2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 5-8, 2025. Vienna, Austria

Ontology-Based Semantic Integration of Multi-Source Heterogeneous Industrial Devices

Rina Wu[1], Jing Bi[1], Ziqi Wang[2], Haitao Yuan[3], Hailiang Zhao[2], Yanan Liu[1] and Jia Zhang[4]

[1]College of Computer Science, Beijing University of Technology, Beijing 100124, China
[2]School of Software Technology, Zhejiang University, Ningbo 315100, China
[3]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[4]Dept. of Computer Science in Lyle School of Engineering, Southern Methodist University, Dallas, TX, USA

2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 979-8-3315-3358-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/SMC58881.2025.11343003

***Abstract*—The 3C industry uses industrial internet platforms to tackle data fragmentation, poor collaboration, and semantic mismatches, driving digital transformation. Although the Object Linking and Embedding for Process Control Unified Architecture (OPC UA) standard supports industrial automation, it faces several limitations, including poor compatibility, ambiguous semantics, and incomplete models, which hinder automated reasoning and system interoperability. To address these issues, this work proposes an integrated framework that combines offline classification of heterogeneous devices using a Character-level Text Convolutional Neural Network (CTCNN) with online automated reasoning over OPC UA information models. CTCNN leverages character-level embeddings and convolutional layers to achieve fine-grained and accurate device type recognition. Furthermore, a deterministic Markov decision process is formulated, and an intelligent agent is trained to perform automatic reasoning based on the OPC UA model structure. This hybrid framework enables effective OPC UA device integration and interconnection. Experimental results demonstrate that CTCNN achieves up to an 18% improvement in device type identification precision, while the combined offline recognition and online reasoning approach enhances the accuracy of automatic OPC UA information model inference by approximately 12% compared to state-of-the-art methods.**

***Index Terms*—OPC UA, semantic adaptation, syntactic interoperability, device type identification, and automatic inference.**

I. Introduction

Nowadays, the 3C manufacturing industry faces several challenges. First, fragmented and isolated data storage impedes effective data mining. Second, low coordination within the industry chain and scattered resources hinder semantic integration and big data analytics [1]. Therefore, industrial internet platforms are essential to solve these problems by comprehensive data digitization [2]. While OPC UA addresses industrial automation challenges through communication and information modeling, its platform independence and scalability drive its adoption [3]. However, many legacy devices lack support, and its models are semantically incomplete, leading to discrepancies that hinder automated reasoning and knowledge extraction.

To address this issue, Liu *et al.* [4] propose leveraging OPC UA and Manufacturing Technology Connect to enable seamless interoperability among different machine tool systems. This approach reduces fragmentation in manufacturing to enable cross-platform communication, but varying systems and devices hinder integration and interoperability. Han *et al.* [5] introduce an OPC UA-based data framework specifically designed for smart production workshops in the lithium-ion battery manufacturing sector. This framework aims to optimize data exchange and integration across various production lines. However, it still faces challenges related to the diversity of devices and the need for real-time data processing. Meanwhile, Panda *et al.* [6] focus on using OPC UA to enable real-time data exchange between field devices, supporting industrial communication for time-sensitive applications. However, these approaches either require time-consuming methods for constructing accurate OPC UA information models or prioritize real-time communication at the cost of model precision. As a result, there is often a trade-off between achieving high model accuracy and ensuring the speed of communication required for industrial environments.

To enhance the accuracy of model construction, Stogl *et al.* [7] propose a method that utilizes OPC UA for skill modeling and service-oriented orchestration, enabling a plug-and-produce capability. Loskyll *et al.* [8] introduce a manufacturing service discovery and orchestration approach based on semantic technologies, facilitating automatic matching and dynamic orchestration of manufacturing services. Bakakeu *et al.* [9] develop an automated inference approach for OPC UA data models, integrating semantic methods and ontology-based reasoning to enhance OPC UA's intelligence in industrial environments. Schiekofer *et al.* [10] apply the SPARQL Protocol and RDF Query Language to efficiently query OPC UA device data, allowing it to be accessed like a knowledge graph instead of relying solely on hierarchical structures. Converting the OPC UA model into an ontology-based system improves interoperability and intelligence. However, most web ontology tools are too slow for real-time factory environments [11], which makes direct reasoning on the OPC UA model a more viable option.

