

Ontology-Based Semantic Reasoning for Multisource Heterogeneous Industrial Devices Using OPC UA

Jing Bi¹, Senior Member, IEEE, Rina Wu, Student Member, IEEE, Haitao Yuan², Senior Member, IEEE, Ziqi Wang³, Graduate Student Member, IEEE, Jia Zhang⁴, Senior Member, IEEE, and MengChu Zhou⁵, Fellow, IEEE

Abstract—The advent of smart manufacturing in Industry 4.0 signifies the era of connections. As a communication protocol, object linking and embedding for process control unified architecture (OPC UA) can address most semantic heterogeneity issues. However, its semantics are not formally defined at the application layer. To address the information silo problem caused by semantic heterogeneity, an integration framework named querying of ontology mapping-based OPC UA (QOMOU) is proposed. QOMOU extracts information models of OPC UA servers into resource description framework triples and utilizes Web Ontology Language for semantic enrichment and inference. Then, an event class semantic similarity calculation (ECSSC) method is proposed for device-type identification, enabling the classification of semantically heterogeneous OPC UA devices. The effectiveness of ECSSC is validated through queries with the RDF query language (SPARQL) protocol in Apache Jena. Experimental results demonstrate that ECSSC improves the accuracy of device identification by approximately 7% compared to benchmark device identification models. Specifically, compared with graph embedding-based methods, QOMOU's query performance is approximately 13% higher, and its query efficiency is 5% higher on average compared to both structured query and extensible markup languages. Moreover, by employing a keyword-matching algorithm, the query accuracy of the existing heterogeneous data integration scheme is improved by 4% on average. This enhancement can boost the operational efficiency of Internet of Things systems based on the OPC UA architecture.

Index Terms—Integration framework, object linking and embedding for process control unified architecture (OPC UA), ontology, semantic heterogeneity, semantic similarity, syntactic interoperability.

Received 8 February 2025; revised 14 March 2025; accepted 29 March 2025. Date of publication 9 April 2025; date of current version 27 June 2025. This work was supported in part by the Beijing Natural Science Foundation under Grant L233005 and Grant 4232049; in part by the National Natural Science Foundation of China under Grant 62173013 and Grant 62473014; in part by the Beihang World TOP University Cooperation Program; and in part by the Sichuan Key-Area Research and Development Program under Grant 2024YFHZ0011. (Corresponding author: Haitao Yuan.)

Jing Bi, Rina Wu, and Ziqi Wang are with the College of Computer Science, Beijing University of Technology, Beijing 100124, China (e-mail: bijing@bjut.edu.cn).

Haitao Yuan is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: yuan@buaa.edu.cn).

Jia Zhang is with the Department of Computer Science, Lyle School of Engineering, Southern Methodist University, Dallas, TX 75205 USA (e-mail: jiazhang@smu.edu).

MengChu Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Digital Object Identifier 10.1109/IJOT.2025.3556934

I. INTRODUCTION

WITH the rise of Industry 4.0, communication between devices has become increasingly important. To achieve the Internet of Things, object linking and embedding for process control unified architecture (OPC UA) emerges as a unified architecture for communication in the Industry 4.0 open platform [1]. OPC UA is a unified communication protocol that facilitates communications between devices and systems from different vendors without knowing underlying implementation details. Fig. 1 illustrates the overview of vertical and horizontal communications. Vertical communication may involve data transmission from lower level devices to upper level management systems, while horizontal communication refers to interactions between devices or systems at the same level. OPC UA standardizes communications from the field to a unified information level, adding meta-data to each data object to describe its type, structure, and characteristics [2].

However, interoperability issues may still arise between OPC UA products from different vendors, presenting challenges when integrating devices from various manufacturers. In other words, OPC UA meets the requirements of syntactic interoperability at the information layer. However, semantic interoperability at the information layer remains undetermined. Distributed data management and interoperability rely on specified ontologies [4] between two or more machines at this point. These ontologies can automatically and accurately interpret the meaning of exchanged data and apply it to valuable objectives. For semantic interoperability, ontologies must consider the metadata exchanged between different systems and environments. Raising the level of semantic interoperability can better achieve communications [5] between the control level and the enterprise one. In these communications, providing convenient human-machine interaction interfaces enables even nontechnical managers to clearly understand the operational status of factories or processes through the interactive interfaces. This helps improve production efficiency [6], reduce equipment integration and management costs [7], and enhance management decision-making.

Semantic interoperability is the most significant obstacle faced by widespread applications of the Internet of Things and automation systems. Industry organizations are trying to implement semantic data models that cover a wide range of industries and systems [8], e.g., the Object Management

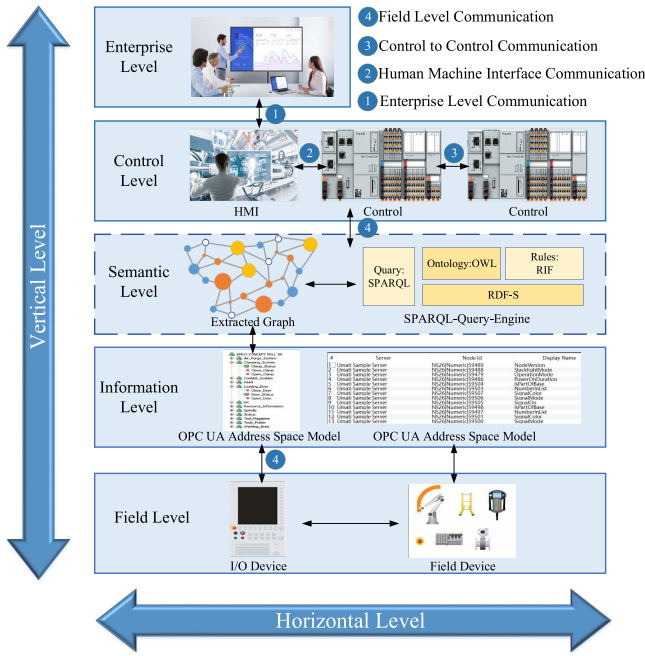


Fig. 1. Overview of vertical and horizontal communications.

Group, Global Standards 1, and Schema.org. However, they are based on a series of industry segments and are not formally defined at the semantic level. In this case, the solution to semantic interoperability is to increase the meaning of data. Data from smart devices is stored and transmitted in multiple formats, with inconsistent and nonstandardized naming conventions and limited descriptions to understand their meanings. There is a considerable amount of literature research on solving interoperability issues at the semantic level. In the early days, manual construction of OPC UA information models is used [9]. Still, this method is cumbersome, error-prone, and costly, and it has obstacles to the widespread application of different devices in industrial production. Based on this, experts have mapped data formats used for exchange in communication protocols to OPC UA information models to achieve compatibility [10]. However, the above-mentioned methods have not effectively solved interoperability at the semantic level efficiently and cannot be widely promoted in industrial production.

Based on the aforementioned analysis, this work proposes an ontology mapping method named querying of ontology mapping-based OPC UA (QOMOU) to tackle the semantic interoperability problems in industrial production. The main contributions are summarized as follows.

- 1) This work extracts the OPC UA information model and converts it into a graph structure. Then, the meaning of data is supplemented with Web Ontology Language (OWL), providing clear structures and significant interpretation of data object meanings.
- 2) This work designs an event class semantic similarity calculation (ECSSC) method to classify OPC UA devices. It calculates the similarity of interpreted concepts among data objects to confirm whether they belong to the same category. Then, event ontology mapping is performed

to support the integration of heterogeneous devices. Finally, ECSSC is compared with benchmark device identification models to verify its superiority.

- 3) This work validates QOMOU's data management and reasoning capabilities by inputting specific RDF query language (SPARQL) queries into the Apache Jena engine's automatic reasoning mechanism. By comparing the query times of datasets with different semantic encapsulation techniques, the query efficiency of QOMOU is validated. Experimental results demonstrate that QOMOU achieves more accurate integration than other graph embedding-based methods. A keyword comparison algorithm is utilized to validate QOMOU's query accuracy. Experimental results demonstrate that QOMOU's query accuracy outperforms its typical peers.

The remainder of this work is organized as follows. Section II discusses related studies on semantic similarity calculation and OPC UA semantic interoperability. Section III shows the overall architecture of QOMOU. Section IV shows the construction of semantic similarity calculation models. Section V conducts experiments to validate QOMOU. Section VI concludes this work.

II. RELATED WORK

A. Semantic Interoperability for OPC UA

Current studies aim to enrich the meaning of data to improve semantic interoperability. For example, Peifeng et al. [17] proposed a semantic model that formalizes variables in events and utilizes SPARQL for querying and accessing data. However, this approach does not provide a clear solution for dealing with heterogeneous data, and it does not validate the accuracy of the query. Mahmoud et al. [18] converted the OPC UA information model into Web Ontology Language description logic, enabling automated reasoning in that language. Wang et al. [19] implemented the transformation and validation of the rules that define shape constraints in the industry foundation class schema based on SPARQL, achieving ontology enhancement. However, this method does not consider the impact of complex network environments that impact the query accuracy of trained models. Wang et al. [20] used a data-driven approach to automatically infer semantics of different devices, providing an ideal solution for issues brought by such heterogeneity. However, this method lacks high-quality OWL datasets and does not consider the long query time of models. To address the aforementioned issues, this work aims to convert the OPC UA information model to OWL ontology, thus improving the model's query accuracy.

