



Construction of fault diagnosis system for control rod drive mechanism based on knowledge graph and Bayesian inference

Xue-Jun Jiang^{1,2} · Wen Zhou^{1,2} · Jie Hou¹

Received: 26 July 2022 / Revised: 25 November 2022 / Accepted: 30 November 2022 / Published online: 21 February 2023

© The Author(s), under exclusive licence to China Science Publishing & Media Ltd. (Science Press), Shanghai Institute of Applied Physics, the Chinese Academy of Sciences, Chinese Nuclear Society 2023

Abstract

Knowledge graph technology has distinct advantages in terms of fault diagnosis. In this study, the control rod drive mechanism (CRDM) of the liquid fuel thorium molten salt reactor (TMSR-LF1) was taken as the research object, and a fault diagnosis system was proposed based on knowledge graph. The subject–relation–object triples are defined based on CRDM unstructured data, including design specification, operation and maintenance manual, alarm list, and other forms of expert experience. In this study, we constructed a fault event ontology model to label the entity and relationship involved in the corpus of CRDM fault events. A three-layer robustly optimized bidirectional encoder representation from transformers (RBT3) pre-training approach combined with a text convolutional neural network (TextCNN) was introduced to facilitate the application of the constructed CRDM fault diagnosis graph database for fault query. The RBT3-TextCNN model along with the Jieba tool is proposed for extracting entities and recognizing the fault query intent simultaneously. Experiments on the dataset collected from TMSR-LF1 CRDM fault diagnosis unstructured data demonstrate that this model has the potential to improve the effect of intent recognition and entity extraction. Additionally, a fault alarm monitoring module was developed based on WebSocket protocol to deliver detailed information about the appeared fault to the operator automatically. Furthermore, the Bayesian inference method combined with the variable elimination algorithm was proposed to enable the development of a relatively intelligent and reliable fault diagnosis system. Finally, a CRDM fault diagnosis Web interface integrated with graph data visualization was constructed, making the CRDM fault diagnosis process intuitive and effective.

Keywords CRDM · Knowledge graph · Fault diagnosis · Bayesian inference · RBT3-TextCNN · Web interface

1 Introduction

The control rod drive mechanism (CRDM) is the actuator of reactor startup, power regulation, and emergency shutdown under accident conditions in nuclear power plants (NPPs) [1]. Once a malfunction occurs in the CRDM, it will considerably threaten the safety and reliability of the entire reactor operation. It is not easy to obtain accurate

knowledge to establish a precise CRDM physical simulation model. Additionally, it is difficult to mine fault feature information through sufficient actual fault data samples associated with CRDM. However, in the process of CRDM design, functional testing, and operation and maintenance, many technique documents, historical unstructured data, and huge amounts of empirical expert knowledge were accumulated. All these provide rich fault knowledge, including fault symptoms description, troubleshooting methods, and fault propagation paths. It is difficult to extract and utilize this knowledge due to its complexity and semantic fuzziness. Therefore, it is of great significance for the plant's operational safety to ascertain the plant status through efficient knowledge-based fault diagnosis approaches, which may largely relieve the operator of mental pressure and assist the operator to make effective decisions to reduce the loss caused by the occurrence of a fault event.

This work was supported by the Young Potential Program of Shanghai Institute of Applied Physics, Chinese Academy of Sciences (No. E0553101).

✉ Jie Hou
houjie@sinap.ac.cn

¹ Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China

² University of Chinese Academy of Sciences, Beijing 100049, China

Unlike data-driven or physics-based fault diagnosis approaches, the knowledge-based fault diagnosis approaches, which place no requirement on complete operation data or a precise mathematical model, are very effective and have explicit interpretations [2, 3]. Accordingly, many knowledge-based fault diagnosis methodologies applied in the field of NPPs have been proposed in the literature, yet the fault diagnosis is subject to limited measurements of fault data in real NPPs. One of the typical applications is the fault diagnosis expert system (ES). Qian et al. [4] built a fault diagnosis expert system based on the event-triggering mechanism in NPPs, and the belief rule base was constructed by Access relational database. Wang et al. [5] developed a rule-based diagnostic platform and applied the multilevel flow model to represent the fault knowledge, which was acquired from domain experts and textbooks. Nevertheless, although fault diagnosis ES is consistent with the habits of human thinking and makes good use of unstructured data, such as expert knowledge, the complexity of knowledge representation restricts the performance of fault diagnosis ES, and it does not take into account the efficiency of knowledge storage and querying. With the development of graph theory and probability statistic theory, the comprehensive graphical formalism for fault knowledge representation and fault uncertain reasoning has become extremely popular. Liu et al. [6] proposed a signed direct graph (SDG) fault diagnosis model integrated with a decision table to solve the time delay problem and obtain the fault propagation path in NPPs. Zhao et al. [7] designed a fault diagnosis expert system based on a dynamic uncertain causality graph for NPP secondary loop, and the rules were presented by a causality graph, which was different from the traditional IF–THEN production rules representation. Ma et al. [8] proposed a simplified SDG model for fault diagnosis to tackle the low efficiency of SDG, and a hybrid method, combining principal component analysis with support vector machine for fault detection and assessing the severity of the fault. Shi et al. [9] proposed a fault diagnosis strategy based on the directed graph model and built a knowledge supplement based on probability and statistics to achieve blind spot diagnosis and location.

From the results of the current literature research, the existing NPP fault diagnosis methodologies based on the graph model mainly revolve around the directed graphs, which consist of a set of vertices connected by directed edges (i.e., arcs). Notably, previous research related to NPP fault diagnosis based on the directed graph mainly focused on the SDG method, which utilizes semiotic analysis to describe the relationship between the vertices. Therefore, SDGs are not convenient for storage and querying due to the lack of exclusive efficient query language and storage tools. In practice, especially in the aspect of NPP fault diagnosis, this is inconvenient for the operator's human–machine interaction. Furthermore, SDGs for fault diagnosis can only

store limited fault knowledge, which is not sufficient to help the operator quickly grasp the occurred exception in NPPs. Therefore, it is necessary to develop a method that is relatively more comprehensive and efficient in utilizing unstructured fault knowledge.

The knowledge graph, which has evolved from the directed graph, is a new form of graphical knowledge representation. The modern knowledge graph, commonly utilized in knowledge bases similar to Google for searching, question and answer, decision making, artificial intelligence reasoning, and other aspects, stems from the 2012 announcement of Google. The knowledge graph is composed of nodes (i.e., vertices) and directed edges (i.e., relationships). Each node in the knowledge graph is called an entity. The knowledge graph is superior to and more complex than a knowledge base because it applies a reasoning engine to generate new knowledge and integrates one or more information sources. In addition, a knowledge graph has a customized and efficient query language and storage database, which render it better adaptability in search, query, and decision making. In practice, knowledge graphs can be classified into two types: domain knowledge graphs and enterprise knowledge graphs [10]. Domain knowledge graphs have attracted extensive attention in both industrial and academic fields. Diverse domain-specific knowledge graphs have been published in the fields of medicine, finance, social media, and energy. Existing review articles have provided a detailed summary with respect to the development of knowledge graphs [10, 11]. Knowledge graphs provide a concise and intuitive abstraction for a variety of domains, where edges capture the relationships between the entities inherent in unstructured data.

In this study, we focused on the application of fault diagnosis analysis combined with a domain-specific knowledge graph. Deng et al. [12] constructed a top-down fault diagnosis event logic knowledge graph, which provides decision support for autonomous fault diagnosis of a robot transmission system. Liu et al. [13] proposed a fault diagnosis approach for mechanical equipment, which combines a one-dimension convolutional neural network (CNN), gated recurrent unit, attention mechanism, and a knowledge graph. The proposed model, verified over rolling bearing datasets under different loads, has shown good performance in fault diagnosis accuracy. Liu et al. [14] proposed a lightweight graph neural network model combined with an electrical equipment failure knowledge graph, which is derived from operational and maintenance records. The simulation results show that the effectiveness and robustness of the proposed method in the aspect of mining transformer concurrent faults are superior to conventional CNNs. Indeed, these excellent research works in the general engineering fields have revealed the superiority of knowledge graphs in fault diagnosis. However, applying

knowledge graphs to NPPs has rarely been reported, especially in terms of fault diagnosis of key equipment or systems in NPPs. Pakonen et al. [15] built a semantic web ontology to represent knowledge about overall nuclear instrumentation and control (I&C) architecture. They demonstrated that this technique can be useful in building rich knowledge models, answering complex queries, and improving nuclear safety. Ontologies are the standard knowledge representation formalisms, which can be employed to define and explain the semantics of the terms used to label and describe the nodes and edges in knowledge graphs [10]. Liu et al. [16] constructed a knowledge graph model of the transformer in China Qinshan NPP for fault maintenance, which can automatically extract the topic, key technology, title, proposal status, document source, and agenda items of technical documents.