This work proposes an online reasoning approach integrated with offline Character-level Text Convolutional Neural Network (CTCNN)-based OPC UA device identification, aiming to automate the construction of unknown OPC UA device information models. By combining text classification,

This work was supported by the Beijing Natural Science Foundation under Grants L233005 and 4232049, the National Natural Science Foundation of China under Grants 62473014 and 62173013, and in part by Beihang World TOP University Cooperation Program (Corresponding author: Jing Bi.)

Authorized licensed use limited to: Zhejiang University. Downloaded on January 31,2026 at 04:57:06 UTC from IEEE Xplore. Restrictions apply.

character-level embeddings, and direct reasoning, the method effectively tackles the challenges of device model generation in dynamic industrial environments [12]. It proposes a standardized framework for automatic OPC UA model generation and a CTCNN-based algorithm using character embeddings to recognize new device types [13]. An intelligent agent reasons over incomplete models via multi-hop relational paths in the embedding space. This enables relation completion, consistency verification [14], and question answering, thereby supporting a more comprehensive semantic understanding.

II. UNIFIED FRAMEWORK FOR DEVICE RECOGNITION

This section presents an automatic method for constructing the information model of an OPC UA-based heterogeneous device integration architecture. The method leverages several techniques, including device type identification using CTCNN, character-based word embedding, and the Proximal Policy Optimization (PPO) algorithm for training the agent. The overall process architecture is illustrated in Fig. 1.

A. Device Classification Task based on CTCNN

Device classification can be viewed as classifying OPC UA information models, which can be approached as text classification. Character-level convolutional neural network for text classification encodes character sequences using a one-hot encoding method, eliminating the need for pre-trained word embeddings [15]. This approach effectively mitigates semantic ambiguity in textual data when a protocol parser converts protocol frames from newly connected devices into JSON format. As illustrated in Fig. 2, it improves the accuracy of text interpretation in structured data. CTCNN begins by generating an alphabet from the device data frame corpus and randomly initializing embeddings for each character. It then obtains the embeddings for the input character sequence using this alphabet. After transforming characters into embeddings, a text sequence of length m is represented as an embedding matrix $D = \left[d_1^T, d_2^T, ..., d_m^T\right]$, where $d_i \in \mathbb{R}^{1\times p}$ denotes the embedding of the i-th character with dimension p. If the sequence exceeds length m, it is truncated; otherwise, it is padded with zeros. The embedding matrix D is then passed through a convolutional layer, where multiple convolution kernels of varying sizes extract local textual features. The extracted feature b_i is expressed as follows:

$$b_i = g\left(w \cdot d_{i:i+q-1} + c\right), \tag{1}$$

where $g(\cdot)$ represents the activation function, while $d_{i:i+q-1}$ denotes the submatrix extracted from the i-th to the $i+q-1$-th row of the embedding matrix. The parameter q corresponds to the window size, $w \in \mathbb{R}^{pq}$ is the convolutional filter, and $c \in \mathbb{R}$ represents the bias term. Once the convolutional kernel processes the embedding matrix, the resulting feature map is given by $\mathbf{b} = [b_1, b_2, ..., b_{m-q+1}]$. Then, the $1-$MAX pooling strategy is used to extract the most important features from the text sequence. It selects the highest value from the feature map $\mathbf{b}$ generated by the convolutional layer, ensuring that the most important information is preserved, *i.e.*,

$$b' = \max\left(\mathbf{b}\right), \tag{2}$$

where b' denotes the highest value within the feature map $\mathbf{b}$.