B. Ontology Mapping Semantic Similarity Calculation

As an emerging text-processing technology, ontology and the semantic Web provide structured and explicit knowledge representation in the conceptual form. Kakad and Dhage [11] defined event ontology mapping, upon which they propose a semantic similarity calculation model for event ontology mapping. It enables richer semantic mappings between event-based information. However, the semantic computation is overly complex, resulting in excessively long mapping times.

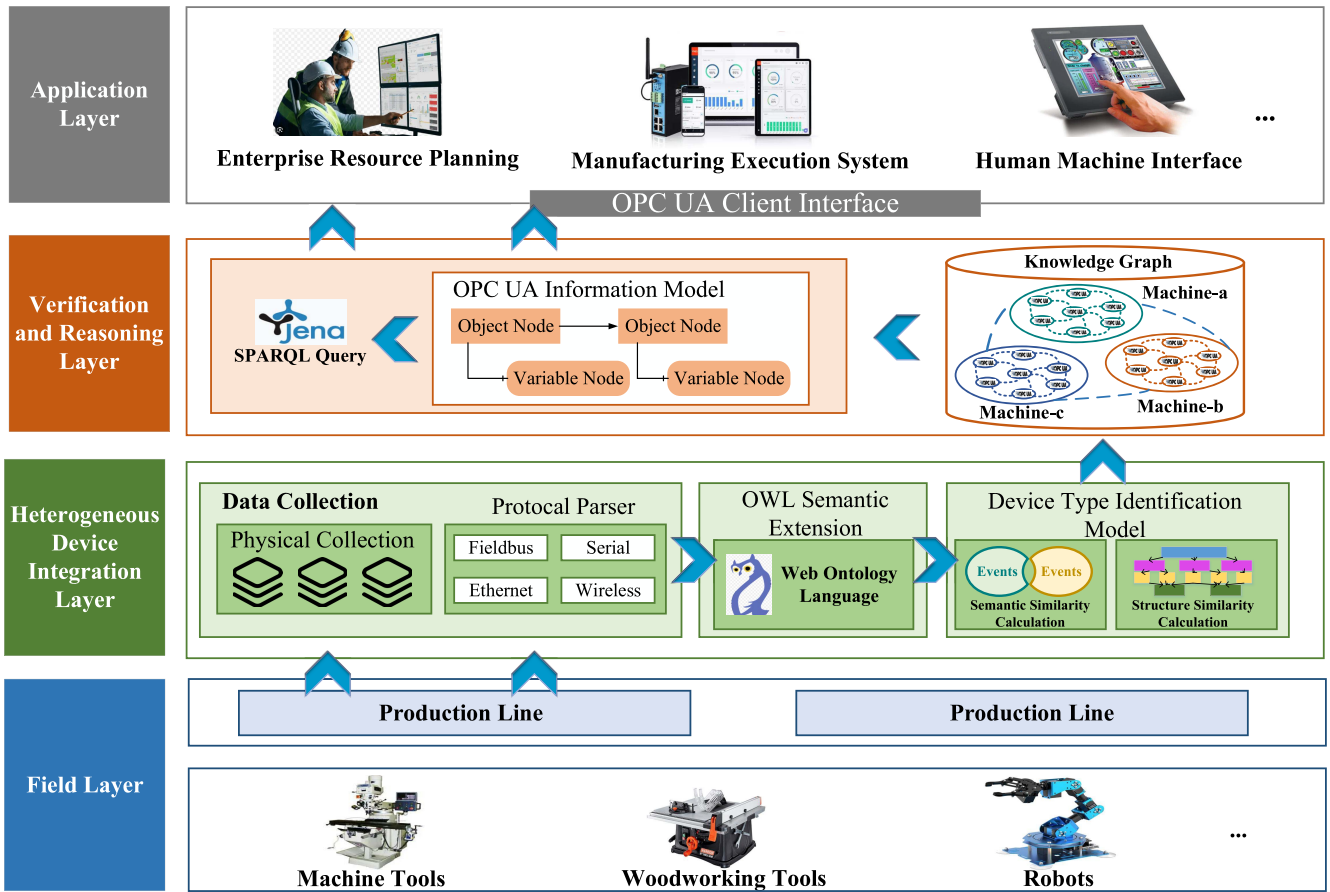


Fig. 2. Conceptual model validation of the integration of OPC UA with ontology, integrating semantic Web technologies with OPC UA for sensor discovery and implementing a semantic access layer to harness the potential of ontologies without disrupting existing OPC UA standards.

Hao et al. [12] aligned ontologies by introducing ontology input and structures. However, this work does not consider the issue of accuracy decline caused by imbalanced sample distribution. De et al. [13] contributed to source descriptions at the dataset, feature, attribute levels, and the integration of source ontology with the specific dataset. However, this work only considers similarity calculations between words and does not incorporate the relational structure within the ontology. Unlike the aforementioned studies, this work combines semantic similarity calculations of event class nodes with node relationships to balance the text diversity, thereby addressing the identification of newly accessed devices.

C. Querying and Answering Systems

The querying and answering system is widely applied in knowledge graphs, providing a method for delivering formalized prior knowledge. It is a machine-interpretable model of implicit knowledge that cannot be obtained through analysis. Gutiérrez et al. [14] utilized ontology-based data access to access heterogeneous manufacturing data from knowledge graphs, representing concepts and attributes related to surface mount manufacturing technologies. They execute SPARQL queries to answer analytical queries about production and faults. However, this approach only applies to relational databases accessed via structured query language

(SQL) queries. Steindl et al. [15] proposed a novel ontology-based approach to access OPC UA data through conventional SPARQL endpoints and custom property functions (CPFs). These CPFs extend the SPARQL query evaluator of the Apache Jena Framework with customized codes executed when CPFs are called in a query. However, due to triple store overload, OPC UA data is stored in a separate database and can only be accessed on demand. To address the overload issue, Mathias et al. [16] suggested that read-only queries should directly access the database, as accessing it through the OPC UA server may increase the computational overhead on the programmable logic controller (PLC). Unlike the above studies, this work adopts a querying and answering system to apply the integrated graph-structured OPC UA device information model.

III. CONCEPTUAL FRAMEWORK

To address the problem of semantic heterogeneity in device integration in current production sites, this section proposes a heterogeneous device integration and application architecture with OPC UA based on graph structures. The proposed architecture is shown in Fig. 2. It consists of four main layers: 1) the field layer; 2) the heterogeneous device integration layer; 3) the verification and reasoning layer; and 4) the application layer.

A. Field Layer

The field layer consists of various devices in the production space. In this architecture, the OPC UA server information uses the simulated OPC UA sample server proposed by UMAIT [25], including information models for machine tools, woodworking tools, robotic arms, etc. Field devices communicate through a unified protocol to ensure effective execution of the production line. When new heterogeneous devices are introduced, device classification is required to achieve integration for these heterogeneous devices.

B. Heterogeneous Device Integration Layer

This layer primarily processes data collected from devices. This heterogeneous information is scattered and unstructured, requiring device-type identification. However, identifying types only based on names is challenging due to the dispersed nature of the information. Therefore, the data is semantically expanded with the OWL technology [26]. To better align with specific requirements of OPC UA-based industrial environments, a custom ontology is developed based on the existing structured dataset. This ontology defines domain-specific concepts, relationships, and attributes, enhancing semantic representation and interoperability. ECSSC treats each semantically expanded OPC UA device as an event class, performs convolution operations to extract feature values, and identifies the device type. Finally, the data is added to the OPC UA address space, enabling the automatic construction of the OPC UA information model. Several instances of the final OPC UA information model are shown in Fig. 3.

C. Verification and Reasoning Layer

This layer plays a significant role as both a validation layer for the previous level and an interface layer for the next level. It is primarily responsible for verifying the correct integration of heterogeneous OPC UA devices and checking the address space of the OPC UA information model. The automatic reasoning tasks related to the graph structure are more focused on providing interfaces for the application layer. The three reasoning tasks are fact prediction, link prediction, and query answering. The problem format of a graph structure is generally $(s, r, ?)$, e.g., $(PosIndirect, Belong, ?)$. Link prediction involves identifying implicit relationships in a graph structure, i.e., relationships that are not explicitly modeled. Fact prediction involves predicting whether a statement is true, such as $(PosIndirect, Belong, Variable)$. Both link prediction and query answering can utilize fact prediction algorithms. However, due to the overwhelming number of candidate entities in the OPC UA information model, and the high-dimensional matrix and vector computations in the embedding model, the high computational cost of evaluating all candidate entities and prediction algorithms makes the verification process excessively expensive [27]. Therefore, link prediction and query answering become the focus of this layer, and a reasoning mechanism is proposed that can effectively query the correct answer without incurring the cost of evaluating all nodes and relationships.