The aforementioned study on the knowledge graph technique is a positive step forward in addressing fault diagnosis in NPPs. However, NPPs are relatively reluctant in adopting new technology. To bridge the practical gaps between knowledge graph and fault diagnosis in NPPs, we proposed a pilot scheme for fault diagnosis based on knowledge graph. The CRDM of the liquid fuel thorium molten salt reactor (TMSR-LF1) was taken as the research object, and we proposed a detailed construction process of knowledge graph for fault diagnosis. Fault query and fault real-time monitoring functions were collaboratively integrated into the developed fault diagnosis system. Simultaneously, Bayesian uncertainty inference combined with the variable elimination algorithm was adopted to enable the development of a relatively intelligent and reliable fault diagnosis system. The major contributions of this study are summarized as follows:

- (1) A detailed construction process of knowledge graph for CRDM fault diagnosis is proposed. The unstructured data of CRDM, including design specifications, operation and maintenance manuals, alarm lists, and empirical expert knowledge, were employed as the resource of the fault knowledge graph.
- (2) Fault query and fault real-time monitoring functions were collaboratively integrated into the developed fault diagnosis system to assist the operator to make timely and effective decisions when an exception occurs.
- (3) Bayesian inference method combined with the variable elimination algorithm was adopted to enable the development of a relatively intelligent and reliable fault diagnosis system.
- (4) A fault diagnosis Web interface with graph data visualization was created, making the process of CRDM fault diagnosis intuitive and efficient.

The remainder of this study is structured as follows: Sect. 2 mainly introduces the related methods utilized to

build the proposed CRDM fault diagnosis system, including knowledge graph construction and storage, fault query, fault alarm monitoring, Bayesian inference, and knowledge graph visualization. Section 3 introduces the specific implementation process of the proposed methods and discusses the relevant results. The conclusion and recommendations for future studies are detailed in Sect. 4.

2 Methodology

In this section, work related to the fault diagnosis ontology model construction and storage for CRDM is introduced, which is the foundation of the CRDM fault diagnosis system. Then, a method to build the fault alarm monitoring module is proposed. This module can detect the fault alarm codes from the input/output controller (IOC) of the experimental physics and industrial control system (EPICS) automatically. To facilitate querying fault knowledge quickly and accurately, a set of methods that can extract entities and recognize the query intent simultaneously are introduced. Finally, a Web interface integrated with knowledge graph visualization is constructed based on the Django web framework. The overall architecture of the fault diagnosis system is shown in Fig. 1.

2.1 Knowledge graph construction

The construction method of knowledge graphs can be grouped into two clusters: top-down and bottom-up [17]. Among these, the mode of bottom-top is to extract knowledge from the data source first and then add the obtained entities and relationships to the knowledge base after it has been reviewed. The mode of top-bottom involves building the conceptual model first before extracting related entities and relationships according to the conceptual model. Ontologies are formal conceptual models, which consist of the knowledge representation and provide the definitions of the concepts and relationships associated with unstructured data [18]. A knowledge base based on an ontology model runs on the directed graph and can answer complex queries with reasoning. The format of fault events concerning CRDM in the fault knowledge corpus is fixed and requires a complete ontology model to update the corpus; therefore, using the top-down mode would be more convenient than the bottom-up model when constructing the knowledge graph.

2.1.1 Ontology construction

The purpose of ontology model construction is to conceptualize domain knowledge explicitly. Prior to building a knowledge graph for a specific domain, the knowledge concepts and the relationships between concepts need to

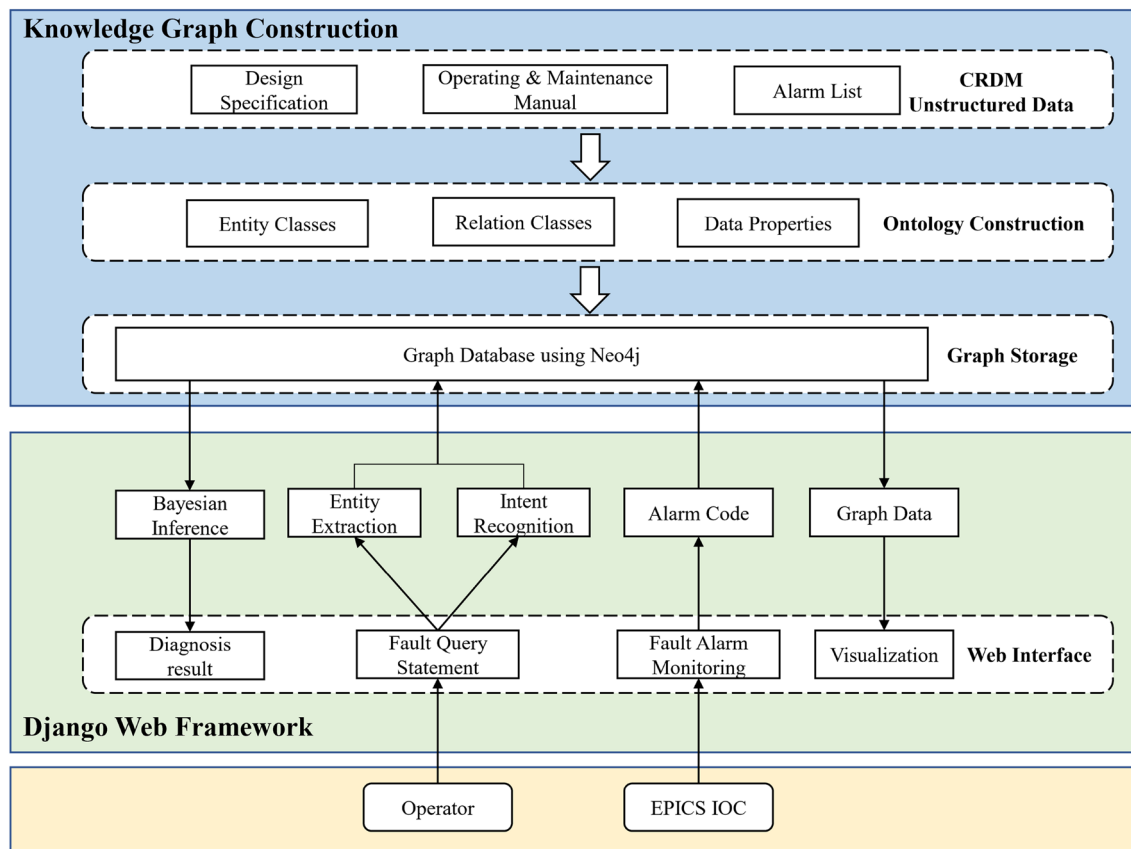


Fig. 1 Overall architecture of the CRDM fault diagnosis system based on knowledge graph

be defined in the ontology model, providing specifications for the extraction of subsequent entities and relationships. At present, the seven-step approach proposed by Stanford University is widely used to construct ontology, and it is more mature and effective to construct ontology with the protégé tool [19, 20]. In this study, we simplified the seven-step approach into four steps. The first step was to build terminology related to CRDM. The second step was to determine the classes and hierarchies of terms. The third step was the construction of the properties and relationships of the classes, the properties included object and data properties. The last step was to determine the individuals by class.

In general, the major task of constructing ontology for fault diagnosis was to define the triples, which is composed of three elements, namely the subject (S), relation (R), and object (O). The triples constitute a fault event, which is mainly derived from the CRDM fault knowledge corpus. Formally, e is used to represent a fault event, and the triples can be represented by formula 1.

$$e = [S] - [r : R] \rightarrow [O] \quad (1)$$

Here, S and O represent the subject and objects in the description text of a fault event, respectively. S and O constitute the entities of the CRDM fault diagnosis knowledge graph. R is the relationship between S and O , which corresponds to the event trigger word in the fault event text.

The ontology of the CRDM fault diagnosis knowledge graph can be divided into four conceptual classes from top to bottom, including *equipment*, *components*, *fault_alarm*, and *fault_symptom*. Among them, the *fault_alarm* represents the fault alarm codes when a fault occurs, and the *fault_symptom* represents the symptom of a fault that appears on a sensor. In the fault knowledge corpus of CRDM, each fault event is represented by unstructured text and the fault events corresponding to conceptual classes need to be decomposed based on the subject–relation–object principle. The decomposed individuals will become the entities stored in the ontology model. Then, the relationships between each conceptual class need to be defined.