The features extracted from the pooling layer are combined into a feature vector, which captures various types of information from the device data frame. This vector is then passed through a fully connected layer, and the SoftMax function is applied for device classification. The cross-entropy loss (CEL) is used to minimize the discrepancy between the predicted outputs and the actual labels. During inference, the category with the highest probability is selected as the final prediction. The CEL function is defined as follows:

$$\mathcal{L} = \frac{1}{N}\sum_j \mathcal{L}_j = \frac{1}{N}\sum_j \sum_{k=1}^{K} z_{jk} \log\left(r_{jk}\right), \tag{3}$$

where K denotes the total number of classes, z_{jk} is a binary function that takes the value 1 when the true class of sample j is k, and zero otherwise. r_{jk} represents the predicted probability that the sample j belong to class k.

B. Deterministic Markov Decision Process

Fact prediction algorithms are useful for query answering and link prediction but require evaluating all candidate entities and relations, resulting in high computational overhead. This becomes particularly challenging with large OPC UA models, as illustrated in Fig. 3. To address this, we propose an OPC UA reasoning model that efficiently traverses the information model to quickly identify the correct answer, avoiding the need to evaluate all entities and relations. We frame the query answering task as a sequential decision-making problem, constrained by a finite time horizon, and model the OPC UA information as a Markov Decision Process (MDP). By training a reinforcement learning (RL) agent on known facts [16], the agent learns to effectively traverse the knowledge graph starting from the source entity. When presented with a query in the form of $(x, y, ?)$ or $(x, ?, z)$, the RL agent aims to identify the appropriate response entities or relations without relying on pre-calculated paths. The agent navigates the graph by following plausible triples to link entities, ultimately reaching the query target. Query answering is framed as a finite-horizon MDP, where the RL agent learns to efficiently find the correct entities or relations.

1) Problem Definition: As the OPC UA information model is organized in a graph structure, we formally define it as $\mathcal{G} = \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$, where $\mathcal{G}$ represents the entire OPC UA information model, $\mathcal{X}$ represents the set of entities, corresponding to the OPC UA nodes, $\mathcal{Y}$ represents the set of relations, corresponding to the OPC UA references, and $\mathcal{Z} = \{(x, y, z) \mid x, z \in \mathcal{X}, y \in \mathcal{Y}\}$ represents the set of triples, which are explicitly encoded. In this data structure, each directed link $l = (x, y, z) \in \mathcal{Z}$ signifies a fact.

2) Reinforcement Learning Agent Solution: We formalize the agent's search process as a deterministic MDP, beginning from $\mathcal{Z}$. The RL agent progressively chooses an outgoing edge in sequence l and moves to a new entity, continuing this process until the target is reached. The components of this process include:

- **States:** The RL agent is determined $v_t = \langle e_t, (x_q, y_q)\rangle$ starting from each time step$t$, where e_t represents the

Authorized licensed use limited to: Zhejiang University. Downloaded on January 31,2026 at 04:57:06 UTC from IEEE Xplore. Restrictions apply.

