time are generally realized by using a multimedia camera device.

d) **Run Time**: The continuous working time of the vehicle equipment in a certain state.

The concept of the vehicle ontology is specifically defined as Table 6:

Some rules for vehicle ontology based on SWRL are:

A. *Car Door(?d)∧Car Door Value(?d, False)∧Car Speed(?z)∧Car Speed Value(?z, ?x)∧swrlb:less Than Or Equal(?x,5)∧Head status (?pe)∧Head status Value(?pe, "looking straight ahead")∧Hand status (?pa)∧Hand status Value(?pa, "holding the steering wheel with both hands")→The vehicle status value (?vs, "slow down parking")∧The driver status value (?ds, "normal driving")*

Rule Description: *Car Door*, *Car Speed*, *Head status*, and *Hand status* are vehicle ontology concepts: Door status, Vehicle speed, Driver's head status, and Driver's hand status;

?d, *?z*, *?pe*, and *?pa* represent instances of *Car Door*, *Car Speed*, *Head status*, and *Hand status*, respectively; *Car Door Value*, *Car Speed Value*, *Head status Value*, and *Hand status Value* are data attributes; *Swrlb:greater Than Or Equal* is the built-in atom of SWRL; *Car Door Value (?d, False)* indicates that the door is closed, *Car Speed Value (?z,?x)∧swrlb:less Than Or qual(?x,5)* indicates that the vehicle speed is less than 5km/h; The result of this rule is that the vehicle is in a

TABLE 6. Vehicle ontology.

First level	Second level	Third level
From Sensors	Steering wheel corner state: *Direction Angle*	
	Clutch pedaling depth: *Clutch Depth*	
	Foot brake status : *Brake Pedal*	
	Throttle step depth: *Throttle Depth*	
	Hand brake status: *Parking Brake*	
	Door status : *Car Door*	
	Seat belt status: *Safety Belt*	
	Manual gear status: *Gear*	
	Vehicle speed: *Car Speed*	
	Engine speed: *Engine Speed*	
	Light state: *Signal Light*	Left turn status: *Left Turn Signal*
		Right turn status: *Right Turn Signal*
		High light state: *High Beam*
		Near lamp status: *Low Beam*
		Warning light status: *Hazard Light*
	Vehicle jitter state: *Vehicle Jitter*	Left and right jitter state: *LR_Jitter*
		Front and rear jitter state: *FB_Jitter*
	Steering wheel: *Steering Wheel*	
From GPS	GPS data reception time: *GPSTime*	
	Longitude: *GPS X*	
	Latitude: *GPS Y*	
	GPS data quality: *GPSQuality*	
	Vehicle driving lane: *On Lane*	
	Vehicle operating status: *Vehicle Run Status*	
	Vehicle travel distance: *Drived Distance*	
From Camera	Driver's head status: *Head status*	
	Driver's hand status: *Hand status*	
Run Time	GPS data status duration: *Time On GPS*	
	Driver's operation state duration: *Time On Action*	
	Time to drive on a marker line: *Time On Tag Line*	
	Time to stay in an area: *Time On Zone*	

"*deceleration*" state and the driver is in a "*normal driving*" state.

B. *Safety Belt (?s)∧Safety Belt Value(?s, False)∧Gear (?g)∧Gear Value(?g, 5)∧Engine Speed(?e)∧Engine Speed Value(?e,?a)∧swrlb:greater Than Or Equal(?a, 3000)∧LR_Jitter (?l, True)∧Head status (?pe)∧Head status Value(?pe, "bow down")∧Time On Action(?c)∧ Time On Action Value(?c, ?x)∧swrlb:greater Than Or Equal(?x, 10)∧Hand status (?pa)∧Hand status Value(?pa, "one hand holding the steering wheel")∧On Lane(?o)∧On Lane Value(?o, "fast lane")→The vehicle status value (?vs, "High speed")∧The driver status value (?ds, "dangerous driving")*

Rule Description: *Safety Belt*, *Gear*, *Engine Speed*, *LR_Jitter*, *Head status*, *Time On Action*, *Hand status*, *On Lane* are vehicle ontology concepts: Seat belt status, Manual gear status, Engine speed, Left and right jitter status, Driver's head status, Driver's operation state duration, Driver's hand state, Vehicle driving lane;

?*s*, ?*g*, ?*e*, ?*l*, ?*pe*, ?*c*, ?*pa*, ?*o* respectively represent instances of the concept; *Safety Belt Value*, *Gear Value*, *Engine Speed Value*, *Head status Value*, *Time On Action Value*, *Hand status Value*, *On Lane Value* are data attributes; *Swrlb:greater Than Or Equal* is the built-in atom of SWRL; The result of this rule is that the vehicle is in a "*high-speed driving*" state and the driver is in a "*dangerous driving*" state.