This study defines three types of relationships between conceptual classes, including *consist_of*, *has_fault*, *fault_cause*, and *has_symptom*. Further, *consist_of* represents the relationship between *equipment* and *component*, *has_fault* represents the relationship between *equipment* and *fault_alarm*, *has_symptom* represents the relationship between *component* and *fault_symptom*, while *fault_cause* represents the relationship between *fault_symptom* and *fault_alarm*. Some entity individuals have data properties, e.g., the property in each individual from the class of *fault_alarm* has three categories, including *description*, *solution*, and *alarm_type*. The *description* property mainly represents the content of the alarm signals, the *solution* property represents the measures to deal with the alarm signal, and the *alarm_type* property mainly gives a warning to the occurred fault.

After defining conceptual classes and relationships, the fault diagnosis ontology model is constructed by the protégé tool. Figure 2 shows the structure of the fault diagnosis ontology model. The display boxes of dot labeling in the fault diagnosis ontology model represent the conceptual classes, while those of diamond marks indicate individual present in each class. The relationships between class and individuals are represented by solid lines with arrows. Furthermore, the relationships between individuals are represented by dotted lines with arrows, and different relationships are distinguished in various colors. In Fig. 2, the No. 3 control rod has a fault alarm code, namely 10JDEGT202XM25. The *fault_cause* of this fault alarm code is the R6H20, which is the *fault_symptom* of the sensor and is named as resolver_6. The resolver_6 is the component of the No. 3 control rod. Thus, the ontology model has clear classifications and affiliation relationships. The specific configuration of the CRDM fault diagnosis ontology model is shown in Table 1.

Table 1 Configuration of the CRDM fault diagnosis ontology model

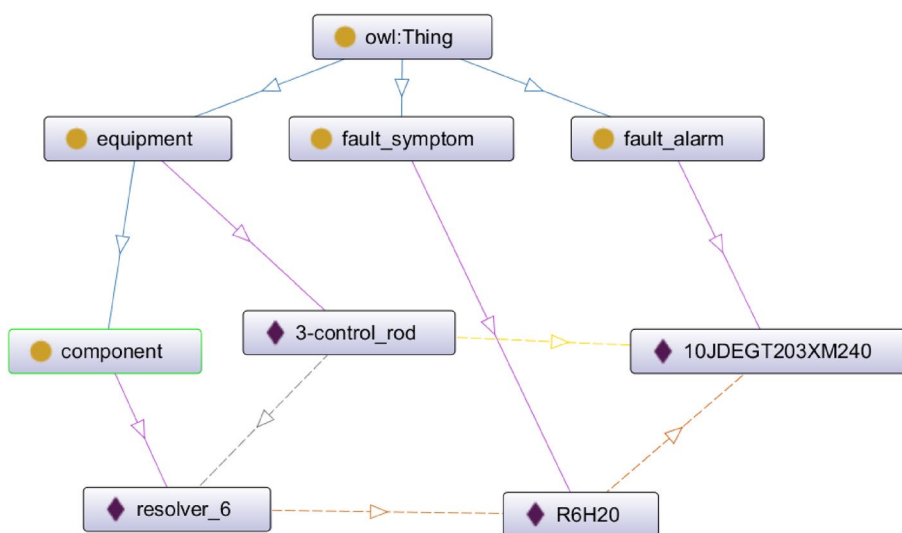
Conceptual classes	Relationships	Data properties
equipment	consist_of	description
component	has_fault	solution
fault_alarm	has_symptom	alarm_type
fault_symptom	fault_cause	

2.1.2 Graph storage

The above-mentioned fault diagnosis ontology model produced by the protégé development platform needs to be stored in a specific physical structure. Neo4j is one of the most popular graph database software, which consists of three elements: node, relationship, and attributes [12]. These elements correspond to the configuration of the constructed CRDM fault diagnosis ontology model. Thus, this study used the Neo4j graph database to store the CRDM fault diagnosis ontology model, which can represent the ontology graph structure clearly. The nodes represent the conceptual classes, and the relationships in Neo4j correspond to the relationships in the ontology model. The data properties of the ontology model will be stored in the attributes from the Neo4j database. Notably, the ontology model cannot be stored in the Neo4j graph database directly. However, the ontology model in the protégé development platform only supports the web ontology language (OWL) format, which conforms to the resource description framework (RDF) syntax. Therefore, the OWL file needs to be converted into an RDF file using the *rdf2rdf* jar package, which can be imported into the Neo4j graph database. Figure 3 shows parts of the knowledge graph stored in the Neo4j database.

Figure 3 shows the nodes corresponding to different entities that are displayed in various colors, and the relationship

Fig. 2 Structure diagram of the CRDM fault diagnosis knowledge graph ontology model



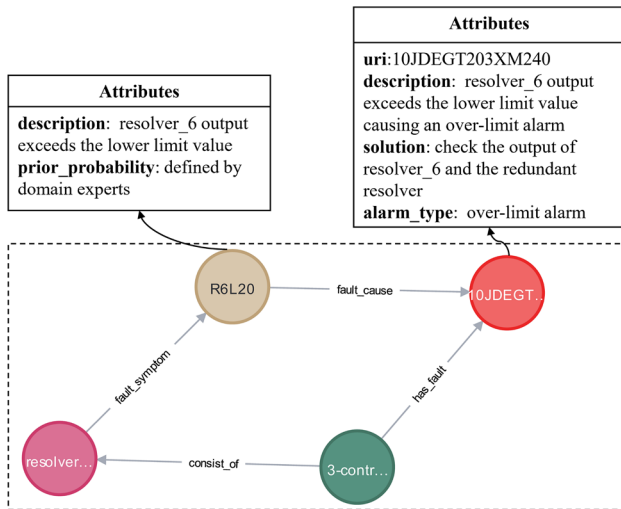


Fig. 3 (Color online) Display of the CRDM fault diagnosis knowledge graph in Neo4j database (partial)

connects the head entity and tail entity. In this figure, the No. 3 control rod used a certain fault event for nuclear fuel compensation, with a fault alarm code named 10JDEGT203XM240, and the fault cause was R6L20, which is related to the No. 6 resolver. The No. 3 control rod consists of No. 6 resolver. Moreover, compared to the conventional SDG model, the fault knowledge graph can directly store more fault information, which covers the prior probability of fault symptom, fault description, and solution. With the help of the Neo4j graph database, detailed information related to the fault alarm can be queried in the Neo4j database using attribute elements handily.

2.2 Fault query

One of the potential of the CRDM fault diagnosis system is to deal with the input query statement and respond to the query content based on the Neo4j graph database. In this process, the fault-related nodes and relationships can be deterministically queried using the constructed fault knowledge graph. This is beneficial for isolating the occurred fault, narrowing the scope of the troubleshooting, and eventually improving the efficiency of fault diagnosis based on Bayesian inference.

Notably, the operator can use the criterion query language format named Cypher to set up a query in the Neo4j database when an exception occurs. Cypher is a friendly declarative, comprehensible, and high-efficiency language in querying node-relationship. The common query format is given as follows:

```
MATCH (n: entity1) - [r: relationship] -> [p: entity2]
WHERE n.uri = an individual in entity1
```

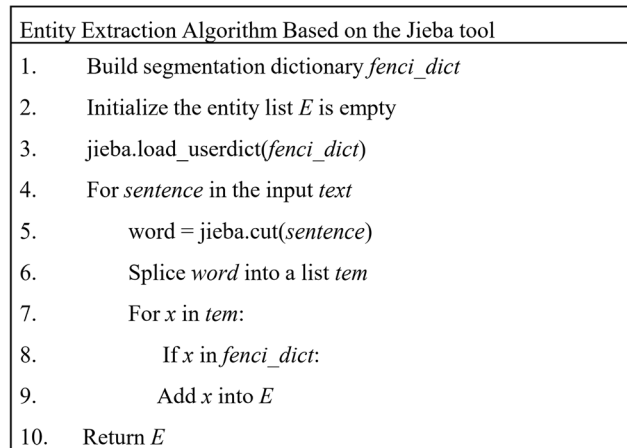


Fig. 4 Detailed algorithm process of entity extraction

RETURN n,r,q

Consequently, in this study, we have introduced a set of methods that can extract the entity and recognize the query intent for fault querying.