D. Application Layer

This layer builds on the verification and reasoning layer to provide more advanced services, offering a unified application interface for clients. Services can be improved according to different business needs, such as production execution systems, and companies can develop other applications to assist with production planning. In addition, cases related to handling equipment failures in production from the previous layer can also be applied at this level. By combining real-time collected disruption information with existing data, production plans can be adjusted appropriately to implement strategy formulation. Interruption information is collected by acquiring equipment operation logs and analyzing downtime, alerts, or abnormal event records to identify production interruptions. Additionally, by combining multiple data sources (such as sensors, logs, and manual feedback) for cross-validation, misjudgments caused by anomalies in a single data source can be avoided. For example, a production interruption is confirmed only when an abnormal sensor signal aligns with equipment log records.

IV. DEVICE-TYPE IDENTIFICATION MODEL

To achieve semantic heterogeneity in OPC UA device integration, an ECSCC-based device-type recognition model is proposed. Inspired by Algergawy et al. [24] on calculating semantic similarity for various events, this approach combines convolutional neural networks (CNNs) with calculating event class similarity and structural similarity. After classifying OPC UA device types, it completes the OPC UA information model to enable automatic construction.

Different device components and attributes have significant differences, but components and attributes of the same type of device share common characteristics. The OPC UA foundation provides reference models for different types of devices. An existing device-related knowledge base and address space are built in the basement. This involves categorizing nodes in the information model of different machines into six types: 1) Variable; 2) VariableType; 3) Object; 4) ObjectType; 5) DataType; and 6) Method. Each OPC UA information node is treated as an event class when a new device is connected. By leveraging the existing knowledge base of devices, the type of the newly connected device is identified, enabling the automatic construction of the information model graph structure.

A. Event Class Semantic Similarity Calculation

The first step in implementing CNN is to construct a text embedding matrix. A given text sequence of length n can be represented as an embedding matrix $R = [r_T^1, r_T^2, \dots, r_T^n]^T$. $r_i \in \mathbb{R}^{1 \times d}$ is the character embedding for the character i , with an embedding dimension of d . In addition, if the text length exceeds n , it is truncated; otherwise, it is padded with zeros. Then, the embedding matrix R is subjected to a convolution operation, and local information features are extracted with convolution kernels. The feature a_1 can be obtained as

$$a_i = f(k \cdot r_{i:i+h-1} + b) \quad (1)$$

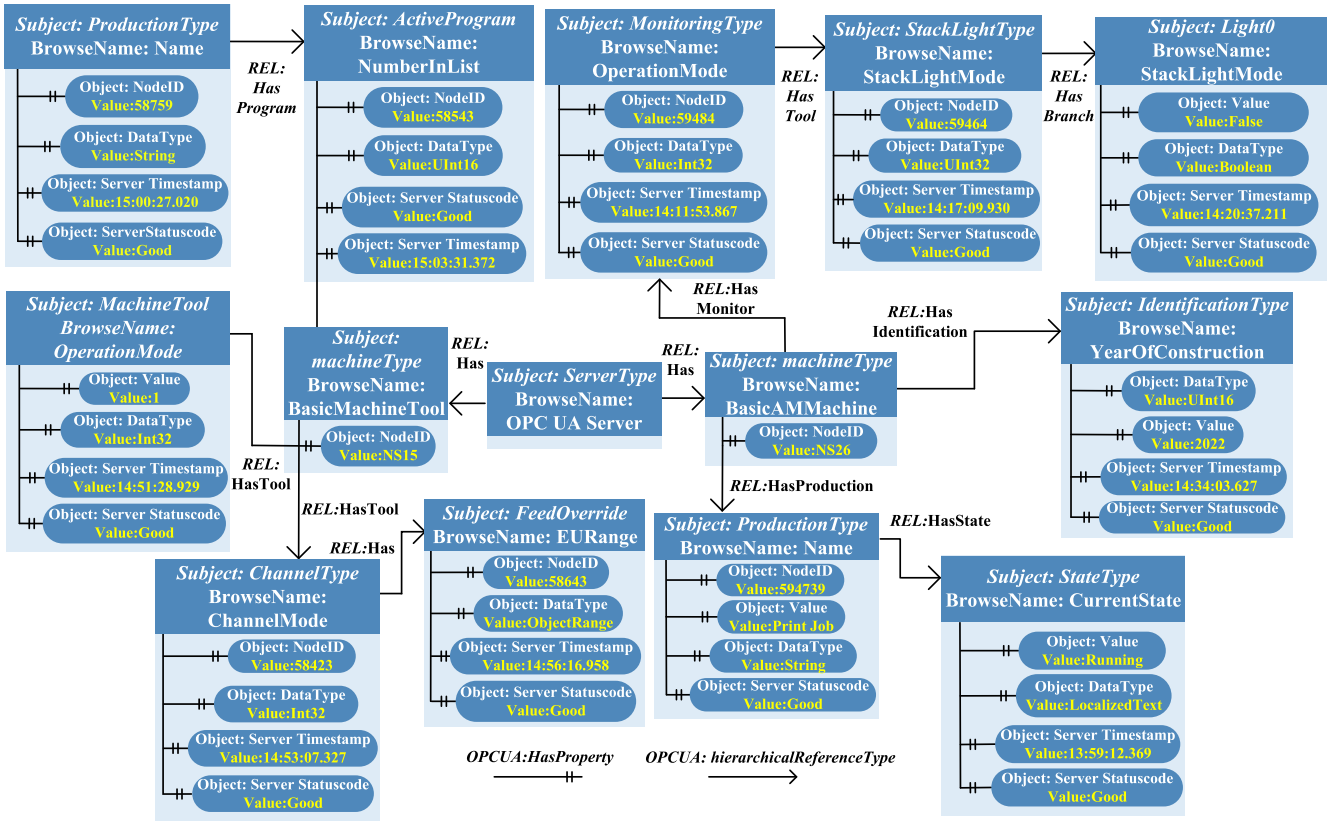


Fig. 3. Example instance nodes of OPC UA servers.

where $f(\cdot)$ denotes the activation function. $c_{i:i+h-1}$ represents the embedding submatrix extracted from the row i to the row $i+h-1$ of C , while retaining all columns. h denotes the window size, $k \in \mathbb{R}^{h \times d}$ is a filter that extracts different feature maps, and $b \in \mathbb{R}$ is a bias term. After the matrix is scanned by the convolution kernel, a feature map $a = [a_1, a_2, \dots, a_{n-h+1}]$ is generated. To simplify the feature description, representative features need to be further extracted. The 1-MAX pooling strategy is used to select the maximum value from the feature map \mathbf{a} obtained by the convolution layer, i.e.,

$$a' = \max(a) \quad (2)$$

where a' represents the maximum value of feature map a . The features obtained in the pooling layer are concatenated to form a feature vector. This feature vector contains different types of information for calculating semantic similarity.

Finally, the feature vector is fed into a fully connected layer to complete device classification. The process of calculating semantic similarity is detailed as follows. Event class is an abstract concept that can be defined as a six-tuple consisting of action a , object o , the time of event t , the location of event p , and the state of event s . It also can represent a group of events with common characteristics, e.g., $C_1 = \{e_{11}, e_{12}, \dots, e_{1n}\}$ and $C_2 = \{e_{21}, e_{22}, \dots, e_{2m}\}$, where n and m denote the number of elements of C_1 and C_2 , respectively. A classic set similarity algorithm is proposed to calculate ontology similarity. The similarity of C_1 and C_2 is denoted as $s_o(C_1, C_2)$, i.e.,

$$s_o(C_1, C_2) = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m s_e(C_{1i}, C_{2j}) \quad (3)$$

where C_{1i} denotes element i in C_1 and C_{2j} denotes element j in C_2 . $s_e(C_1, C_2)$ denotes the element similarity of C_1 and C_2 . It is calculated by multiplying the syntactic and semantic similarities with their respective weights and then adding them together, i.e.,

$$s_e(C_1, C_2) = \sigma_q \cdot s_q(C_1, C_2) + \sigma_t \cdot s_t(C_1, C_2) \quad (4)$$

where σ_q and σ_t are the weights of syntax and semantic similarity, and $\sigma_q + \sigma_t = 1$. Moreover, $s_q(C_1, C_2)$ and $s_t(C_1, C_2)$ denote the syntactic and semantic similarity between C_1 and C_2 , and they are obtained from (5) and (6), respectively







$$s_q(C_1, C_2) = \frac{2 \sum_{i=1}^n \sum_{j=1}^m \phi(C_{1i}, C_{2j})}{l(C_1) + l(C_2)} \quad (5)$$

where $\phi(C_{1i}, C_{2j})$ is the longest common substring between two elements C_{1i} and C_{2j} , and $l(\cdot)$ denotes the length of an event.