In the traffic management ontology we built, the concept of setting is **Traffic = On Tag Line∪On Globe∪Traffic vehicle**. Where: **On Tag Line** represents the traffic marker and **On Globe** represents the zone marker. **Traffic vehicle** indicates the state of the traffic vehicle. Here, **On Tag Line** = {center solid line, traffic edge line, pressure lane boundary line, right edge line, sidewalk edge line, outer traffic edge line}; **On Globe** = {No parking area, school area, crosswalk, bus stop, hospital area, cross traffics, sharp bend area, slope area, bridge area}; **Traffic vehicle**= {straight, stop, turn left, turn right, turn around, change lane, overspeed, low speed, collision, retrograde, etc.}.

The concept of the traffic management ontology is specifically defined as Table 7:

Some rules about traffic management ontology based on SWRL rules are as follows:

C. *Traffic vehicle (?r)∧Traffic vehicle Value(?r, "stop")∧ No Pass Area (?n)∧ No Pass Area Value(?n, True)→The traffic safety status (?rs, "violation")*

Rule Description: *Traffic vehicle* and *No Pass Area* are traffic management ontology concepts: Traffic vehicle, No parking area;

?*r*, ?*n* represent instances of the concept, respectively; *Traffic vehicle Value*, *No Pass Area Value* is a data attribute; The result of this rule is that the traffic safety status is "*violation*".

D. *Traffic vehicle (?r)∧Traffic vehicle Value(?r, "straight")∧High Way (?h)∧High Way Value(?h, True)→The traffic safety status (?rs, " speed limit 120km/h")*

TABLE 7. Traffic management ontology.

First level concept	Second level concept
On Tag Line	Solid line driving: *Middle Line*
	Traffic edge line: *Border Line*
	Lane dividing line: *Lane Line*
	Right edge line of the traffic: *Right Border Line*
	Sidewalk edge line: *Cross Border Line*
	Outer traffic edge line: *Out Border Line*
On Globe	No parking area: *No Pass Area*
	School area: *School*
	Crosswalk: *Pedestrian Crossing*
	Bus stop: *Bus Station*
	Hospital area: *Hospital*
	Cross traffics: *Crossing Traffic*
	Sharp bend area: *Corner*
	Slope area: *Slope*
	Arch bridge area: *Bridge*
	High way: *High Way*
Traffic vehicle	

Rule Description: *Traffic vehicle* and *High Way* are traffic management ontology concepts: Traffic vehicle, High way, ?*r*, ?*h* represent instances of the two concepts respectively; *Traffic vehicle Value* and *High Way Value* are data attributes; The result of this rule is that the traffic safety status is "*speed limit 120km/h*".

These inference rules can only be used to manage vehicles and traffics separately. In order to better realize traffic safety management, it is necessary to correlate vehicle ontology and traffic management ontology, and implement new rule reasoning according to ontology rules.

We construct the deep learning model proposed in the section III to discovery new inference rules. Based on the training of a large number of triples *(concept, relationship, concept)* in the data set, the neural network can be used to predict the *relationship* between the target *concept* pairs. Training curve of the deep neural network model trained in our experiment is shown in Fig 5.

The top-5 candidates concept pairs predicted by the model is specifically showed in Table 8. Concept1 is from vehicle ontology and Concept2 is from traffic management ontology. The probability that a concept pair can form a new inference rule, predicted by our model, is shown in column 3.