2.2.1 Entity extraction

The purpose of entity extraction is to identify the subject of the query statement entered by the operator. The subject (i.e., entity) is the major component of executing the Neo4j graph database query paradigm, which has been stored in the fault knowledge graph. In this study, the Jieba tool with the precise mode was adopted to segment the fault query sentence for extracting the entity. It is one of the most popular Chinese word segmentation modules developed in Python. The fault diagnosis of CRDM is mainly relevant at the equipment level, which leads to a finite number of terms. Therefore, a domain terms dictionary corresponding to CRDM was built to ensure that terminology can be segmented accurately. Then, each segmented word was queried individually in the dictionary. If matching words with the same meaning could be retrieved in the dictionary, the word was extracted as the recognized entity from the fault query statement. In this way, the entities related to the terms of CRDM can be accurately identified and extracted from the fault query statement. Figure 4 reveals the detailed algorithm process of entity extraction.

2.2.2 Intention recognition

Similarly, the query intent is associated with the relationship between entities. The aim of intent recognition in this study was to classify the query intent from the operator, who uses the CRDM fault diagnosis system to query the information about the fault event and make decisions. For

example: the operator wants to find out the possible fault cause of a fault alarm code *10JDEGT203XM240*. The input sentence contains the entity named *10JDEGT203XM240*, and the fault cause is the intent of the query. First, the fault diagnosis system needs to recognize this intent and extract the entity. Then, the fault diagnosis system uses the extracted entity and the classified intent to search for the answers in the constructed knowledge graph database. During this process, natural language understanding is critical to the performance of the CRDM fault diagnosis system. In Chinese, there are different expressions of query intention, but they all have the same semantics, e.g., *What are the causes of the fault?* or *What is the problem causing the fault?* These two queries are related to fault cause. Hence, inspired by previous research [21], a method integrating a three-layer robustly optimized bidirectional encoder representation from transformers (RBT3) pre-training approach and text convolutional neural network (TextCNN) was introduced to address intent recognition. Figure 5 illustrates a high-level view of the proposed RBT3-TextCNN model.

The model architecture of BERT is a multilayer bidirectional transformer encoder based on the original transformer mode and released in the *tensorflow* library [22]. The BERT model provides a powerful context-dependent sentence representation and can be used for intent classification, which is a various target task [21]. Particularly, RBT3 is a small and effective pre-trained model derived

from BERT [24]. Additionally, the TextCNN is a variation model of the CNN, which has a better performance in terms of Chinese text classification [23]. In this study, the RBT3-TextCNN model was adopted to achieve high classification accuracy and low computational cost. The RBT3-TextCNN model consists of a word embedding layer, convolutional layer, global max-pooling layer, and fully connected layer, as shown in Fig. 5.

For a single-sentence classification task based on the BERT model, a special classification embedding ([CLS]) is inserted as the first token and a special token ([SEP]) is added as the last token. Hence, the input sentence with n words is constructed as $X = ([CLS], x_1, x_2, \dots, x_n, [SEP])$. A word x_i can be transformed into a word vector v_i , the formula is as follows:

$$v_i = \text{RBT3}(x_i), i = 1, 2, \dots, n \tag{2}$$

$$V = [v_1, v_2, \dots, v_n] \tag{3}$$

Consequently, the RBT3 model generates a word vector matrix $V \in R^{n \times d}$, where n is the input length and d is the word vector dimension.

Subsequently, the feature map is generated by the convolutional calculation in the TextCNN layer. The convolution kernel needs to be set as $w \in R^{h \times d}$, where h is the height and d is the width. In this study, three types of convolution

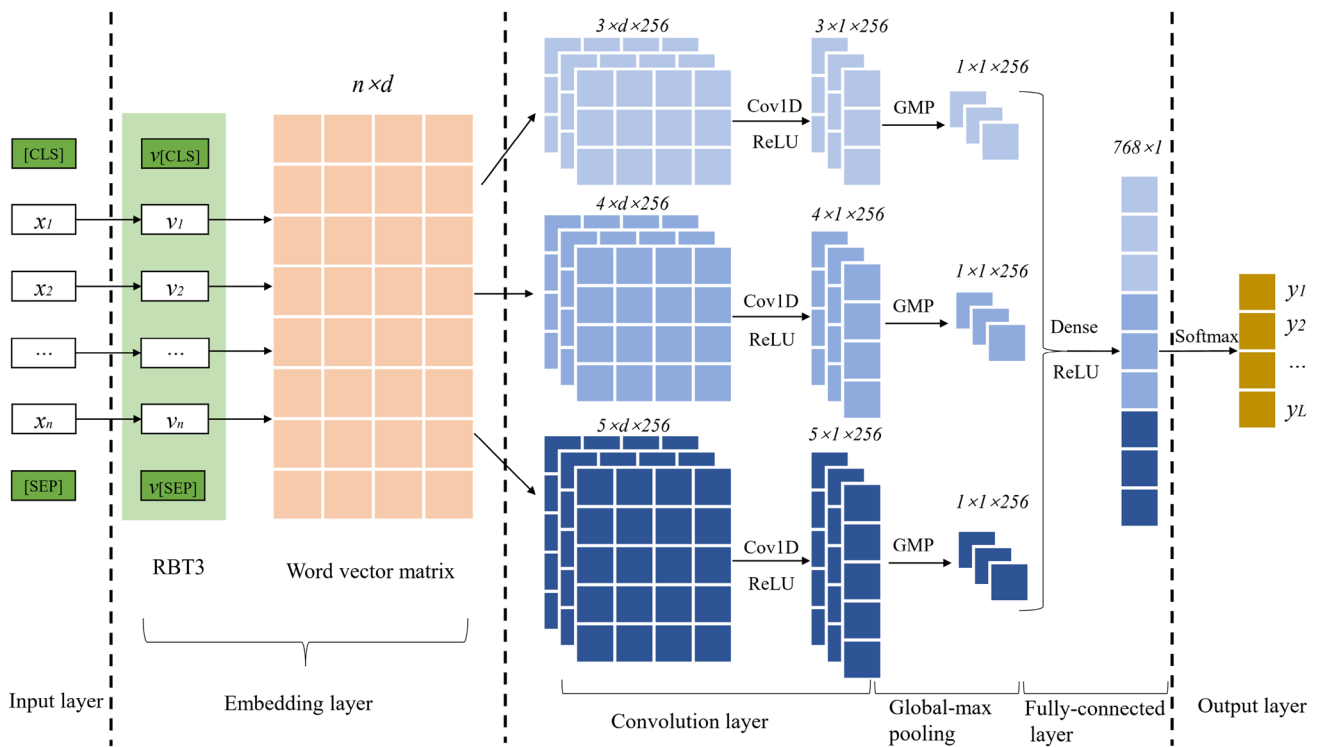


Fig. 5 (Color online) Overall structure of the RBT3-TextCNN model

kernels were set, including $w_1 : (h_1 = 3, d)$, $w_2 : (h_2 = 4, d)$, and $w_3 : (h_3 = 5, d)$. Among these, the input and output channels of the convolution kernel were 1 and 256, respectively. The feature c_{ij} generated by the word vector in the window $v_{i:i+h-1}$, $h = [h_1, h_2, h_3]$ is given by the following formulas:

$$c_{ij} = \sum_{j=1}^3 \text{ReLU}(w_j \cdot v_{i:i+h_j-1} + b), i = 1, 2, \dots, n \quad (4)$$

Here, $b \in R$ is a bias term, and ReLU is a nonlinear activation function. The feature $c_j \in R^{n-h_j+1}$ is generated by the convolution operation of the convolution kernel w and the word vector V , as shown in the following formula:

$$c_j = [c_{1j}, c_{2j}, \dots, c_{(n-h_j+1)j}], j = 1, 2, 3 \quad (5)$$

Then, the most important feature information is compressed and retained through the global max-pooling (GMP) operation, which only takes the maximum value of each feature. The new feature z is obtained as follows:

$$c'_j = \text{GMP}\{c_j\} = [c'_{1j}, c'_{2j}, \dots, c'_{kj}], j = 1, 2, 3, \quad k = n - h_j + 1, \quad (6)$$

$$z = \text{concat}(c'_1, c'_2, c'_3) \text{ (axis} = -1), \quad (7)$$

where concat is the splicing operation and k is the total number of kernels. The final output classification result y of the fully connected layer is as follows:

$$y = \text{softmax}(w_{\text{dense}} \cdot (z \odot r) + b_{\text{dense}}), \quad (8)$$

where w_{dense} and b_{dense} are weights and bias terms of the fully connected layer, respectively; $r \in R^h$ is the mask vector used to randomly drop elements out in z ; the symbol \odot represents the element-level multiplication calculation; and softmax denotes the activation function, which can efficiently tackle the problem of multi-categories classification. Meanwhile, the sparse categorical cross-entropy is adopted as the loss function in this model, as shown in the following formula:

$$\text{Loss} = - \sum_{i=1}^T y_i \log p_i \quad (9)$$

Among them, T goes over all the output classes, p_i indicates the probability of the class as predicted by the model, and y_i represents the real probability of the class as provided by the input data.