In calculating semantic similarity, it is important to understand the concept of sememes. The smallest semantic units in language serve as basic elements composing vocabulary and linguistic meaning. The formula for calculating the similarity of semantics is given as

$$s_t(C_1, C_2) = \frac{|E_{c_1 \leftrightarrow c_2}|}{|c_1| + |c_2|} \cdot \left(\frac{\sum_{i=1}^n \sum_{j=1}^m s_q(C_{1i}, C_{2j})}{|P_{c_1 \leftrightarrow c_2}|} \right) \quad (6)$$

where c_1 and c_2 are two sets of sememes. $|c_1|$ and $|c_2|$ are the numbers of sememes in the sets. $|E_{c_1 \leftrightarrow c_2}|$ is the number of sememes with semantic relationships in two sets. $|P_{c_1 \leftrightarrow c_2}|$ is the number of pairs from semantic sets with semantic

<div> CNC</div> <div> IR</div> <div> SOM</div>			<table><tr><th>types datasets</th><th>CNC</th><th>IR</th><th>SOM</th><th>MAM</th><th>PM</th><th>SCM</th></tr><tr><td>training</td><td>462</td><td>488</td><td>267</td><td>466</td><td>267</td><td>346</td></tr><tr><td>validation</td><td>151</td><td>200</td><td>88</td><td>155</td><td>88</td><td>168</td></tr><tr><td>testing</td><td>151</td><td>200</td><td>88</td><td>155</td><td>88</td><td>168</td></tr><tr><td>total</td><td>764</td><td>888</td><td>445</td><td>776</td><td>445</td><td>682</td></tr></table>	types datasets	CNC	IR	SOM	MAM	PM	SCM	training	462	488	267	466	267	346	validation	151	200	88	155	88	168	testing	151	200	88	155	88	168	total	764	888	445	776	445	682
types datasets	CNC	IR		SOM	MAM	PM	SCM																															
training	462	488		267	466	267	346																															
validation	151	200		88	155	88	168																															
testing	151	200		88	155	88	168																															
total	764	888	445	776	445	682																																
<div> MAM</div> <div> PM</div> <div> SCM</div>																																						
Types of OPC UA devices																																						
Statistics on OPC UA device type identification datasets																																						

(a)

<pre>ns=0;i=17991 WorkOrderVariable Variable ns=0;i=2020 ValueAsText Variable ns=3;i=6006 AccessLevel_CurrentWrite_NotUser Variable ns=0;i=2022 Int32 Variable ns=3;i=6004 Massfolder_Dynamic Object ns=0;i=2021 NumericRange Variable ns=3;i=6005 Boolean Variable ns=0;i=17990 ByteString Variable ns=3;i=6002 Access_Operators Object ns=0;i=2023 SByte Variable ns=3;i=6003 Guid Variable</pre>	<pre>ns=0;i=17991;s=Demo.WorkOrder.WorkOrderVariable WorkOrderVariable Variable ns=0;i=2020;s=Demo.CTT.MultiStateValueDiscreteType.Int64.ValueAsText ValueAsText Variable ns=3;i=6006;s=Demo.CTT.AccessLevel_CurrentWrite_NotUser Variable ns=0;i=2022;s=Demo.CTT.AllProfiles.Arrays.Int32 Int32 Variable ns=3;i=6004;s=Demo.Massfolder_Dynamic Massfolder_Dynamic Object ns=0;i=2021;s=Demo.Static.Arrays.NumericRange NumericRange Variable ns=3;i=6005;s=Demo.Static.Matrix.Boolean Boolean Variable ns=0;i=17990;s=Demo.Static.Scalar.ByteString ByteString Variable ns=3;i=6002;s=Demo.AccessRights.Access_Operators Access_Operators Object ns=0;i=2023;s=Demo.CTT.AllProfiles.Arrays.SByte SByte Variable ns=3;i=6003;s=Demo.CTT.AllProfiles.Scalar.Guid Guid Variable</pre>
Data preview of an OPC UA information model	Data preview of an OPC UA information model after OWL

(b)

Fig. 4. Information on the industrial dataset and OWL-based information model. (a) OPC UA devices and data division for identification. (b) Data of information model and information model after OWL knowledge graph.

relationships. Finally, the semantic similarity between C_1 and C_2 is denoted as $s(C_1, C_2)$, which is given as

$$\sum_{d \in \{o, e, q, t\}} \sigma(x_d) s_d(C_1, C_2) \quad (7)$$

where η_k denotes the degree of impact of each part on the entire system. The final similarity is calculated by multiplying the values of (3)–(6) by their respective weights and then summing them together. To avoid biases caused by human factors, QOMOU adopts a sigmoid function $\sigma(x)$ to calculate the weights, i.e.,

$$\sigma(x) = \frac{1}{1 + e^{-5(x-\alpha)}} \quad (8)$$

where x is the value of syntactic similarity or semantic similarity, -5 is a constant that controls the smoothness of the curve, which helps to avoid the generation of outliers, and α is a parameter that controls the symmetric center position of the curve.

B. Event Class Structure Similarity Calculation

An event class ontology consists of event class nodes and relationships. The calculation of event class similarity ultimately focuses on structure. The structural similarity is defined as $s_{str}(S_1, S_2)$, where S_1 is a set of event classes. Its structure is circular, with S_1 as the center and r as a semantic radius. In S_1 , each node is at a distance p from S_1 and $p \leq r$.

The nodes in S_1 are represented as a triple ($< pre_S, rel, S >$), where pre_S denotes the previous node, rel denotes the relationship between the two nodes, and S represents the event class of the current node.

The process for calculating the structural similarity of event classes is shown in Algorithm 1. A direct neighbor node refers to retrieving all neighboring nodes with a path length of 1. The final semantic similarity is obtained as $S_f = \max\{s(C_1, C_2), s_{str}(S_1, S_2)\}$. The device type is determined after obtaining the maximum similarity between the OPC UA device and the preconstructed information model.

The SoftMax function is used in the fully connected layer to perform classification, and the cross-entropy loss function (CEL) is employed to minimize the gap between the prediction results and the ground-truth classes. The value with the highest computed prediction is chosen as the final prediction result. The CEL function is shown as follows:

$$L = \frac{1}{M} \sum_i L_i = \frac{1}{M} \sum_i \sum_{c=1}^M y_{ic} \log S_f \quad (9)$$

where M denotes the number of device types. y_{ic} represents a delta operator that is 1 when the actual device category i matches category c , and 0 otherwise. S_f is the final similarity of the device. The form of the loss function is a cross-entropy loss, which measures the difference between the model output similarity S_f and delta operator y_{ic} . The model adjusts its parameters to gradually reduce the value of the

Algorithm 1: Calculation of Structure Similarity

Input: S_1, S_2 , similarity threshold value (tv)
Output: $s_{str}(S_1, S_2)$.

- 1 Initialize parameters and $p = 1, r = 5, s_{str}(S_1, S_2) = 0$. Record the maximum similarity between nodes and $ms = 0$. Choose a node from S_2 as pre_A and select a node from S_1 as pre_B . As a temporary variable for storing similarity values and $sim = 0$
- 2 **while** $p \leq r$ **do**
- 3 Get all direct neighbor nodes of pre_A as $nodesA$
- 4 Get all direct neighbor nodes of pre_B as $nodesB$
- 5 **for** $nodeA$ in $nodesA$ **do**
- 6 $ms = 0$
- 7 **for** $nodeB$ in $nodesB$ **do**
- 8 **if** $tv \leq s(nodeA, nodeB)$ and $ms \leq s(nodeA, nodeB)$ **then**
- 9 $sim = s(nodeA, nodeB)/p$
- 10 **if** $nodeA.rel == B.rel$ **then**
- 11 $sim = sim*(1 + (p/r*10))$
- 12 **end**
- 13 $ms = s(nodeA, nodeB)$
- 14 **end**
- 15 **end**
- 16 $s_{str}(CS_1, CS_2) = s_{str}(CS_1, CS_2) + sim$
- 17 **end**
- 18 $p = p + 1$
- 19 **end**

loss function. During the training process, the model updates the input parameters through the back propagation algorithm, the accuracy improvement resulting from the loss reduction is demonstrated in the next section through experimental results comparing QOMOU with other state-of-the-art embedded models.

V. EXPERIMENTAL RESULTS

This section evaluates the performance of ECSSC in the OPC UA device-type identification model and compares it with benchmark models. To validate QOMOU, this section first validates the semantic similarity calculation model and verifies the OPC UA information query model. The validation of the information query model needs to discuss the relationship between devices and their components, as well as the ability to identify abnormal situations in device events and handle these anomalies.