The new relationship between the target concept pairs obtained through the neural network model can combine two

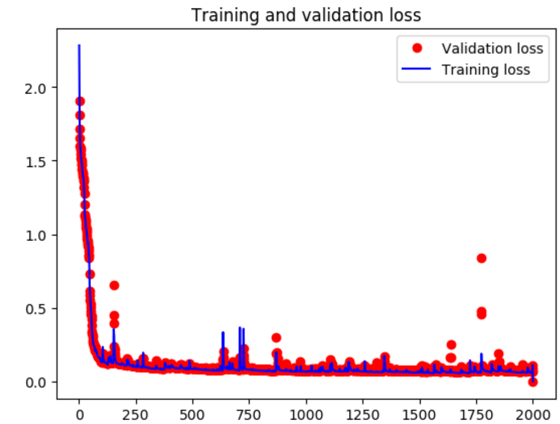

FIGURE 5. Training and validation loss of the proposed neural network.

TABLE 8. The Top-5 predicted probability of concept pairs.

Concept1	Concept2	Relation	Probability
Vehicle status	*Traffic vehicle*	*Equivalence*	0.8743
On Lane	*High Way*	*Inclusion*	0.7472
On Lane	*Slope area*	*Inclusion*	0.5708
On Lane	*Crosswalk*	*Inclusion*	0.5325
Brake Pedal	*Cross Border Line*	*Parataxis*	0.4630

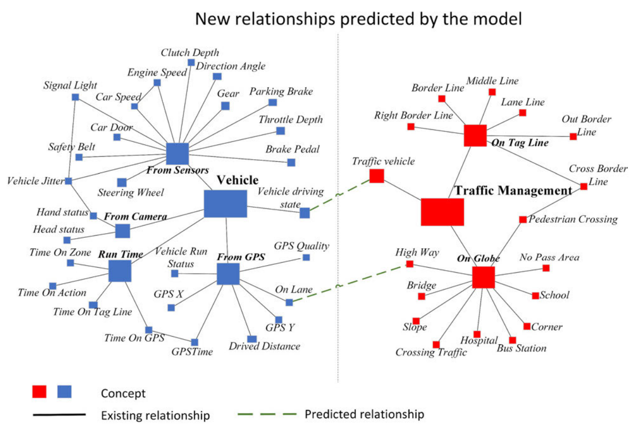

FIGURE 6. New relationships predicted by the proposed model.

relatively independent knowledge representations, as shown in Fig 6. The diagram briefly shows the relationship between the concepts that exist in two relatively independent ontologies. Through the learning of a large number of triples existing in each ontology, the neural network model predicts new relationships between the target concept pairs represented by the green dotted lines in Fig 6. And this is a key step for linking and expanding rule bases.

All the predicted concept pair shown in the Table 8 could be used to generate new rules, without loss of generality, in our experiment, two concept pairs with the highest probability are used to generate rules. The number of the candidate concept pairs can be determined according to the specific application with different data sets.

Based on the prediction of the neural network model, we can infer that two concepts, *vehicle status* from vehicle ontology and *Traffic vehicle* from traffic management ontology, have an equivalence relationship. This combines rules *A.* and *C.* to implement new inference rules *E.*:

i. *The vehicle status value (?vs, "slow down parking") = Traffic vehicle (?r)∧Traffic vehicle Value(?r, "stop")*
ii. *Car Door(?d)∧Car Door Value(?d, False)∧Car Speed(?z)∧Car Speed Value(?z, ?x)∧swrlb:less Than Or Equal(?x,5)∧Head status (?pe)∧Head status Value(?pe, "looking straight ahead")∧Hand status (?pa)∧Hand status Value(?pa, "holding the steering wheel with both hands")→Traffic vehicle (?r)∧Traffic vehicle Value(?r, "stop")∧The driver status value (?ds, "normal driving")*
iii. *Car Door(?d)∧Car Door Value(?d, False)∧Car Speed(?z)∧Car Speed Value(?z, ?x)∧swrlb:less Than Or Equal(?x,5)∧Head status (?pe)∧Head status Value(?pe, "looking straight ahead")∧Hand status (?pa)∧Hand status Value(?pa, "holding the steering wheel with both hands")∧No Pass Area (?n)∧No Pass Area Value(?n, True)→Traffic vehicle (?r)∧Traffic vehicle Value(?r, "stop")∧No Pass Area (?n)∧No Pass Area Value(?n, True)∧The driver status value (?ds, "normal driving")*
iv *Car Door(?d)∧Car Door Value(?d, False)∧Car Speed(?z)∧Car Speed Value(?z, ?x)∧swrlb:less Than Or Equal(?x, 5)∧Head status (?pe)∧Head status Value(?pe, "looking straight ahead")∧Hand status (?pa)∧Hand status Value(?pa, "holding the steering wheel with both hands")∧No Pass Area (?n)∧No Pass Area Value(?n, True)→The traffic safety status (?rs, "violation")∧The driver status value (?ds, "normal driving")*
v *Car Door(?d)∧Car Door Value(?d, False)∧Car Speed(?z)∧Car Speed Value(?z, ?x)∧swrlb:less Than Or Equal(?x,5)∧Head status (?pe)∧Head status Value(?pe, "looking straight ahead")∧Hand status (?pa)∧Hand status Value(?pa, "holding the steering wheel with both hands")∧No Pass Area (?n)∧No Pass Area Value(?n, True)→The traffic safety status (?rs, "violation")*
E *Car Door(?d)∧Car Door Value(?d, False)∧Car Speed(?z)∧Car Speed Value(?z, ?x)∧swrlb:less Than Or Equal(?x,5)∧Head status (?pe)∧Head status Value(?pe, "looking straight ahead")∧Hand status (?pa)∧Hand status Value(?pa, "holding the steering wheel with both hands")∧No Pass Area (?n)∧No Pass Area Value(?n, True)→The traffic safety status (?rs, "violation")*

For rule *B.* and rule *D.*, it is inferred that *On Lane* and *High Way* have an inclusion relationship, whereby new inference rules *F.* can be implemented:

i *On Lane(?o)∧On Lane Value(?o, "fast lane")∈High Way (?h)∧High Way Value(?h, True)* ii *Safety Belt (?s)∧Safety Belt Value(?s, False)∧Gear (?g)∧Gear Value(?g, 5)∧Engine Speed(?e)∧Engine Speed Value(?e, ?a)∧swrlb:greater Than Or Equal(?a, 3000)∧LR_Jitter (?l, True)∧Head status (?pe)∧Head status Value(?pe, "bow down")∧Time On Action(?c)∧ Time On Action Value(?c, ?x)∧swrlb:greater Than Or equal(?x, 10)∧Hand status (?pa)∧Hand status Value(?pa, "one hand holding the steering wheel")∧On Lane(?o)∧On Lane Value(?o, "fast lane")∧Traffic vehicle (?r)∧Traffic vehicle Value(?r, "straight")∧High Way (?h)∧High Way Value(?h, True)→The vehicle status value (?vs, "High speed")∧ The driver status value (?ds, "dangerous driving")∧ The traffic safety status (?rs, "speed limit 120km/h")*

iii *Safety Belt (?s)∧Safety Belt Value(?s, False)∧Gear (?g)∧Gear Value(?g, 5)∧Engine Speed(?e)∧Engine Speed Value(?e, ?a)∧swrlb:greater Than Or Equal(?a, 3000)∧LR_Jitter (?l, True)∧Head status (?pe)∧Head status Value(?pe, "bow down")∧Time On Action(?c)∧ Time On Action Value(?c, ?x)∧swrlb:greater Than Or qual(?x, 10)∧Hand status (?pa)∧Hand status Value(?pa, "one hand holding the steering wheel")∧On Lane(?o)∧On Lane Value(?o, "fast lane")∧Traffic vehicle (?r)∧Traffic vehicle Value(?r, "straight")→The traffic safety status (?rs, "violation")∧ The driver status value (?ds, "dangerous driving")*

F *Safety Belt (?s)∧Safety Belt Value(?s, False)∧Gear (?g)∧Gear Value(?g, 5)∧Engine Speed(?e)∧Engine Speed Value(?e, ?a)∧swrlb:greater Than Or Equal(?a, 3000)∧LR_Jitter (?l, True)∧Head status (?pe)∧Head status Value(?pe, "bow down")∧Time On Action(?c)∧ Time On Action Value(?c, ?x)∧swrlb:greater Than Or equal(?x, 10)∧Hand status (?pa)∧Hand status Value(?pa, "one hand holding the steering wheel")∧On Lane(?o)∧On Lane Value(?o, "fast lane")∧Traffic vehicle (?r)∧Traffic vehicle Value(?r, "straight")→The traffic safety status (?rs, "violation")∧The driver status value (?ds, "dangerous driving")*