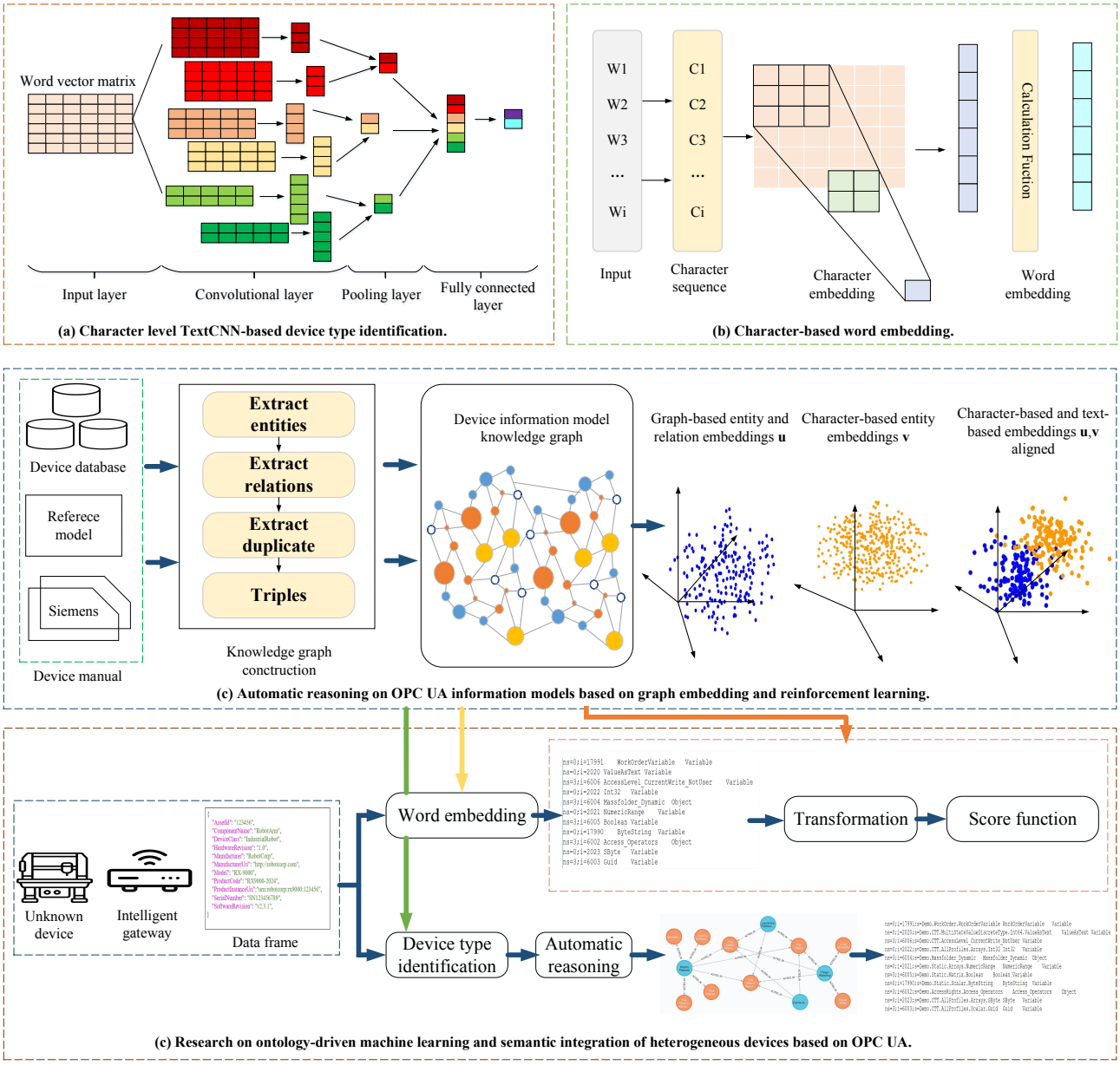


Fig. 1. The integrated model for automatic OPC UA information model construction and device interconnection.

state node encountered at each time step t, and (x_q, y_q) represents the source entity and relation of the query. In this formulation, e_t holds state-dependent information, while (x_q, y_q) provides global context to guide the search. Since OPC UA entities and relations are discrete symbols, we use knowledge graph embeddings to represent them as low-dimensional vectors, making the solution scalable. Each circle and line segment in $\mathcal{Z}$ is given a compact multi-element vector, with $x \in \mathbb{R}^d$ for entities and $y \in \mathbb{R}^d$ for relations. The state vector x_t is represented by concatenating the dense vectors of e_t, x_q, and y_q.

- **Actions:** Starting from the initial node of the answer-search graph, the agent must effectively traverse the knowledge graph by sequentially following high-probability facts that lead to the correct answer. At time step t, the set of possible actions $P_t \in P$ consists of all paths extending outward from the current entity e_t, *i.e.*, $P_t=\{(y, e) \mid (e_t, y, e) \in \mathcal{Z}\}$. This means the agent can choose which outgoing edge to follow based on the query being answered. Exceeding a certain time limit is not allowed, and the intelligent agent must complete the search within the specified time frame. A line segment that can find its path that is equipped to each P_t, acting as a "stop" action when the process of searching for the correct answer has been carried out for a specified number of steps T.
- **Transitions:** During OPC UA knowledge graph traver-

Authorized licensed use limited to: Zhejiang University. Downloaded on January 31,2026 at 04:57:06 UTC from IEEE Xplore. Restrictions apply.