2.3 Fault alarm monitoring

Numerous fault alarm codes have been set up to detect the state of each crucial component of the CRDM in TMSR-LF1. The related information about these alarm codes of

the occurred fault, including fault location, fault cause, and solution, has been stored in the CRDM fault diagnosis knowledge graph constructed by the above-mentioned method. The operator can use these fault alarm codes to query detailed fault information in the CRDM fault diagnosis system. However, once the fault alarm occurs, the operator needs to grasp the key information of the fault alarm as soon as possible and make a response to the fault quickly and accurately. Therefore, it is necessary to monitor the fault alarm codes at any time and present the information related to the appeared fault alarm to the operator automatically.

In this study, a fault alarm monitoring module was developed based on WebSocket, which is a new protocol provided by the Internet Engineering Task Force. The WebSocket protocol is a web technology that provides an effective and standardized solution for bidirectional, full-duplex communication between a web browser and a web server [25]. Unlike the HTTP protocol, the WebSocket protocol allows bidirectional communication, which implies that the WebSocket server can push data to the client without the user's request. WebSocket is a persistent protocol and responds to messages timely. Hence, once the fault alarm monitoring module in the CRDM fault diagnosis system establishes a connection with the IOC of the EPICS through the WebSocket server, the module can monitor the fault alarm code whenever necessary, greatly reducing the latency and consumption of network bandwidth and hardware resources. The technical architecture diagram of the CRDM fault alarm monitoring module is shown in Fig. 6.

According to Fig. 6, the fault alarm code is set as a value of the process variable (PV) in the IOC of the EPICS. PyEpics is an interface for the channel access (CA) library of the EPICS to the Python programming language. The

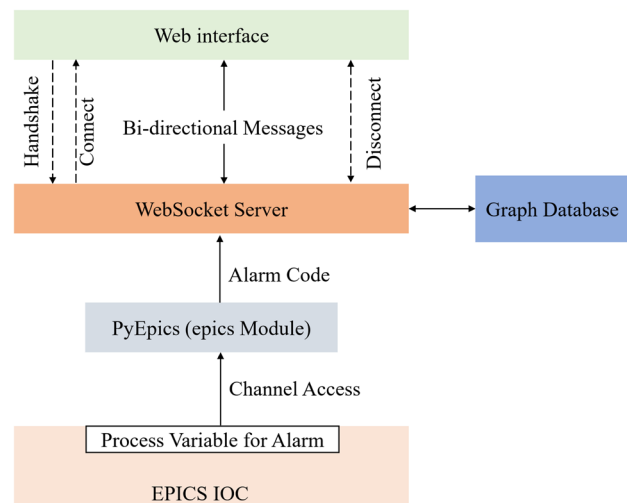


Fig. 6 (Color online) Diagram of the CRDM fault alarm monitoring module

PyEpics provides an *epics* module to Python, with methods for reading from and writing to PV via the CA protocol. The WebSocket server does not rely on the graphical interface and can be deployed in the Python environment. Thus, the alarm code can be obtained by PyEpics and saved to the WebSocket server. In this study, the fault alarm monitoring module was integrated into the Django web framework, which is a high-level Python web framework. With the help of the fault alarm monitoring module, the CRDM fault diagnosis system can monitor the occurrence of the CRDM fault alarm in real-time. When a fault alarm emerges, the system can capture the fault alarm code automatically and display the detailed information of this fault alarm in the Web interface. The operator can make effective decisions quickly using the information provided by the fault diagnosis system.

2.4 Bayesian inference

The limited fault data and knowledge give rise to the uncertainty of diagnostic outputs, reducing the reliability of the fault diagnostic model [26]. Bayesian inference is an accurate prediction method that is especially useful when there is a lack of sufficient fault data but precise inference information is still needed. Bayesian inference has gained prominence in the nuclear field, including in radioactive substance identification [27] and nuclear data evaluation [28], and has a proven availability and reliability. Bayesian networks (BNs), based on strict probabilism, are a type of probabilistic graphical model based on Bayesian inference that effectively describes causality in uncertain problems and realizes inference and prediction. Consequently, it has

attracted extensive attention in the field of intelligent fault diagnosis [29, 30].

The information corresponding to the CRDM fault components, reasons, and symptoms has been stored in the Neo4j graph database. That is, if an exception occurs, the operators can use the Cypher language to quickly figure out the possible fault equipment and the causes to achieve the goal of fault isolation, which drastically reduces the scope of fault diagnosis. For instance, the following query statement can be used to query all the causes of an occurred fault:

```
MATCH (f:fault_symptom)->[r:fault_cause]-(n:reason_node) RETURN f,r,n
```

Therefore, this reasoning can be deemed as certain based on the Neo4j graph database. During the construction of the fault knowledge graph, the nodes and relationships related to the detected fault are clearly defined. Nevertheless, it still often contains unclear and uncertain knowledge in the practical application of a fault diagnosis system that is knowledge base oriented. In this study, a BN for uncertainty reasoning with the variable elimination algorithm was proposed to evaluate the exact likelihood of fault symptom occurrence on the foundation of a deterministic fault isolation exploited knowledge graph and the prior knowledge of fault symptoms. The process of BN for fault diagnosis based on the fault knowledge graph is shown in Fig. 7.

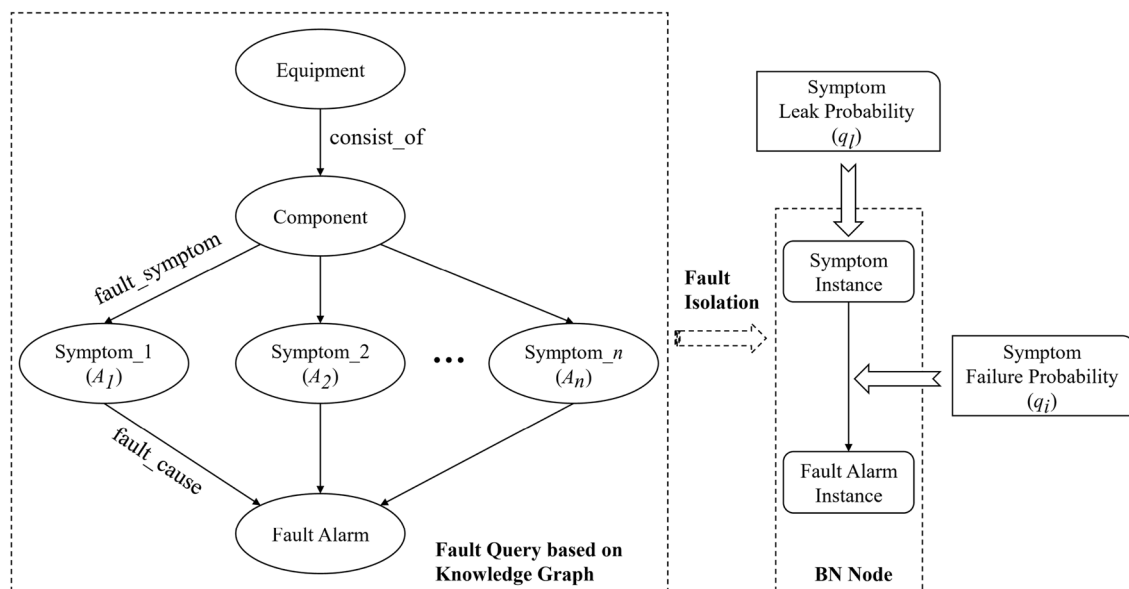


Fig. 7 BN construction for fault diagnosis based on fault isolation using the fault knowledge graph

2.4.1 BNs

BNs are well-established graphical formalisms for encoding conditional probabilistic relationships between uncertain variables [31]. The structure of BNs is a directed acyclic graph (DAG) in which the nodes represent variables, arcs signify the existence of direct causal relationships between the linked variables, and strengths of these relationships are expressed by conditional probabilities. BNs aim to model conditional dependence, and therefore causation, by representing conditional dependence by edges in a DAG. BNs are ideal for taking an event that occurred and predicting the likelihood that any one of several possible known causes was the contributing factor. Bayes' theorem is the foundation for uncertainty inference based on the BNs.