A. Industrial Dataset

To the best of our knowledge, no existing datasets are available for this experiment. Therefore, we construct four datasets for experiments, which are derived from node sets of the officially published OPC UA companion specification.¹ It includes data from 15 typical devices in the industrial manufacturing sector, categorized into six classes, including computer numerical control machine tools (CNC), industrial robots (IRs),

TABLE I
STATISTICS OF DATASETS

Dataset	Companion Spec.	Ent.	R.	Trip.
FES	-	147	43	1063
UADS	DI, PLCOpen	270	39	830
FDSO	DI, FDI, <i>etc.</i>	189	47	793
FDST	DI, CNC, <i>etc.</i>	196	36	1312

sorting machine (SOM), marking machines (MAMs), patching machine (PM), and scribing machine (SCM), each following its respective OPC UA specification. According to ISA88, these information models can also be considered part of the physical models. Thus, it can be transformed into OWL with the source tool Lions.² The resulting graph structure contains extensive process descriptions and technical resource information. These descriptions can provide detailed descriptions of different device nodes, thereby providing numerical values for subsequent event similarity calculations. Then, these words are randomly grouped into word sequences, each representing a sample. Fig. 4(b) shows the data preview of the OPC UA information model and that after being structured into a graph. Finally, the samples are labeled according to the device type, resulting in 4000 samples. Fig. 4(a) shows the training set, the validation one, and the test one are randomly split in a ratio of 6:2:2. *Companion Spec.* represents the associated specifications, *Ent.* indicates the number of unique entities, *R.* denotes the number of relationships, and *Trip.* refers to the total number of triples in the dataset. The four datasets include FAPS Empty Server (FES), Unified Automation Demo Server (UADS), FAPS Demo Server One (FDSO), and FAPS Demo Server Two (FDST). Among them, UADS follows the specifications associated with DI and PLCOpen. FDSO adheres to DI, FDI, ISA95, AutomationML, AutoID, and MachineVision. FDST complies with DI, CNC, Robotic, PackML, IOLink, and Plastics Rubber.

For validating the integration framework, these node sets primarily specify the type system of the information model and rarely define concrete instance examples. Therefore, we adopt extended triple datasets generated by ECSSC. Table I presents the statistical information of the datasets.

B. Experimental Results

To verify the effectiveness of the event class similarity calculation model, the nodes in the information model are categorized into six classes and visually distinguished by different colors. By crawling the nodes and their neighboring ones in the graph, the text of each data record is converted into vector representations to construct a feature matrix. Clustering analysis is then performed with the feature matrix to obtain cluster labels. Then, principal component analysis (PCA) is applied to reduce the dimensionality of the high-dimensional similarity matrix, and the results are projected into a 3-D space for visualization. It is shown in Fig. 5 that the distribution of six classes of nodes is scattered. Thus, the heterogeneity of

¹<https://github.com/OPCFoundation/UA-Nodeset>

²<https://github.com/hsu-aut/lion>

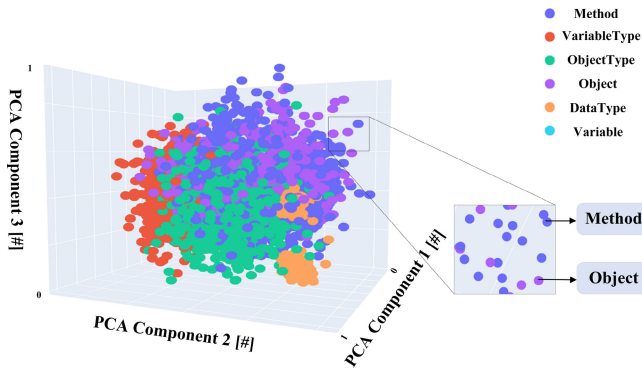


Fig. 5. Clustering status of the information model before similarity calculation.

each device is not resolved, and it is necessary to partition nodes of each device.

The computed similarity values are added to the feature matrix as parameters, and the state after clustering the matrix is shown in Fig. 6. It is shown that the states of nodes in the graph are reasonable and similar concepts tend to form clusters. OPC UA nodes of the same type cluster together. For example, nodes belonging to the variable are marked by the blue color and are closer in feature distance.

After verifying the effectiveness of ECSSC, the performance of its device-type identification model is evaluated by comparing it with four other device-type identification models. Since ECSSC incorporates semantic similarity as an improvement factor in the enhanced CTCNN model, its recognition accuracy is compared with other classic TextRNN models and their variants. To provide a clearer baseline comparison, several benchmark models, including TextRNN and vanilla CNN, are selected for evaluation. TextRNN captures sequential dependencies in text data with recurrent neural networks, while CNN-based methods are leveraged to extract local features through convolutional operations. These models are adopted as fundamental baselines to highlight the performance improvement achieved by ECSSC.

TextRNN [30]: CNN is replaced by bidirectional long short-term memory (BiLSTM), which extracts features from the input character embeddings with BiLSTM, and the SoftMax function is used to classify OPC UA device types.

TextRNN_Att [31]: An attention mechanism is additionally added based on TextRNN.

DPCNN [32]: It combines CNN with a deep pyramid structure, enabling it to capture long-distance dependencies in the device-type text.

CTCNN [33]: It creates an alphabet through a corpus and randomly initializes character embeddings to form an embedding matrix. The matrix undergoes convolution operations to obtain feature maps. Finally, the features are concatenated to form a feature vector. This feature vector is fed into a fully connected layer for device classification.

Three evaluation metrics are used to evaluate the performance of the device-type recognition model, including Precision (P), Recall (R), and F1-score ($F1$), which are

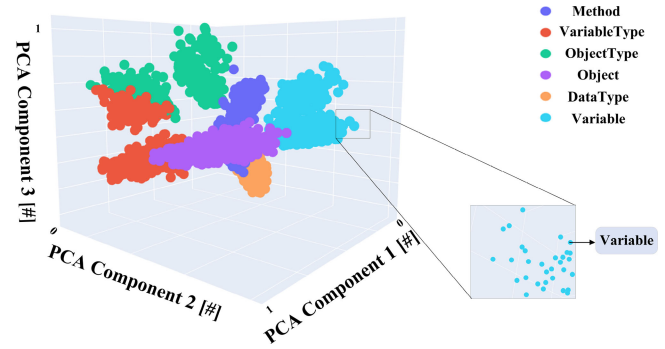


Fig. 6. Clustering status of the information model after similarity calculation.

TABLE II
NOTATIONS OF TP, FP, AND FN

Actual result	Prediction result	
	0	1
0	True Negatives (TN)	False Positive (FP)
1	False Negative (FN)	True Positive (TP)

TABLE III
PERFORMANCE COMPARISON OF DEVICE-TYPE IDENTIFICATION MODELS

Models	TN	FP	P	R	$F1$
TextRNN	1850	450	0.8326	0.8452	0.8322
TextRNN_Att	1920	280	0.9192	0.9162	0.9162
DPCNN	1930	260	0.9237	0.9217	0.9238
CTCNN	1980	120	0.9736	0.9657	0.9821
ECSSC	1995	90	0.9842	0.9913	0.9917

calculated as follows:

$$P = \frac{TP}{FP + TP} \quad (10)$$

$$R = \frac{TP}{FN + TP} \quad (11)$$

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (12)$$

where the meanings of TP, FP, and FN are shown in Table II.

The experimental results are shown in Table III. To provide a more comprehensive evaluation of the classification performance, TN and FP results are given in Table III. With these additional metrics, the analysis is not limited to Precision, Recall, and F1 Score. Still, it offers a deeper understanding of identifying device types correctly while minimizing classification errors. They demonstrate that ECSSC achieves the highest accuracy in device identification. The confusion matrix of ECSSC, as shown in Fig. 7, verifies that the proposed model accurately identifies OPC UA devices.

C. Model Capability Validation

To answer the question “which variables belong to a certain machine?,” in an OWL model without semantic reasoning, a significant issue arises where the components in the third level in Fig. 3 cannot be categorized into the first-level equipment types. This means that the relationships between superclasses and subclasses [34] must be manually annotated, resulting in

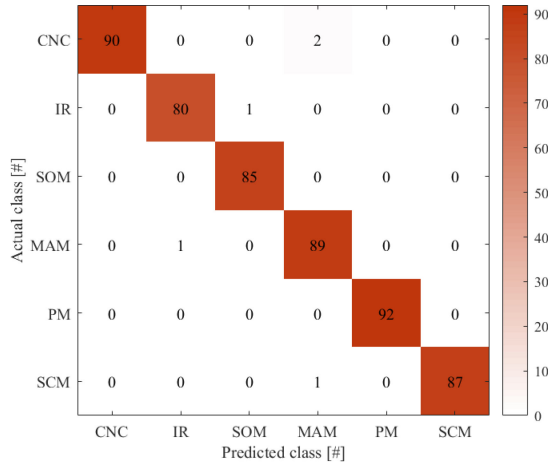


Fig. 7. Confusion matrix of ECSSC.

```

SELECT ?variable ?nodeID ?variableName ?variableType ?machineID
?machineName
#Querying information from OWL
WHERE {
?urnid owl:sameVariableAs OpcSS:AllMachines.
#Selecting variables from OPC UA machines.
?machine OpcUa:hasComponent* ?variable.
?machine OpcUa:browseName* ?machineName.
?variable OpcUa:belongsTo ?machineID.
?variable OpcUa:browseName ?variableName.
?variable OpcUa:typeDefinition ?variableType.
#Filtering variable types.
FILTER( ?variableType = "IdentificationType" || ?variableType =
"ProductionType" || ?variableType = "MonitoringType").
#Querying from a Specific Time Point
?node OpcUa:histValues ( ?Time ?Value "2024-03-16T08:00:00Z"
"2024-03-28T08:20:00Z").
}
LIMIT 5

```

Listing 1. Query statement-part A.

increased data volume and difficulty in querying. Therefore, class hierarchy inference and reverse reasoning are introduced to ensure query completeness. The query statement is shown in Listing 1, and the OWL reasoning is shown in Listing 2. The part ownership relationship query results are shown in Fig. 8. The results demonstrate that with the OWL inference engine, subclasses within OPC UA nodes can be automatically recognized as superclasses without explicitly specifying the relationships among them.