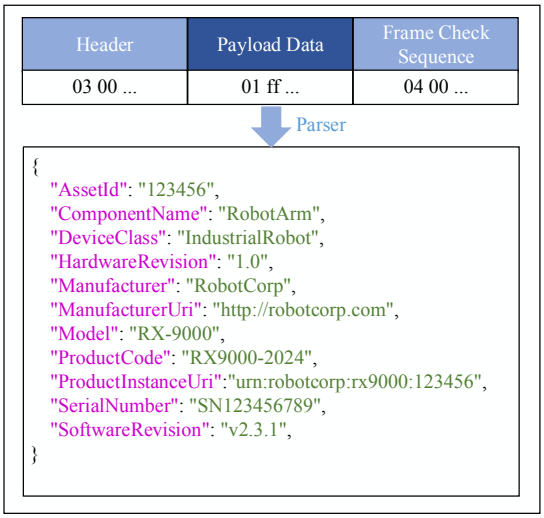


Fig. 2. OPC UA device protocol frame converted to JSON format.

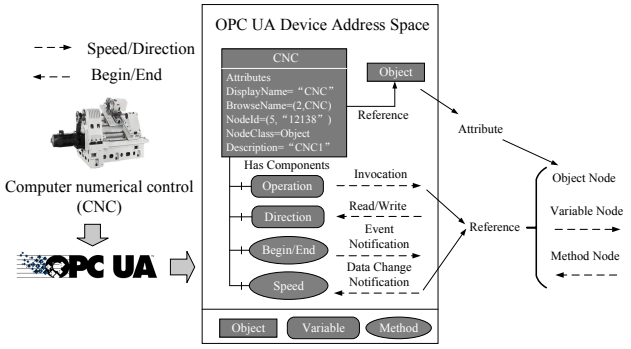


Fig. 3. OPC UA device information model (using computer numerical control as an example).

sal, the environment evolves controllably under $\mathcal{G}$'s rules. The agent's state x_t updates upon reaching new entities via selected edges, while the query-answer pair persists until task completion.

- **Rewards:** Due to imperfect OPC UA-to-knowledge graph mapping, agents must represent new concepts and infer through fuzzy, incomplete paths. A reward mechanism guides them toward likely correct answers amid uncertainty:

$$U(x_t)=U_b(x_t)+(1-U_b(x_t))\cdot h(x_q,y_q,e_t), \quad (4)$$

$$U_b(x_t)=\begin{cases}1 & \text{if } (x_q,y_q,e_t)\in\mathcal{G}\\ 0 & \text{otherwise,}\end{cases} \quad (5)$$

where (4) represents the target scoring function and (5) represents the basic scoring function, with $h(x_q,y_q,e_t)$ evaluating the rationality of the embedding function for x_q,y_q,e_t. If the target e_t is the correct object according to $\mathcal{G}$, the RL agent receives a reward of 1. Otherwise, the agent is assigned a "not-good" score, which is computed using the pre-trained embedding score function $h(x_q,y_q,e_t)$. Here, h is kept in its general form, allowing substitution with any advanced embedding model.

C. Training the Agent Using Proximal Policy Optimization

We train the agent using a policy-based method with the PPO algorithm. PPO is a technique that guides systematic steps toward gradual improvement, refining the policy through several epochs of stochastic gradient ascent. Additionally, PPO is well-suited for reinforcement learning tasks in non-aggregated, continuous action environments and can handle both discrete and continuous state spaces [17].

III. EXPERIMENTAL RESULTS AND DISCUSSION

A. Dataset Description

Due to the lack of existing datasets, we constructed four experimental datasets from official OPC UA companion specification node sets. 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 (IR), sorting machine (SOM), marking machines (MAM), patching machine (PM), and scribing machine (SCM), each following its respective OPC UA specification. The four datasets include FAPS Empty Server (FES), Unified Automation Demo Server (UADS), FAPS Demo Server One (FDSO), and FAPS Demo Server Two (FDST).