Bayes' theorem is stated mathematically as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{10}$$

where A and B are events and $P(B) \neq 0$.

- (1) $P(A)$ refers to a prior probability without any given conditions.
- (2) $P(B)$ refers to the probability of observing B , known as marginal probability.
- (3) $P(A|B)$ refers to the probability of event A occurring given that B is true, also called the posterior probability of A given B .
- (4) $P(B|A)$ refers to the probability of event B occurring given that A is true.

In the Bayesian context, $P(A|B)$ and $P(B|A)$ are all conditional probabilities. For each node B in BN, $P(A|B)$ is completely described in the CPT. Hence, the prerequisite in building the BN model for decision support and risk assessment is to define the necessary conditional probability tables (CPTs), which rely on the judgment of domain experts. For a node in BN, along with n parents, which are Boolean, the size of the CPTs grows exponentially with n . This is known to be practically complex and intractable. To tackle the parameter explosion problem existing in CPTs, the Leaky Noisy-OR function has proven to be useful in simplifying the elicitation of complex CPTs in BNs involving Boolean variables.

2.4.2 Leaky Noisy-OR function

The Leaky Noisy-OR function is a good approximation of a wide range of BN model fragments and requires only $n + 1$ parameters for the full CPT specification [31]. Formally, the definition of the Leaky Noisy-OR function is as follows:

Let $X = \{X_1, X_2, \dots, X_n\}$ be n Boolean variables, i.e., $X_i = \{T, F\}$, $i = 1, 2, \dots, n$. Let Y be a Boolean variable with parent nodes $\{X_1, X_2, \dots, X_n\}$. Then, the Leaky Noisy-OR function would be defined as follows [32]:

$$P(Y = T|X_p) = \begin{cases} 1 - (1 - q_l) \prod_{X_i \in X_T} q_i & X_T \neq \emptyset \\ q_l & X_T = \emptyset \end{cases}, \tag{11}$$

where X_p represents a set of states for all parent nodes, X_T represents the set of all the parent nodes of T in X_p , q_l represents the leak probability, and q_i represents the failure probability. The corresponding concepts are defined as:

- (1) Leak probability q_l : Due to insufficient analysis, the cause of the fault event is not detected, while the corresponding fault symptom is detected. The probability of such happening is called leak probability.
- (2) Failure probability q_i : In contrast, the failure probability represents the probability that the cause is detected but the fault symptom is not detected because of monitoring means, measurement error, and low frequency of fault. In most cases, there are not enough fault cases, especially in NPP. Estimating the failure probability by expert experience and failure mechanisms is recommended.

2.4.3 Variable elimination

Bayesian inference is a method of statistical inference in which the Bayesian theorem is used to update the posterior probability of a hypothesis as more evidence or information becomes available. More specifically, inference in BNs can diagnose the causes given the fault occurrence under causality. Several exact approaches for computing posterior probabilities in BN have been proposed and implemented, including variable elimination (VE), junction tree propagation, and approximate inference. [33]. Among these, the VE algorithm, adopted in this study for the sake of the efficiency-complexity trade-off, is fundamental and easily understood. It tackles changes to the knowledge base more easily than other approaches. The detailed process of the VE algorithm is shown in Table 2:

Assuming that the BN contains n random variables to be eliminated and the type of each variable is Boolean, each step eliminates a random variable, and there are at most k probability distribution functions associated with each random variable. Then, the time complexity of the VE algorithm is $O(n \times 2^k)$.

2.5 Knowledge graph visualization

Knowledge graph visualization can intuitively display the complex knowledge stored in the knowledge graph and helps the operator understand the equipment relationships of the

Table 2 Process of VE algorithm based on BNGiven $VE(N, x, E, e, \sigma)$:**Input:** N : Bayesian Network x : query variable E : list of observed variableswhere $X = \{x\}$ and E are disjoint subsets of U e : an observed value σ : an elimination ordering for variables $U - X \cup E$ **Output:** $P(x|E = e)$ **Steps:**1. $F = \{\text{all probability distributions in } N\}$, i.e., F is the joint probability distribution of N 2. $E = e$, E is one of the factors in F 3. while ($\sigma \neq \emptyset$):4. remove the first variable Z from σ 5. $F \leftarrow \text{Elim}(F, Z)$

6. end while

7. $h(Q) = \prod_F f$, $f \in F$ 8. return $h(Q) / \sum_Q h(Q)$ **Elim** (F, Z):**Input:** F : a set of functions Z : the variable to be eliminated**Output:** F^* : a new set of functions**Steps:**1. delete all functions ($\{f_1, f_2, \dots, f_k\}$) related to Z from F 2. $g \leftarrow \prod_i^k f_i$ 3. $h \leftarrow \sum_Z g$ 4. put h back into F and get a new set of functions F^* 5. return F^*

structures and fault knowledge of the CRDM. In this study, the node-link method combined with the force-directed layout was utilized to visualize the CRDM knowledge graph in the Web interface. The node-link method can map the entities and relationships saved in the knowledge graph to the nodes and lines of the two-dimensional plane, respectively. The force-directed layout uses the spring model to simulate the interaction between nodes through exclusion and attraction, which can reflect the close relationships between entities and topological attributes in the knowledge graph.

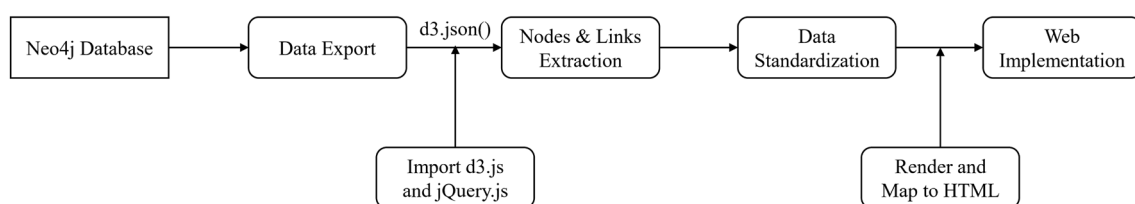
Most existing visual icon libraries support the force-directed layout, including Echarts, D3.js, and Neovis.js. These libraries can be used to visualize graph data stored in the Neo4j graph database. Among these, D3.js is a JavaScript library for manipulating documents based on data. D3.js combines powerful visualization components and strictly adheres to Web standards and does not rely on any framework; both the aesthetic and performance of D3.js out-compete those of others.

The version of *d3.v4.min.js* was adopted in this study. This version not only supports SVG but also the use of Canvas. The flowchart for visualizing graph data in the Neo4j

database is shown in Fig. 8. First, the graph data stored in the Neo4j database were exported as a JSON file, and the entities and relationships constitute a dictionary data format. Then, the entities and relationships were extracted and saved as nodes and links, respectively. Notably, the obtained nodes needed to be cleaned to remove duplicate individuals. Finally, the extracted nodes and links were rendered by the D3.js library and mapped to HTML, which was implemented in the Web interface.

3 Results and discussion

In this section, the experiment results related to the RBT3-TextCNN model proposed for fault query are introduced. The performance of the RBT3-TextCNN is verified by the dataset generated by the actual fault knowledge corpus of CRDM in TMSR-LF1. Then, a case study about the Bayesian inference based on the constructed CRDM fault knowledge graph is introduced in terms of fault diagnosis. Next, the practical implementation of the fault alarm monitoring module based on the WebSocket protocol and the Web

**Fig. 8** Flowchart of visualizing data stored in the Neo4j graph database

interface for the CRDM fault diagnosis system are described in detail.

3.1 Intent recognition experiment

3.1.1 Dataset

This study collected 753 fault query description texts from the CRDM in TMSR-LF1, and the corpus of the constructed knowledge graph is mainly derived from the design specification, operation and maintenance manual, and alarm list. The short sentences for training the RBT3-TextCNN model were categorized into four classes, namely fault cause, fault phenomenon, fault solution, and equipment composition relationship. The average length of each sentence was 15.7 words and the number of total characters was approximately 11,800. Each sentence in the corpus was manually labeled. The dataset was divided into the training set and the test set at a ratio of 7:3. The information about fault query text distribution is shown in Table 3.