It can be seen that a new inference rule combining vehicle ontology and traffic management ontology is realized. This is based on new relationship association inferred from the two concepts that belongs to the vehicle and the traffic management ontology respectively. In this experiment, as we have integrated vehicle and road information, the rule bases are integrated and expanded, more knowledge-based reasoning for road safety is achieved. By combining a large number of sensor data from the vehicle system with the urban road supervision system with our proposed deep learning-based method, the knowledge reasoning in multi-ontology is realized. Moreover, with this method, the IoT system can handle unencountered situation, thus, the road safety situation can be more effectively monitored and evaluated.

V. CONCLUSION

This paper proposed a deep learning-based method to realize automatic new inference rules discovery by association of multiple domain ontologies' rule bases for IoT applications. The experimental results obtained in a case study of traffic safety monitoring prove the effectiveness of the proposed method. Firstly, a scalable end-to-end tree boosting based model is utilized to determine the parameters needed for rule base to avoid the impact of uncertain factors to the results of knowledge-based reasoning. Next, as triples of concept pair with relationship actually form a semantic network, these triples can be combined together as paths in this network, thereby can be vectorized as input of neural network. Then, new inference rules which are based on multiple ontologies were discovered through a model of two-way GRU with attention mechanism. Finally, the association of rule bases in multiple ontologies is established, which expands the rule base for knowledge-based reasoning. Thus, the better understanding of the IoT application scenarios is achieved through our proposed method.

REFERENCES

[1] T. Qiu, X. Liu, K. Li, Q. Hu, A. K. Sangaiah, and N. Chen, "Community-aware data propagation with small world feature for Internet of vehicles," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 86–91, Jan. 2018.

[2] J. Wang, Z. Zhang, B. Li, S. Lee, and R. S. Sherratt, "An enhanced fall detection system for elderly person monitoring using consumer home networks," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 23–29, Apr. 2014.

[3] J. Zhao, X. Zheng, R. Dong, and G. Shao, "The planning, construction, and management toward sustainable cities in China needs the environmental Internet of Things," *Int. J. Sustain. Develop. World Ecol.*, vol. 20, no. 3, pp. 195–198, 2013.

[4] J. Liu, H. Ren, M. Wu, J. Wang, and H.-J. Kim, "Multiple relations extraction among multiple entities in unstructured text," *Soft Comput.*, vol. 22, no. 13, pp. 4295–4305, 2018.

[5] J. Liu, M. Zhou, L. Lin, H.-J. Kim, and J. Wang, "Rank Web documents based on multi-domain ontology," *J. Ambient Intell. Humanized Comput.*, to be published.

[6] Z. Song, A. A. Cárdenas, and R. Masuoka, "Semantic middleware for the Internet of Things," in *Proc. IEEE Internet Things (IoT)*, Nov. 2010, pp. 1–8.

[7] T. Qiu, R. Qiao, and D. Wu, "EABS: An event-aware backpressure scheduling scheme for emergency Internet of Things," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 72–84, Jan. 2018.

[8] W. He, G. Yan, and L. D. Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1587–1595, May 2014.

[9] T. Qiu, X. Wang, C. Chen, M. Atiquzzaman, and L. Liu, "TMED: A spider-Web-like transmission mechanism for emergency data in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8682–8694, May 2018.

[10] D. Zeng, Y. Dai, F. Li, S. Sherratt, and J. Wang, "Adversarial learning for distant supervised relation extraction," *Comput., Mater. Continua*, vol. 55, no. 1, pp. 121–136, 2018.

[11] S. R. Kalidindi, S. R. Niezgoda, G. Landi, S. Vachhani, and T. Fast, "A novel framework for building materials knowledge systems," *Comput., Mater. Continua*, vol. 17, no. 2, pp. 103–125, 2010.

[12] C. Wu, E. Zapevalova, and Y. Chen, "Time optimization of multiple knowledge transfers in the big data environment," *Comput., Mater. Continua*, vol. 54, no. 3, pp. 269–285, 2018.