When the device encounters an abnormal event, the architecture queries the issue and raises an abnormal alarm. The code for querying anomalies is displayed in Listing 3, and the detection results are shown in Fig. 9. The results demonstrate that the architecture can remind the staff to switch the mode from automatic to manual, which helps prevent an abnormal situation during production.

D. Efficiency and Accuracy of Queries

In this section, the experiments on the query efficiency and query accuracy of QOMOU are conducted to validate its effectiveness [35]. SQL datasets with and without format wrapping and Extensible Markup Language (XML)-based

```

@prefix : <http://www.zenodo.org/records/.com#> .
@prefix owl: <http://www.zenodo.org/2002/07/owl#> .
@prefix rdf: <http://www.zenodo.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <XML Schema> .
@prefix rdfs: <http://www.zenodo.org/2000/01/rdf-schema#> .
#Subclass and Superclass Reasoning
[ruleBelongsToMonitoringType: (?p :hasMonitor ?m) (?m :hasTool ?g) (?g
:hasBranch ?q) (?p browseName: 'BasicAMMachine')-> (?q
rdf:belongsTo :p)]
#Inverse Reasoning of Inclusion Relationship
[ruleInverse: (?p :hasVariable ?m) -> (?m :belongsTo ?p)]

```

Listing 2. OWL reasoning.

QUERY RESULTS

Table

Raw Response

Showing 1 to 5 of 5 entries

Search:

Show 50 entries

	variable	nodeID	variableName	variableType	belongsTo	machineName
1	<https://zenodo.org/records/6336935#monitor/59022>	"59022"	"Manufacturer"	"ProductionType"	"NS19"	"BasicWoodWorking"
2	<https://zenodo.org/records/6336935#monitor/58764>	"58764"	"OperationMode"	"MonitoringType"	"NS26"	"BasicAMMachine"
3	<https://zenodo.org/records/6336935#monitor/59489>	"59489"	"NodeVersion"	"StateType"	"NS26"	"BasicAMMachine"

Fig. 8. Query results of the part ownership relationship.

QUERY RESULTS					
Table Raw Response					
Showing 1 to 3 of 3 entries Search: Show 50 entries					
event	eventType	value	timestamp	statusCode	
1	<https://zenodo.org/records/6336935#event/59031>	"Alarm"	"false"	"20:14:26"	"good"
2	<https://zenodo.org/records/6336935#event/59026>	"CurrentMode"	"AUTOMATIC"	"20:14:26"	"good"
3	<https://zenodo.org/records/6336935#event/59029>	"Error"	"false"	"20:14:28"	"good"

Fig. 9. Query results of the part anomalies.

datasets are selected to compare query efficiency. Moreover, 4000 OPC UA server data samples are extracted, with query times recorded for every 300 pieces of data. Fig. 10 compares query times for different semantic Web technologies. It is shown that the query time of QOMOU is smaller than that of SQL and XML semantic integration schemes, validating its effectiveness.

To compare the performance of QOMOU, it is evaluated against graph embedding-based methods. Four different embedding models based on different embedding assumptions (TransE [37], DistMult [38], ComplEx [39], and HolE [40]) are utilized, and hyperparameter optimization with grid search is conducted for each embedding model to identify the optimal training parameters for each dataset. Since these four classic models represent different embedding assumptions and

TABLE IV
COMPARISON OF QUERY-ANSWERING PERFORMANCE BETWEEN QOMOU AND GRAPH EMBEDDING-BASED METHODS

Dataset	Score	Model				
		TransE	DistMult	ComplEx	HolE	QOMOU
FES	MRR	70	76	78	52	80
	@10	79	89	87	64	86
	@3	73	84	83	55	87
	@1	67	68	73	46	80
UADS	MRR	77	73	79	60	80
	@10	84	88	89	65	88
	@3	80	81	87	61	89
	@1	73	63	84	57	87
FDSO	MRR	88	77	80	49	85
	@10	94	92	91	59	89
	@3	90	86	88	51	92
	@1	85	68	72	43	89
FDST	MRR	62	73	74	48	80
	@10	72	85	84	60	87
	@3	65	76	77	53	82
	@1	56	65	70	41	68

TABLE V
QUERY ACCURACY ANALYSIS

Methods	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Float Ratio
Probability factor framework	60	49	58	68	58	57	13.5 ± 4.5
Logical object-oriented interaction	69	62	57	56	61	55	19.5 ± 3.5
QOMOU	71	65	63	60	64	56	4.2 ± 1.8

```

SELECT ?event ?eventType ?value ?timestamp ?statuscode
?machineName
WHERE {
#Selecting events from BasicAMMachine Procedure and Unit
?proc ISA88:isMonitoringInProcessStage ?Process .
?urnid owl:isAssignTo ?Process .
FILTER( ?proc = OpcSS:UnitProcedureWarning).
FILTER( ?unit = OpcSS:BasicAMMachine).
#Selecting timestamps of events
?Process ISA88:hasInput ?starttime.
?stimeDE Alarm59031:hasDescription
OpcSS:StartTimeProcess;
Error59029:hasDescription /
CurrentMode:Value ?starttime.
#Selecting events from OPC UA machines.
?machine OpcUa:hasEvent* ?event .
?event OpcUa:type* ?eventType .
?event OpcUa:hasValue ?value .
?event OpcUa:monitorTime ?timestamp .
?event OpcUa:state ?statuscode .
FILTER(?variableType = "State")||?variableType = "MonitoringType") .
#Extracting from RDF information
?node OpcUa:histValues ( ?Time ?Value ?starttime ?endtime ) .

```

Listing 3. Query statement-part B.

modeling approaches, and have been widely validated with multiple datasets, the performance of QOMOU is chosen as a metric for comparison with these four methods. The performance evaluation criteria follow the evaluation protocol [41] for the query-answering task. For each dataset in

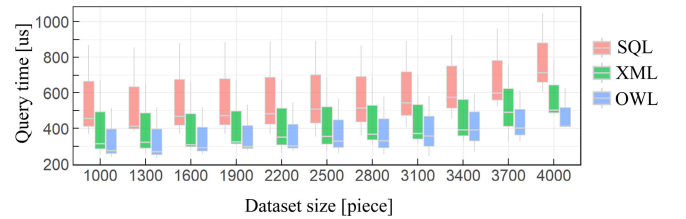


Fig. 10. Query time for different semantic Web technologies.

Table I, each triple (s, r, o) in the test set is converted into a query. QOMOU takes s and r as input and outputs a ranked list of candidate answers, E_o keeps the scores in a descending order, and $E_o = o_1, \dots, o_L$. After removing all other correct answer entities in the original test set, the rank r_o of the correct answer entity o is recorded within the candidate set E_o . If the model performs well, the correct answer entity o has a higher rank than all other candidates, as it is assigned a higher confidence score. After recording the ranks for all test examples, we calculate two types of metrics.

- 1) *Hits@N*: The proportion of test examples where the target answer entity is ranked within the top @N predictions.
- 2) *Mean Reciprocal Rank (MRR)*: The average of $(1/r_o)$ across all test examples, where r_o is the rank of the correct answer entity o .

For the data described in Table I, smaller embedding dimensions between 50 and 100 yield the best results for graph-based embedding methods. MRR, @10, @3, and @1 represent evaluation metrics of Mean Reciprocal Rank, Hits@10, Hits@3, and Hits@1, respectively. The best method in each category is highlighted. MRR and Hits@N scores vary in [0.0, 1.0], with higher scores indicating better prediction performance. Table IV presents the performance of QOMOU and graph embedding-based models on the dataset. It is shown that QOMOU outperforms all four datasets except for the FDSO dataset. This is due to the limited semantic descriptions in this dataset, leading to less effective device recognition by ECSSC.