B. Experimental Results

To evaluate the device recognition accuracy of CTCNN, it is compared with three text classification models.

TextRNN [18]: CNN is replaced by Bidirectional Long Short-Term Memory (Bi-LSTM), which extracts features from the input character embeddings with Bi-LSTM, and the SoftMax function is used to classify OPC UA device types.

TextRNN_Att [19]: An attention mechanism is additionally added based on TextRNN.

DPCNN [20]: It combines CNN with a deep pyramid structure, enabling it to capture long-distance dependencies in the device-type text.

The accuracy of the model in classifying OPC UA devices is evaluated using the following metrics: Precision (*P*), Recall (*R*), and F1-score (*F1*), which are calculated as follows:

$$P=\frac{T_2}{F_1+T_2} \quad (6)$$

$$R=\frac{T_2}{F_2+T_2} \quad (7)$$

$$F1=2\times\frac{P\times R}{P+R} \quad (8)$$

where T_1, T_2, F_1, and F_2 defined in Table I.

The experimental results are shown in Table II. CTCNN achieves the best performance in device type identification, with the highest Precision, Recall, and F1-score, as well as the fewest misclassified samples. These results highlight its strong generalization and high accuracy. CTCNN's superiority comes from its CNN architecture, which effectively extracts textual features while capturing temporal patterns. TextRNN suffers from vanishing gradients, weakening its performance. TextRNN_Att improves precision with attention but still underperforms CNNs in local feature extraction.

Authorized licensed use limited to: Zhejiang University. Downloaded on January 31,2026 at 04:57:06 UTC from IEEE Xplore. Restrictions apply.

DPCNN enhances classification with deeper conv layers, reducing errors compared to RNNs. Overall, CTCNN outperforms all baselines due to better feature extraction, deeper architecture, and optimized parameters, making it the most robust model.

TABLE I
NOTATIONS OF T_1, T_2, F_1, AND F_2

Actual result	**Prediction result**	
	0	1
0	True Negatives (T_1)	False Positive (F_1)
1	False Negative (F_1)	True Positive (T_2)

TABLE II
COMPARISON OF THE PERFORMANCE OF DEVICE TYPE IDENTIFICATION MODELS

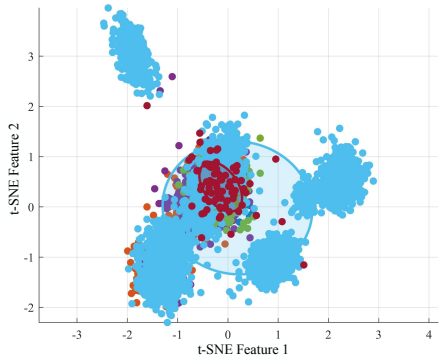
Model	T_1	F_1	P	R	$F1$
TextRNN	1850	450	0.8326	0.8452	0.8322
TextRNN_Att	1920	280	0.9192	0.9162	0.9162
DPCNN	1930	260	0.9237	0.9217	0.9238
CTCNN	1980	120	0.9736	0.9657	0.9821

C. Ablation Experiment

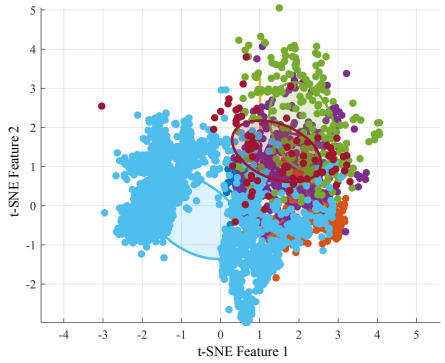
Ablation study evaluates OPC UA clustering using automatic inference-only, CTCNN-only, and combined, with t-SNE visualization from ComplEx embeddings. As shown in Fig. 4, similar concepts tend to form clusters. CTCNN effectively improves the accuracy of device identification. The maximum mean distances in Figs. 4(a), 4(b), and 4(c) are 1.0744, 1.0204, and 0.7902, respectively, with the average distance decreasing by 26.45%. Table III compares our reasoning agent with SOTA embedding methods across four datasets. The results show our agent achieves higher scoring accuracy [21], outperforming baselines and validating the model's strong reasoning capability.