[13] L. Xiang, Y. Li, W. Hao, P. Yang, and X. Shen, "Reversible natural language watermarking using synonym substitution and arithmetic coding," *Comput., Mater. Continua*, vol. 55, no. 3, pp. 541–559, 2018.

[14] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs," in *Proc. Semantics*, 2016, pp. 1–4.

[15] W. He, Y. Feng, and D. Zhao, "Improving knowledge base completion by incorporating implicit information," in *Proc. Joint Int. Semantic Technol. Conf.*, Berlin, Germany: Springer-Verlag, 2015, pp. 141–153.
[16] L. Tari, *Knowledge Inference*. New York, NY, USA: Springer-Verlag, 2013, pp. 1074–1078.
[17] M. M. Richter and R. O. Weber, *Case-Based Reasoning*. New York, NY, USA: Springer-Verlag, 2016.
[18] A. J. A. Robinson and A. Voronkov, *Handbook of Automated Reasoning*. Amsterdam, The Netherlands: Elsevier, 2001.
[19] F. van Harmelen, V. Lifschitz, and B. Porter, *Handbook of Knowledge Representation*. Amsterdam, The Netherlands: Elsevier, 2008.
[20] C. K. Riesbeck and R. C. Schank, *Inside Case-Based Reasoning*. London, U.K.: Psychology Press, 2013.
[21] M. Suda, C. Weidenbach, and P. Wischnewski, "On the saturation of YAGO," in *Proc. 5th Int. Joint Conf. Automat. Reasoning*. Berlin, Germany: Springer-Verlag, 2010, pp. 441–456.
[22] W. Y. Wang, K. Mazaitis, and W. W. Cohen, "Programming with personalized pagerank: A locally groundable first-order probabilistic logic," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 2129–2138.
[23] W. Y. Wang, K. Mazaitis, N. Lao, T. Mitchell, and W. W. Cohen, "Efficient inference and learning in a large knowledge base: Reasoning with extracted information using a locally groundable first-order probabilistic logic," *Mach. Learn.*, vol. 100, no. 1, pp. 101–126, 2015.
[24] W. W. Cohen, "TensorLog: A differentiable deductive database," 2016, *arXiv:1605.06523*. [Online]. Available: https://arxiv.org/abs/1605.06523
[25] S. Jang, M. Megawati, J. Choi, and M. Yi, "Semi-automatic quality assessment of linked data without requiring ontology," in *Proc. Int. Semantic Web Conf. (ISWC)*. Berlin, Germany: Springer-Verlag, 2015, pp. 45–55.
[26] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data Knowl. Eng.*, vol. 25, no. 1, pp. 161–197, 1998.
[27] M. Wang, J. Wang, L. Guo, and L. Harn, "Inverted XML access control model based on ontology semantic dependency," *Comput., Mater. Continua*, vol. 55, no. 3, pp. 465–482, 2018.
[28] E. Thomas, J. Pan, and Y. Ren, "TrOWL: Tractable OWL 2 reasoning infrastructure," in *Proc. 7th Int. Conf. Semantic Web Res. Appl.* Berlin, Germany: Springer-Verlag, 2010, pp. 431–435.
[29] D. Tsarkov and I. Palmisano, "Chainsaw: A metareasoner for large ontologies," in *Proc. OWL Reasoner Eval. Workshop*, 2012, pp. 19–27.
[30] J. Mendez, "jcel: A modular rule-based reasoner," in *Proc. OWL Reasoner Eval. Workshop*, 2012, pp. 130–135.
[31] R. A. Armas, G. B. Cuenca, and I. Horrocks, "MORe: Modular combination of OWL reasoners for ontology classification," in *Proc. Semantic Web-ISWC*, vol. 7649, 2012, pp. 1–16.
[32] B. Sertkaya, "The ELepHant reasoner system description," in *Proc. OWL Reasoner Eval. Workshop*, 2013, pp. 87–93.
[33] Y. Kazakov, M. Krötzsch, and F. Simančík, "The incredible ELK," *J. Automat. Reasoning*, vol. 53, no. 1, pp. 1–61, 2014.
[34] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, "HermiT: An OWL 2 reasoner," *J. Automat. Reasoning*, vol. 53, no. 3, pp. 245–269, 2014.
[35] Y. Zhou, Y. Nenov, B. C. Grau, and I. Horrocks, "Pay-as-you-go OWL query answering using a triple store," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1142–1148.
[36] Y. K. Hooi, M. F. Hassan, and A. M. Shariff, "A survey on ontology mapping techniques," in *Advanced in Computer Science and Its Applications*. Berlin, Germany: Springer, 2014, pp. 829–836.
[37] Z. Ma, F. Zhang, and L. Yan, "Fuzzy semantic Web ontology mapping," in *Fuzzy Knowledge Management for the Semantic Web*. Berlin, Germany: Springer, 2014, pp. 157–180.
[38] S. Prasad, Y. Peng, and T. Finin, "A tool for mapping between two ontologies using explicit information," in *Proc. AAMAS Workshop Ontol. Agent Syst.*, Bologna, Italy, 2002, pp. 1–5.
[39] A. H. Doan, J. Madhavan, and P. Domingos, "Learning to map between ontologies on the semantic Web," in *Proc. 11th Int. Conf. World Wide Web*, 2002, pp. 662–673.