In validating the query accuracy rate, the accuracy [36] of existing heterogeneous data integration schemes is compared. The formula calculates the accuracy rate QAR of a heterogeneous data integration scheme, measuring the proportion of successfully matched data entries out of the total integrated ones. S_R represents the number of successfully integrated data entries, and R is the total number of entries. The accuracy rate is obtained by dividing S_R by R . A higher value indicates better accuracy in data integration, i.e.,

$$QAR = \frac{S_R}{R} \times 100\%. \quad (13)$$

Experiments are conducted to validate the query accuracy of QOMOU [42]. The comparison methods include the probability factor framework [43] and the logical object-oriented interaction [44]. Furthermore, the experimental data are divided into four groups, each with 1000 data points. Additionally, for more comprehensive statistical analysis, 1000 randomly sampled data points are selected as group 5, and some noisy data are introduced into it to form group 6. This ensures that accurate fluctuation comparisons can be made after completing experiments for all six groups. Table V shows the accuracy and fluctuation ranges of three methods. It is illustrated that the accuracy of each technique fluctuates within a specific range. The accuracy in Table V is expressed as a percentage, with values ranging from [0, 100]. During the experiment, the float ratio calculates the standard deviation for each group, and the final fluctuation range is determined by combining these values, taking the minimum and maximum standard deviations across all groups. It effectively encompasses the overall fluctuation range of the entire dataset. Due to the utilization of event-class semantic integration algorithms, the accuracy of QOMOU is above average, indicating higher query accuracy. The data in Table V also show that the semantic reasoning accuracy of QOMOU is generally moderate in noisy dataset.

To compare these methods, the performance improvement value PI is given as

$$PI = \frac{NV - OV}{OV} \times \frac{1}{n} \times 100\% \quad (14)$$

where NV represents the numerical value of the model used in this work, OV represents the numerical value of another model, and n represents the number of data groups. ECSSC improves performance by approximately 7% compared to the baseline device recognition models. Compared to graph embedding-based methods, QOMOU achieves approximately

13% higher query performance. Additionally, compared to structured queries and XML, ontology-based queries improve query efficiency by 5% on average. Furthermore, a keyword-matching algorithm improves the query accuracy of heterogeneous data integration solutions by 4% on average.

VI. CONCLUSION

Industry 4.0 transforms traditional manufacturing into intelligent manufacturing. This work emphasizes the importance of semantic interoperability in the factory. Moreover, OPC UA is a practical knowledge model suitable for factory workshops. However, due to data heterogeneity, achieving unified management and communication in factories with OPC UA is highly challenging. Currently, ontology mapping methods based on OPC UA suffer from slow speed and low efficiency. Therefore, a QOMOU model is designed to solve the above problems. QOMOU maps the OPC UA information model to the resource description framework in ontology technology and creates a Web Ontology Language in a graph structure. Additionally, an ECSSC method is proposed and validated in device-type recognition, achieving approximately 7% higher accuracy than benchmark models. It addresses the semantic heterogeneity issue of devices produced by different manufacturers. Finally, compared with graph embedding-based methods, QOMOU's query performance is approximately 13% higher. Experimental results show that the query time with QUMOU is reduced by 5% on average than those with SQL and XML. Furthermore, QUMOU achieves 4% higher query accuracy on average than other state-of-the-art querying models.

In the future, we plan to refine our semantic mapping algorithm to enhance the accuracy of queries by integrating similarity models [45] of event classes [46]. We aim to link unknown entities to the manufacturing knowledge graph to enable the construction of automated equipment models and real-time data integration, thereby advancing prediction maintenance. We will use reinforcement learning to infer missing relationships in large-scale and semantically incomplete OPC UA models. Through reinforcement learning, we will design an automatic inference mechanism that leverages existing data and patterns to predict these missing relationships and assess the validity of the proposed triples. In this way, we can not only enrich the semantic information of the knowledge graph but also provide more accurate support for equipment maintenance and fault prediction, further enhancing the intelligence and reliability of production systems.

REFERENCES

- [1] A. Busboom, "Automated generation of OPC UA information models—A review and outlook," *J. Ind. Inf. Integr.*, vol. 39, no. 3, pp. 100602–100617, Mar. 2024.
- [2] K. Ahmed, I. Jeon, G. Piccialli, and B. Francesco, "From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5031–5042, Aug. 2022.
- [3] N. Zhang et al., "Physical-layer authentication for Internet of Things via WFRFT-based gaussian tag embedding," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 9001–9010, Sep. 2020.
- [4] C. Yang et al., "Ontology-based knowledge representation of industrial production workflow," *Adv. Eng. Informat.*, vol. 58, no. 2, pp. 102185–102196, Oct. 2023.