IV. CONCLUSION

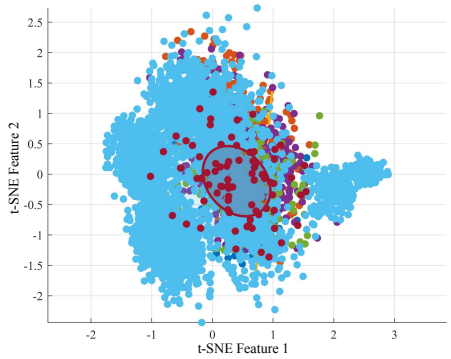
In industrial automation, it is crucial to ensure compatibility between traditional devices and the Object Linking and Embedding for Process Control Unified Architecture (OPC UA) protocol. At the same time, accurately defining the semantics of the OPC UA information model is essential for enabling efficient automated reasoning and semantic interoperability. However, addressing the challenges of poor real-time performance due to offline device classification and low accuracy from online automatic reasoning remains difficult. To overcome these issues, this work combines the Character-level Text Convolutional Neural Network (CTCNN) with a trained agent for automatic reasoning on the OPC UA information model, providing both high real-time performance and high accuracy. In the first phase of this architecture, CTCNN is used for device type recognition. This serves as the foundation for the second phase, which involves embedding the OPC UA knowledge graph. The reasoning task is then formulated as a deterministic MDP and solved by a policy-based reinforcement learning agent. CTCNN improves device type classification accuracy by up to 18%, while the inference accuracy increases by 12%.



(a) Clustering results of OPC UA device nodes using only automatic inference.



(b) Clustering results of nodes using only the CTCNN-based device type identification model.



(c) Clustering results of OPC UA nodes using both methods.

Fig. 4. t-SNE projection of OPC UA information model embedding.

REFERENCES

[1] S. Zhang, Y. Wu, X. Zhang, Z. Feng, L. Wan, and Z. Zhuang, "Relation-Aware Heterogeneous Graph Network for Learning Intermodal Semantics in Textbook Question Answering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 9, pp. 11872–11883, Sept. 2024.

Authorized licensed use limited to: Zhejiang University. Downloaded on January 31,2026 at 04:57:06 UTC from IEEE Xplore. Restrictions apply.

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

Dataset	Score	Model				
		TransE	DistMult	ComplEx	HolE	Ours
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**

[2] H. Zhao, Z. Wang, G. Cheng, W. Qian, P. Chen, J. Yin, S. Dustdar, and S. Deng, "Online Workload Scheduling for Social Welfare Maximization in the Computing Continuum," *IEEE Transactions on Services Computing,* doi: 10.1109/TSC.2025.3570845.

[3] F. Zhu, J. Xu, Z. Li, K. Yang, and J. Zhang, "Automatic Estimation of Fresnel Zones in Migrated Dip-Angle Gathers Using Semantic Segmentation Model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, no. 5915214, pp. 1–14, Jan. 2024.

[4] S. Shin, "An OPC UA-Compliant Interface of Data Analytics Models for Interoperable Manufacturing Intelligence," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3588–3598, May. 2021.

[5] Y. Han, Y. Hu, Y. Wang, G. Jia, C. Ge, C. Zhang, and X. Huang, "Research and Application of Information Model of a Lithium Ion Battery Intelligent Manufacturing Workshop Based on OPC UA," *Batteries*, vol. 10, no. 1, pp. 52–64, Dec. 2020.

[6] R. Cupek, M. Drewniak, and A. Ziebinski, "Information Models for a New Generation of Manufacturing Systems–a Case Study of Automated Guided Vehicle," *2019 IEEE International Conference on Systems, Man and Cybernetics, (SMC)*, 2019, Bari, Italy, pp. 858–864.