[40] M. Ehrig and Y. Sure, "Ontology mapping—An integrated approach," in *Proc. Eur. Semantic Web Symp.*, vol. 3053, 2004, pp. 76–91.
[41] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
[42] M. A. Helou, M. Palmonari, and M. Jarrar, "Effectiveness of automatic translations for cross-lingual ontology mapping," *J. Artif. Intell. Res.*, vol. 55, no. 1, pp. 165–208, 2016.
[43] L.-Y. Zhang, J.-D. Ren, and X.-W. Li, "OIM-SM: A methodfor ontology integration based on semantic mapping," *J. Intell. Fuzzy Syst.*, vol. 32, no. 3, pp. 1983–1995, 2017.
[44] Z. Wang, R. Bie, and M. Zhou, "Hybrid ontology matching for solving the heterogeneous problem of the IoT," in *Proc. IEEE 11th Int. Conf. Trust. Secur. Privacy Comput. Commun. (TrustCom)*, Liverpool, U.K., Jun. 2012, pp. 1799–1804.
[45] R. Wang, L. Wang, L. Liu, G. Chen, and Q. Wang, "Combination of the improved method for ontology mapping," *Phys. Procedia*, vol. 25, no. 22, pp. 2167–2172, 2012.
[46] E. L. Goodman, E. Jimenez, and D. Mizell, "High-performance computing applied to semantic databases," in *Proc. Extended Semantic Web Conf. Semanic Web, Res. Appl.* New York, NY, USA: Springer-Verlag, 2010, pp. 31–45.
[47] B. Motik, Y. Nenov, R. R. Pi, I. Horrocks, and D. Olteanu, "Parallel materialisation of datalog programs in centralised, main-memory RDF systems," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1–9.
[48] J. Urbani and C. Jacobs, "RDF-SQ: Mixing parallel and sequential computation for top-down OWL RL inference," in *Graph Structures for Knowledge Representation and Reasoning*. Cham, Switzerland: Springer, 2015.
[49] Y. Kazakov and P. Klinov, "Advancing ELK: Not only performance matters," in *Description Logics*. Athens, Greece, 2015, pp. 233–248.
[50] E. Oren, S. Kotoulas, G. Anadiotis, R. Siebes, A. ten Teije, and F. van Harmelen, "MARVIN: Distributed reasoning over large-scale semantic Web data," *J. Web Semantics*, vol. 7, no. 4, pp. 305–316, 2009.
[51] J. Urbani, S. Kotoulas, J. Maassen, F. van Harmelen, and H. Bal, "OWL reasoning with WebPIE: Calculating the closure of 100 billion triples," in *Proc. Semantic Web, Res. Appl.* Berlin, Germany: Springer, 2010, pp. 213–227.
[52] J. Urbani, F. van Harmelen, S. Schlobach, and H. Bal, "QueryPIE: Backward reasoning for OWL horst over very large knowledge bases," in *Proc. Int. Conf. Semantic Web*. New York, NY, USA: Springer-Verlag, 2011, pp. 730–745.
[53] Z. Zhou, G. Qi, C. Lui, P. Hitzler, and R. Mutharaju, "Scale reasoning with fuzzy-EL+ ontologies based on MapReduce," in *Proc. Workshop Weighted Logics Artif. Intell.*, 2013, pp. 87–93.
[54] Z. Zhou, G. Qi, Z. Wu, and J. Shi, "A platform-independent approach for parallel reasoning with OWL EL ontologies using graph representation," in *Proc. IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2015, pp. 80–87.
[55] M. Nickel, V. Tresp, and H. P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Bellevue, WA, USA, Jun. 2011, pp. 809–816.
[56] L. Drumond, S. Rendle, and L. Schmidt-Thieme, "Predicting RDF triples in incomplete knowledge bases with tensor factorization," in *Proc. ACM Symp. Appl. Comput.*, 2012, pp. 326–331.
[57] A. Bordes, J. Weston, and R. Collobert, "Learning structured embeddings of knowledge bases," in *Proc. AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Aug. 2011, pp. 301–306.
[58] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1955–1961.
[59] J. R. Quinlan, "Learning logical definitions from relations," *Mach. Learn.*, vol. 5, no. 3, pp. 239–266, 1990.
[60] L. Galarraga, C. Teflioudi, K. Hose, and F. M. Suchanek, "Fast rule mining in ontological knowledge bases with AMIE+," *VLDB J.*, vol. 24, no. 6, pp. 707–730, 2015.
[61] N. Lao, T. Mitchell, and W. W. Cohen, "Random walk inference and learning in a large scale knowledge base," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Edinburgh, U.K., 2011, pp. 27–31.
[62] H. Paulheim and C. Bizer, "Type inference on noisy RDF data," in *Proc. Int. Semantic Web Conf.* Berlin, Germany: Springer, 2013, pp. 510–525.
[63] A. Gangemi, A. G. Nuzzolese, and V. Presutti, "Automatic typing of DBpedia entities," in *Proc. Int. Conf. Semantic Web*, 2012, pp. 65–81.
[64] T. Kliegr, "Linked hypernyms: Enriching DBpedia with targeted hypernym discovery," *J. Web Semantics*, vol. 31, pp. 59–69, Mar. 2015.
[65] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[66] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.