- [5] H. Yuan, J. Bi, Z. Wang, J. Yang, and J. Zhang, "Partial and cost-minimized computation offloading in hybrid edge and cloud systems," *Expert Syst. Appl.*, vol. 250, no. 15, pp. 1–13, Sep. 2024.
- [6] J. Bi, Z. Wang, H. Yuan, J. Zhang, and M. Zhou, "Cost-minimized computation offloading and user association in hybrid cloud and edge computing," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16672–16683, May 2024.
- [7] J. Bi, Z. Wang, H. Yuan, J. Qiao, J. Zhang, and M. Zhou, "Self-adaptive teaching-learning-based optimizer with improved RBF and sparse autoencoder for complex optimization problems," *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, London, U.K., pp. 7966–7972.
- [8] R. Xu et al., "Automatic semantic modeling of structured data sources with cross-modal retrieval," *Pattern Recognit. Lett.*, vol. 177, no. 7, pp. 7–14, Jan. 2024.
- [9] F. Pauker, T. Frühwirth, B. Kittl, and W. Kastner, "Systematic approach to OPC UA information model design," *Multidiscip. Digit. Publ. Inst.*, vol. 9, no. 2, pp. 28–36, Mar. 2016.
- [10] S. Cavalieri and G. Salafia, "Insights into mapping solutions based on OPC UA information model applied to the industry 4.0 asset administration shell," *CIRP J. Manuf. Sci. Technol.*, vol. 57, no. 3, pp. 321–326, Nov. 2020.
- [11] S. Kakad and S. Dhage, "Cross domain-based ontology construction via Jaccard semantic similarity with hybrid optimization model," *Expert Syst. Appl.*, vol. 178, no. 5, pp. 57–74, Sep. 2021.
- [12] Z. Hao, W. Mayer, J. Xia, G. Li, L. Qin, and Z. Feng, "Ontology alignment with semantic and structural embeddings," *J. Web Semant.*, vol. 78, no. 4, pp. 98–112, Oct. 2023.
- [13] S. De, M. Niklas, R. Brian, J. Mottok, and P. Brada, "Semantic mapping of component framework interface ontologies for interoperability of vehicle applications," *Procedia Comput. Sci.*, vol. 170, no. 3, pp. 813–818, Jul. 2020.
- [14] Y. Gutiérrez, I. Abreu, A. Montoyo, R. Muñoz, and S. Estévez-Velarde, "KD SENSOMERGER: An architecture for semantic integration of heterogeneous data," *Eng. Appl. Artif. Intell.*, vol. 132, no. 7, pp. 107854–107865, Jun. 2024.
- [15] G. Steindl, T. Frühwirth, and W. Kastner, "Ontology-based OPC UA data access via custom property functions," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Zaragoza, Spain, 2019, pp. 95–101.
- [16] S. G. Mathias, S. Schmied, and D. Großmann, "An investigation on database connections in OPC UA applications," *Procedia Comput. Sci.*, vol. 170, no. 11, pp. 602–609, Jan. 2020.
- [17] L. Peifeng, Q. Lu, Z. Xingwei, and T. Bo, "Joint knowledge graph and large language model for fault diagnosis and its application in aviation assembly," *IEEE Trans. Ind. Informat.*, vol. 20, no. 6, pp. 8160–8169, Jun. 2024.
- [18] E. Mahmoud, E. Ahmed, H. Hossam, S. Saifur, R. Jo, and H. Shin, "Energy-efficient synchronization in Industrial Internet of Things: An intelligent neighbor-knowledge approach," *IEEE Trans. Ind. Informat.*, vol. 20, no. 6, pp. 8548–8558, Jun. 2024.
- [19] C. Wang, L. Zhang, and W. Yan, "Enhancement and validation of IfcOWL ontology based on shapes constraint language (SHACL)," *Autom. Construct.*, vol. 160, no. 7, Mar. 2023, Art. no. 105293.
- [20] B. Wang, B. Song, Y. Li, Q. Zhao, and B. Tan, "Towards smart sustainable cities: Addressing semantic heterogeneity in building management systems using discriminative models," *Sustain. Cities Soc.*, vol. 62, no. 6, pp. 102367–102387, Jul. 2020.
- [21] A. Patel, V. Tiwari, M. Ojha, and O. Vyas, "Ontology-based detection and identification of complex event of illegal parking Using SPARQL and description logic queries," *Data Knowl. Eng.*, vol. 226, no. 5, pp. 120149–120167, Oct. 2023.
- [22] X. Zhang and X. Ming, "Implementation path and reference framework for Industrial Internet platform (IIP) in product service system using industrial practice investigation method," *Adv. Eng. Informat.*, vol. 51, no. 6, pp. 128–134, Oct. 2022.
- [23] T. Krijnen and J. Beetz, "An efficient binary storage format for IFC building models using HDF5 hierarchical data format," *Autom. Construct.*, vol. 113, no. 4, pp. 103134–103146, Mar. 2020.
- [24] A. Algergawy, R. Nayak, and G. Saake, "Element similarity measures in XML schema matching," *Inf. Sci.*, vol. 180, no. 24, pp. 4975–4998, Aug. 2022.
- [25] O. Sande, M. Fojcik, M. Hernes, R. Cupek, and M. Drewniak, "Implementation of OPC UA communication traffic control for analog values in an automation device with embedded OPC UA servers," *Procedia Comput. Sci.*, vol. 225, no. 4, pp. 2322–2332, Aug. 2023.
- [26] A. Liu, D. Zhang, Y. Wang, and X. Xu, "Knowledge graph with machine learning for product design," *Int. Acad. Product. Eng.*, vol. 71, no. 1, pp. 117–120, Mar. 2022.
- [27] J. Bi et al., "Long-term water quality prediction with transformer-based spatial-temporal graph fusion," *IEEE Trans. Autom. Sci. Eng.*, early access, Jan. 27, 2025, doi: [10.1109/TASE.2025.3535415](https://doi.org/10.1109/TASE.2025.3535415).
- [28] B. Idrissi, A. Mamouny, and M. Elmaallam, "RDF/OWL storage and management in relational database management systems: A comparative study," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7604–7620, Oct. 2022.
- [29] V. Naveen and A. Kumar, "An efficient and scalable SPARQL query processing framework for big data using mapreduce and hybrid optimum load balancing," *Data Knowl. Eng.*, vol. 148, no. 7, pp. 102239–102249, Oct. 2019.
- [30] N. Heidenreich and D. Mohr, "Recurrent neural network plasticity models: Unveiling their common core through multi-task learning," *Comput. Methods Appl. Mech. Eng.*, vol. 426, no. 303, pp. 113024–113040, Jun. 2024.
- [31] A. Ghada, P. Niels, and N. Goran, "Attention-based bidirectional long short-term memory networks for extracting temporal relationships from clinical discharge summaries," *J. Biomed. Informat.*, vol. 123, no. 103, pp. 103915–103923, Nov. 2021.
- [32] A. Wei, Y. Wei, H. Shao, Y. Shou, T. Meng, and K. Li, "Edge-enhanced minimum-margin graph attention network for short text classification," *Expert Syst. Appl.*, vol. 251, no. 4, pp. 124069–124083, Oct. 2024.
- [33] X. Li, S. Zhang, P. Jiang, M. Deng, X. V. Wang, and C. Yin, "Using object-oriented coupled deep learning approach for typical object inspection of transmission channel," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 347, no. 6, pp. 113864–11397, Mar. 2024.
- [34] S. Wei, M. Luo, Z. Yang, and X. Lin, "Knowledge graph based OPC UA information model automatic construction method for heterogeneous devices integration," *Robot. Comput.-Integr. Manuf.*, vol. 88, no. 24, pp. 102736–102747, Aug. 2024.
- [35] J. Bi, Z. Wang, H. Yuan, J. Zhang, and M. Zhou, "Self-adaptive teaching-learning-based optimizer with improved RBF and sparse autoencoder for high-dimensional problems," *Inf. Sci.*, vol. 630, no. 56, pp. 463–481, Jun. 2023.
- [36] V. Christou et al., "Heterogeneous hybrid extreme learning machine for temperature sensor accuracy improvement," *Expert Syst. Appl.*, vol. 203, no. 4, pp. 113864–11397, Sep. 2024.
- [37] C. Yang, R. Zheng, X. Chen, and H. Wang, "Content recommendation with two-level TransE predictors and interaction-aware embedding enhancement: An information seeking behavior perspective," *Inf. Process. Manage.*, vol. 60, no. 4, pp. 0306–4573, Jul. 2023.
- [38] J. Lu, L. T. Yang, B. Guo, Q. Li, H. Su, and G. Li, "Enhancing IoT data and semantic interoperability based on entity tree embedding under an edge-cloud framework," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3322–3338, Feb. 2023.
- [39] W. Li, W. Deng, K. Wang, L. You, and Z. Huang, "A complex-valued transformer for automatic modulation recognition," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 22197–22207, Jun. 2024.
- [40] R. Diego, M. Amira, and S. Sabrina, "Enhancing downstream tasks in knowledge graphs embeddings: A complement graph-based approach applied to bilateral trade," *Procedia Comput. Sci.*, vol. 225, no. 12, pp. 3692–3700, Jun. 2023.
- [41] Y. Sun and J. Platoš, "Abstractive text summarization model combining a hierarchical attention mechanism and multiobjective reinforcement learning," *Expert Syst. Appl.*, vol. 248, no. 15, pp. 123356–123367, Aug. 2024.
- [42] H. Li and Y. Yang, "Keyword targeting optimization in sponsored search advertising: Combining selection and matching," *Electron. Commerce Res. Appl.*, vol. 56, no. 4, pp. 117488–117501, Oct. 2022.
- [43] Y. Lina, S. Quan, N. Anne, and L. Xue, "Things of interest recommendation by leveraging heterogeneous relations in the Internet of Things," *ACM Trans. Internet Technol.*, vol. 16, no. 4, pp. 1–25, Nov. 2016.
- [44] X. Bin, T. Kanter, and R. Rahmani, "Logical interactions for heterogeneous IoT entities via virtual world mirrors in support of ambient assisted living," *J. Ambient Intell. Smart Environ.*, vol. 8, no. 5, pp. 565–580, Oct. 2016.
- [45] H. Sun and W. Xiao, "Similarity weight learning: A new spatial and temporal satellite image fusion framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, no. 10, pp. 1–17, Mar. 2022.
- [46] G. Bassi et al., "A FPGA-based architecture for real-time cluster finding in the LHCb silicon pixel detector," *IEEE Trans. Nucl. Sci.*, vol. 70, no. 6, pp. 1189–1201, Jun. 2023.



Jing Bi (Senior Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 2003 and 2011, respectively.

She is currently a Professor with the College of Computer Science, Beijing University of Technology, Beijing, China. She has over 150 publications in international journals and conference proceedings. Her research interests include distributed computing, cloud computing, large-scale data analytics, machine learning, and performance optimization.

Prof. Bi is currently an Associate Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS.



Ziqi Wang (Graduate Student Member, IEEE) received the B.E. degree in Internet of Things from Beijing University of Technology, Beijing, China, in 2022, where he is currently pursuing the master's degree with the College of Computer Science.

His research interests include mobile-edge computing, task scheduling, intelligent optimization algorithms, and big data.

Mr. Wang received the Best Paper Award in 2024 ICAIS and ISAS and the Best Application Paper Award in the 21st IEEE ICNSC.



Rina Wu (Student Member, IEEE) received the B.E. degree in software engineering from Beijing Technology and Business University, Beijing, China, in 2023. She is currently pursuing the master's degree with the Faculty of Information Technology, School of Software Engineering, Beijing University of Technology, Beijing.

Her research interests include Industrial Internet of Things, OPC UA, industrial big data, and machine learning.



Jia Zhang (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Illinois at Chicago, Chicago, IL, USA, in 2000.

She is currently the Cruse C. and Marjorie F. Calahan Centennial Chair in Engineering and a Professor with the Department of Computer Science, Lyle School of Engineering, Southern Methodist University, Dallas, TX, USA. Her research interests emphasize the application of machine learning and information retrieval methods to tackle data science

infrastructure problems, with a recent focus on scientific workflows, provenance mining, software discovery, knowledge graph, and interdisciplinary applications of all of these interests in Earth science.



Haitao Yuan (Senior Member, IEEE) received the Ph.D. degree in computer engineering from New Jersey Institute of Technology (NJIT), Newark, NJ, USA in 2020.

He is currently a Deputy Director of the Department of Science and Technology Innovation, Wenchang International Aerospace City, Wenchang, Hainan, China. He is currently an Associate Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His research interests include the

Internet of Things, edge computing, deep learning, data-driven optimization, and computational intelligence algorithms.

Dr. Yuan is named in the world's top 2% of Scientists List. He received the Chinese Government Award for Outstanding Self-Financed Students Abroad, the 2021 Hashimoto Prize from NJIT, the Best Work Award in the 17th ICNSC, and the Best Student work Award Nominees in 2024 IEEE SMC. He is an Associate Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE INTERNET OF THINGS JOURNAL, and *Expert Systems With Applications*.



MengChu Zhou (Fellow, IEEE) received the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

Then, he joined New Jersey Institute of Technology, Newark, NJ, USA, where he is currently a Distinguished Professor. He has over 900 publications, including 12 books, 600+ journal papers (450+ in IEEE transactions), 28 patents, and 29 book chapters. His interests are in Petri nets, automation, Internet of Things, and big data.

Dr. Zhou is a Fellow of IFAC, AAAS, CAA, and NAI.