[7] J. Pfrommer, D. Stogl, K. Aleksandrov, S.E. Navarro, B. Hein, and J. Beyerer, "Plug and produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems," *Automa tisierungstechnik*, vol. 63, no. 10, pp. 52–64, Oct. 2019.

[8] P. Patlakas, I. Christovasilis, L. Riparbelli, F. Cheung, and E. Vakaj, "Semantic web-based automated compliance checking with integration of Finite Element analysis," *Advanced Engineering Informatics*, vol. 61, no. 6, pp. 132–164, Aug. 2024.

[9] C. Yang, Y. Zheng, X. Tu, R. Ala-Laurinaho, J. Autiosalo, O. Seppänen, and K. Tammi, "Ontology-based knowledge representation of industrial production workflow," *Advanced Engineering Informatics*, vol. 58, no. 8, pp. 46–74, Oct. 2023.

[10] M. Bakken, and A. Soylu, "Chrontext: Portable SPARQL queries over contextualised time series data in industrial settings" *Expert Systems with Applications*, vol. 226, no. 4, pp. 57–74, Sep. 2023.

[11] J. Bi, Z. Wang, H. Yuan, J. Zhang, and M. Zhou, "Cost-Minimized Partial Computation Offloading in Cloud-Assisted Mobile Edge Computing Systems," *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Honolulu, Oahu, HI, USA, 2023, pp. 5052–5057.

[12] 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 of Things Journal*, vol. 11, no. 9, pp. 16672–16683, May 2024.

[13] O. Sande, M. Fojcik, 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 Computer Science*, vol. 225, no. 4, pp. 2322–2332, Aug. 2023.

[14] 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," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, London, United Kingdom, pp. 7966–7972.

[15] X. Li, S. Zhang, P. Jiang, M. Deng, X. Vincent Wang, and C. Yin, "Using Object-oriented Coupled Deep Learning Approach for Typical Object Inspection of Transmission Channel," *International Journal of Applied Earth Observation and Geoinformation*, vol. 347, no. 6, pp. 113864–11397, Mar. 2024.

[16] J. Bakakeu, J. Franke, S. Baer, and H. Klos, Joern Peschke, "Reasoning over OPC UA Information Models using Graph Embedding and Reinforcement Learning," *2020 International Conference on Artificial Intelligence for Industries (AI4I)*, 2020, California, USA, pp. 40–47.

[17] J. Bi, Z. Wang, H. Yuan, X. Wu, J. Zhang, and M. Zhou, "Long-term Water Quality Prediction with Transformer-based Spatial-Temporal Graph Fusion," *IEEE Transactions on Automation Science and Engineering*, doi: 10.1109/TASE.2025.3535415.

[18] G. de la Cruz, M. Lira, O. Luaces, and B. Remeseiro, "Eye-LRCN: A Long-Term Recurrent Convolutional Network for Eye Blink Completeness Detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5130–5140, Apr. 2024.

[19] K. Zhang, A. Li, B. Lin, Y. Jiang, J. Chen, Y. Zhang, and Y. dong, "DFINet: Dual-Way Feature Interaction Network for Long-Term Series Forecasting," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, no. 2527112, pp. 1–12, Jul. 2024.

[20] X. Shen, M. Shao, S. Pan, T. Yang, X. Zhou, and K. Li, "Domain-Adaptive Graph Attention-Supervised Network for Cross-Network Edge Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 12, pp. 17842–17855, Dec. 2024.

[21] H. Yuan, Q. Hu, M. Wang, S. Wang, J. Bi, *et al.*, "Data-Filtered Prediction With Decomposition and Amplitude-Aware Permutation Entropy for Workload and Resource Utilization in Cloud Data Centers," *IEEE Internet of Things Journal*, vol. 12, no. 12, pp. 19189–19201, Jun. 2025.

Authorized licensed use limited to: Zhejiang University. Downloaded on January 31,2026 at 04:57:06 UTC from IEEE Xplore. Restrictions apply.