[67] T. Qiu, B. Li, W. Qu, E. Ahmed, and X. Wang, "TOSG: A topology optimization scheme with global-small-world for industrial heterogeneous Internet of Things," *IEEE Trans. Ind. Informat.*, to be published. doi: 10.1109/TII.2018.2872579.

JIN LIU received the B.S. degree from Lanzhou University, the M.S. degree from the University of Electronic Science and Technology of China, and the Ph.D. degree from Washington State University, USA, in 2007. He is currently an Associate Professor with Shanghai Maritime University. His research interests include web data mining, machine learning, and the Internet of Things. He is also a member of CAAI and CCF.

XIN ZHANG received the B.E. degree from Shanghai Maritime University, in 2017, where he is currently pursuing the M.E. degree. His research interests include data mining and machine learning.

YUNHUI LI received the B.E. and M.E. degrees from Shanghai Maritime University, in 2017 and 2019, respectively. He is currently with CFETS Information Technology (Shanghai) Co., Ltd. His research interests include data mining, the Internet of Things, computer vision, and machine learning.

JIN WANG received the B.S. and M.S. degrees from the Nanjing University of Posts and Telecommunication, China, in 2002 and 2005, respectively, and the Ph.D. degree from Kyung Hee University, South Korea, in 2010. He is currently a Professor with the Changsha University of Science and Technology, and the Fujian University of Technology. His research interests mainly include routing protocol and algorithm design, and performance evaluation for wireless ad hoc and sensor networks. He is also a member of ACM.

HYE-JIN KIM received the B.A. degree in child welfare and the M.E. degree in child education from Woosuk University, South Korea, and the Ph.D. degree in computer science from the University of Bristol, U.K. After working with the Science and Engineering Research Support Society for two years as the Managing Director, she is currently with Sungshin Women's University, South Korea. She is also an Adjunct Professor with the Vision University of Jeonju, South Korea. She published about 20 articles by 2015. Most of her publications are indexed by SSCI, SCI(E), and SCOPUS. She is also a member of ACM and KIIT. Her current research interests include E-learning and U-learning.