3.1.2 Training details

The configuration parameters of the RBT3 model were quoted from the open-source website¹. RBT3 has three hidden layers and was derived from the BERT-Base model, which has 768 hidden states. Therefore, the input word vector–matrix dimension was set to 768. The optimizer and the maximum length (MaxLen, 50 tokens) were cited from the literature [21]. The TextCNN filter size, activation function, and pooling method were extracted from the literature [23]. The total number of TextCNN was equal to the

input word vector–matrix dimension. Besides, another three hyper-parameters, including batch size, epochs, and learning rate, were generated via fine-tuning based on the comparison experiment. As shown in Fig. 9, the RBT3-TextCNN model training stopped before 20 epochs. For conservative consideration, the initial training epoch was set to 25. The batch size was set to 8 for the sake of validation loss and validation accuracy. The learning rate was automatically adjusted according to the training accuracy. Adam was used for optimization with the initial learning rate value of 5e-5 [21]. The training details are shown in Table 4. The operating environment was set as an 8-core Intel Core i7-7740 K CPU@4.30 GHz and the GPU was Geforce GTX 1080Ti. Moreover, the major dependency environments were tensorflow-gpu-1.14.0 and keras-2.3.1.

3.1.3 Results

To test the validity of the model with respect to the intention recognition using the above-mentioned dataset, we selected the TextCNN and RBT3 as the baseline models to compare

Table 3 Statistics of the dataset used in this study

Category	Total
Fault cause	255
Fault solution	204
Fault phenomenon	169
Equipment relationship	125

Table 4 Parameters of the RBT3-TextCNN model

Parameter names	Value
RBT3 hidden layers	3
Word vector–matrix dimension	768
MaxLen	50
Batch size	8
TextCNN filter size	3,4,5
Total number of TextCNN filter	256×3
Activation function of TextCNN	ReLU
Pooling layer	GMP
Optimizer	Adam
Learning rate	5×10 ⁻⁵
Epochs	25
Classification number	4

¹Resource is available: <https://github.com/ymcui/Chinese-BERT-wwm>

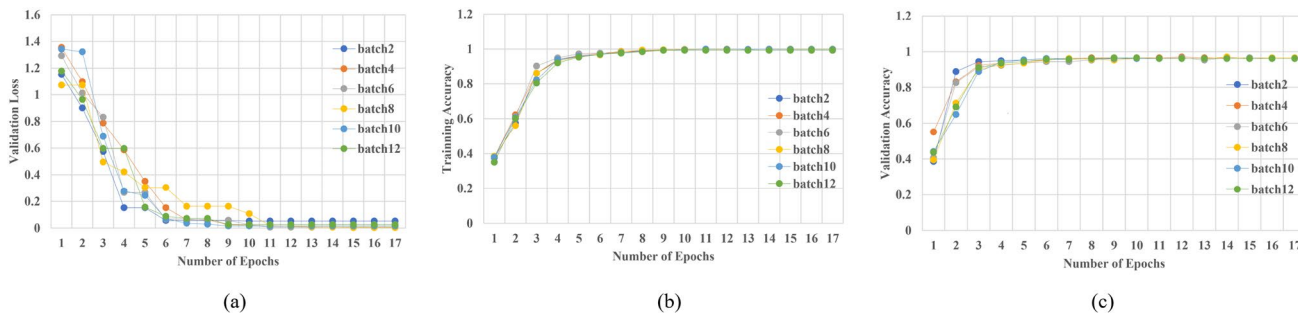


Fig. 9 (Color online) Performance of batch size with different values. **a** Validation Loss. **b** Training accuracy. **c** Validation accuracy

with RBT3-TextCNN. The experiment involved the use of two indicators to evaluate the model performance. Among them, the macro average composite index (MaF1) is the weighted average of macro accuracy (MaP) and macro average recall rate (MaR). Considering the uneven distribution of various samples under multiple classification issues, the accuracy of different models was evaluated based on the MaF1 indicator, which is widely adopted to evaluate the performance of classification models in machine learning. Assuming the dataset can be divided as (C_1, C_2, \dots, C_n) , the MaF1 formula is as follows:

$$\text{MaF1} = \frac{2 \times \text{MaP} \times \text{MaR}}{\text{MaP} + \text{MaR}}, \tag{12}$$

$$\text{MaP} = \frac{1}{n} \sum_{i=1}^n P_i, \tag{13}$$

$$\text{MaR} = \frac{1}{n} \sum_{i=1}^n R_i, \tag{14}$$

$$P_i = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{15}$$

$$R_i = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{16}$$

where n is the number of categories in the dataset, TP represents the number of samples correctly predicted in C_i , TP + FP represents the number of samples predicted in C_i , and TP + FN represents the number of real samples in C_i . Hence, P_i (i.e., Precision) reflects the proportion of true positive samples among the positive samples by the classifier, and R_i (i.e., Recall) reflects the proportion of correctly predicted positive samples in the total positive samples. MaF1 combines the effects of P_i and R_i . In addition, the total training time of different models was calculated to evaluate the efficiency.

RBT3 is a powerful model for extracting the characteristics of input statements, especially under the circumstance of limited training datasets. RBT3 has many parameters, and the training time complexity is higher than that of the other models. However, TextCNN is proposed as a relatively fast-training model, but its performance is subject to limited training datasets. In this study, TextCNN and RBT3 were chosen as the baseline models, respectively. The small dataset utilized in this study was derived from the CRDM fault corpus of the TMSR-LF1. The results are shown in Table 5.

Table 5 Comparison of different models based on the CRDM intention recognition dataset

Model	MaF1	Total training time (s)
TextCNN	55.75%	70.35
RBT3	96.03%	51.61
RBT3-TextCNN	97.13%	56.92

The TextCNN model took more training time to achieve convergence than the RBT3 model based on the same batch size. The RBT3 model showed a better performance in recognizing the CRDM-related fault query intent than the TextCNN model. Compared with RBT3 and TextCNN, the RBT3-TextCNN had a 1.1% and 2.36% advantage in intent recognition performance, respectively, but the total training time gap among these three models was not obvious.

3.2 Case study using Bayesian inference

Taking the fault event D “The deviation of No. 3 control rod position in TMSR-LF1 exceeds the technique setting value” as an example to illustrate the inference based on BN, the reasoning result consistently was the posterior probability of causes corresponding to the fault. As shown in Fig. 10, A_1, A_2, A_3 refer to high-frequency electromagnetic interference, sensor improper installation, and sensor breakdown respectively, which are independent of each other. There is an OR relationship between D and A_1, A_2, A_3 , which are the fault symptoms of D . In this study, domain experts directly endow the prior probability $(P(A_1), P(A_2), P(A_3))$ of the fault symptoms, leak probability q_l , and failure probability q_f . Accordingly, the conditional probabilities $(P(D|A_1, A_2, A_3))$ in CPT were calculated by the Leaky Noisy-OR function, and the results are shown in Fig. 10.

Given the observed evidence is $P(D = T)$, the calculation steps of posterior probability $(P(A_1|D = T))$ are shown below:

To begin, $P(A_1|D = T)$ is defined by the Bayesian theorem:

$$P(A_1|D = T) = \frac{P(A_1, D = T)}{P(D = T)} \tag{17}$$

Then, assuming that the elimination order is $\{A_1, A_2, A_3\}$, the joint probability distribution can be decomposed based on the VE algorithm as follows:

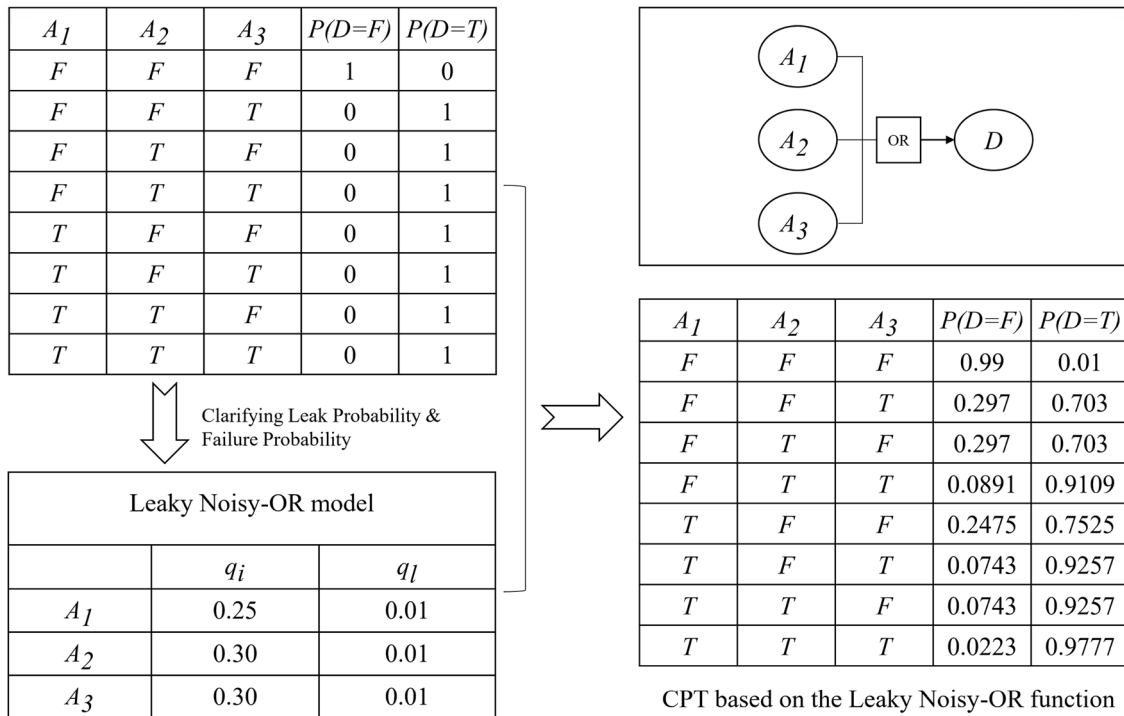


Fig. 10 Result of the CPT based on the Leaky Noisy-OR function

$$\begin{aligned}
 P(A_1, D = T) &= \sum_{A_2, A_3} P(D = T, A_1, A_2, A_3) \\
 &= \sum_{A_2} \left(\sum_{A_3} P(D = T | A_1, A_2, A_3) \cdot P(A_1, A_2, A_3) \right) \\
 &= \sum_{A_2} \left(\sum_{A_3} P(D = T | A_1, A_2, A_3) \cdot P(A_1) \cdot P(A_2) \cdot P(A_3) \right) \\
 &= \sum_{A_2} P(A_1) \cdot P(A_2) \left(\sum_{A_3} P(A_3) \cdot P(D = T | A_1, A_2, A_3) \right) \\
 &= P(A_1) \cdot \sum_{A_2} P(A_2) \cdot m(A_1, A_2) \\
 &= P(A_1) \cdot m(A_1)
 \end{aligned} \tag{18}$$

where

$$\begin{aligned}
 m(A_1, A_2) &= \left(\sum_{A_3} P(A_3) \cdot P(D = T | A_1, A_2, A_3) \right) \\
 m(A_1) &= \sum_{A_2} P(A_2) \cdot m(A_1, A_2)
 \end{aligned} \tag{19}$$

Thus, three variables $\{A_1, A_2, A_3\}$ are reduced to one target variable $\{A_1\}$.

Besides, the marginal probability $P(D = T)$ is defined as:

$$\begin{aligned}
 P(D = T) &= \sum_{A_1} P(A_1, D = T) \\
 &= \sum_{A_1} P(A_1) \cdot m(A_1)
 \end{aligned} \tag{20}$$

Therefore,

$$P(A_1 | D = T) = \frac{P(A_1) \cdot m(A_1)}{\sum_{A_1} P(A_1) \cdot m(A_1)} \tag{21}$$

Indeed, the aforementioned posterior probability calculation can be implemented by a Python library, named Pgmgy, which is a pure Python implementation for Bayesian Networks with a focus on modularity and extensibility. In this study, the Pgmgy Python library was adopted and integrated with the VE algorithm to realize real-time uncertainty reasoning.

This case illustrates that the Bayesian diagnostic network using the Leaky Noisy-OR function can calculate the conditional probability of exception accurately in reality, according to the prior probabilities corresponding to fault symptoms. In addition, the fault propagation path can be extracted explicitly with the help of the constructed knowledge graph. Thus, the nodes associated with the fault were isolated to facilitate the usage of the VE algorithm to update the posterior probability of being reliable, which can give sequential suggestions for further troubleshooting the fault symptoms.

Table 6 Result of fault diagnosis based on Bayesian inference

Fault symptom	Description	Prior probability	Diagnosis result
A_1	High-frequency electromagnetic interference	0.20	$P(A_1 = T D = T) = 0.8193$
A_2	Sensor improper installation	0.15	$P(A_2 = T D = T) = 0.8642$
A_3	Sensor breakdown	0.11	$P(A_3 = T D = T) = 0.9017$

The diagnosis results based on Bayesian inference are presented in Table 6. In the absence of additional evidence, the probabilities of the occurrence of the direct fault symptoms of case fault D were updated. The troubleshooting sequence of the cause of the failure was as follows: $A_3 \rightarrow A_2 \rightarrow A_1$. This can provide an effective troubleshooting scheme for the operator.

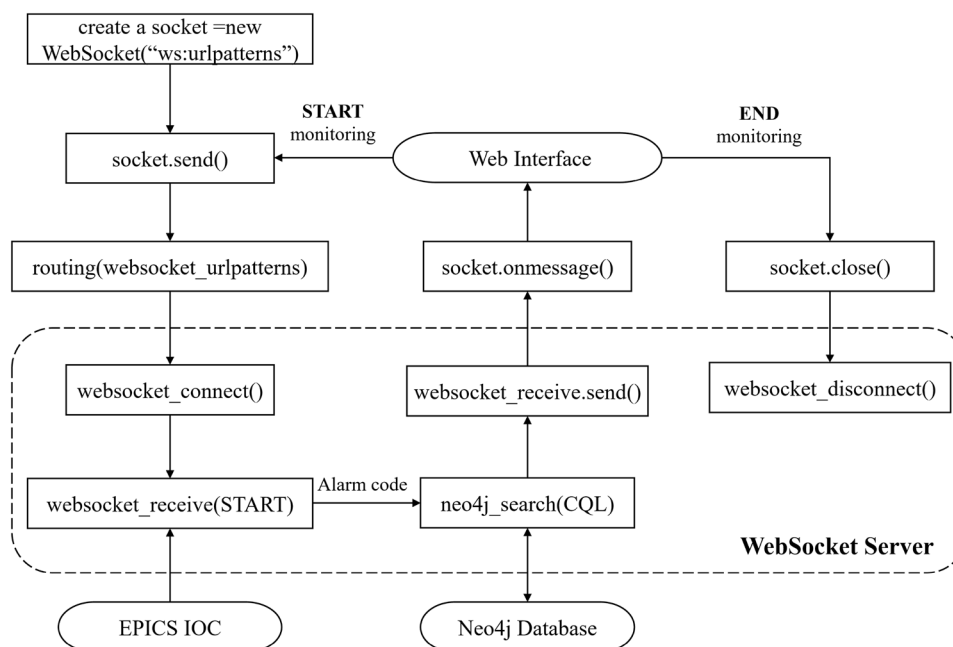
3.3 Fault alarm monitoring based on WebSocket

The advantage of the CRDM fault alarm monitoring module based on WebSocket is that it can monitor the alarm code from the IOC in EPICS through the Web interface quickly. The CRDM fault diagnosis system can give a specific description and the solution corresponding to the detected fault alarm, which can help the operator to grasp the equipment running status efficiently. In this study, the operating platform of the CRDM fault diagnosis system was built by the Django web framework, which does not support WebSocket Protocol directly; thus, the first step was to install the module named Channels. The Channels were built around a basic low-level spec called asynchronous server gateway interface (ASGI), and it can extend the abilities of Django to handle WebSockets, chat protocol, and IoT protocols. Then,

both the creation of a socket for using the WebSocket protocol and a uniform resource locator (URL) beginning with the *ws* label were required, respectively. The URL was used to execute the WebSocket protocol rather than the HTTP request processing; therefore, it needs to be defined in the ASGI application as the WebSocket URL pattern. The Channels can be shipped with *generic consumers* that consolidate common functionality, especially for WebSocket handling. The *WebSocketConsumer* was adopted to construct the WebSocket server.

The detailed implementation of this module is shown in Fig. 11. This socket is responsible for communicating with the WebSocket Server, accepting information from the WebSocket server, and sending operation requests from the Web interface. The basic handlers of the WebSocket server in this study include *websocket_connect*, *websocket_receive*, *websocket_disconnect*, and *neo4j_search*. When the connection between the WebSocket server and the Web interface is successful, the WebSocket server will respond to the START and END monitoring requests from the Web interface. The *neo4j_search* uses the captured fault alarm code to match the related prior knowledge stored in the Neo4j graph database.

Fig. 11 Detailed implementation process of the CRDM fault alarm monitoring module



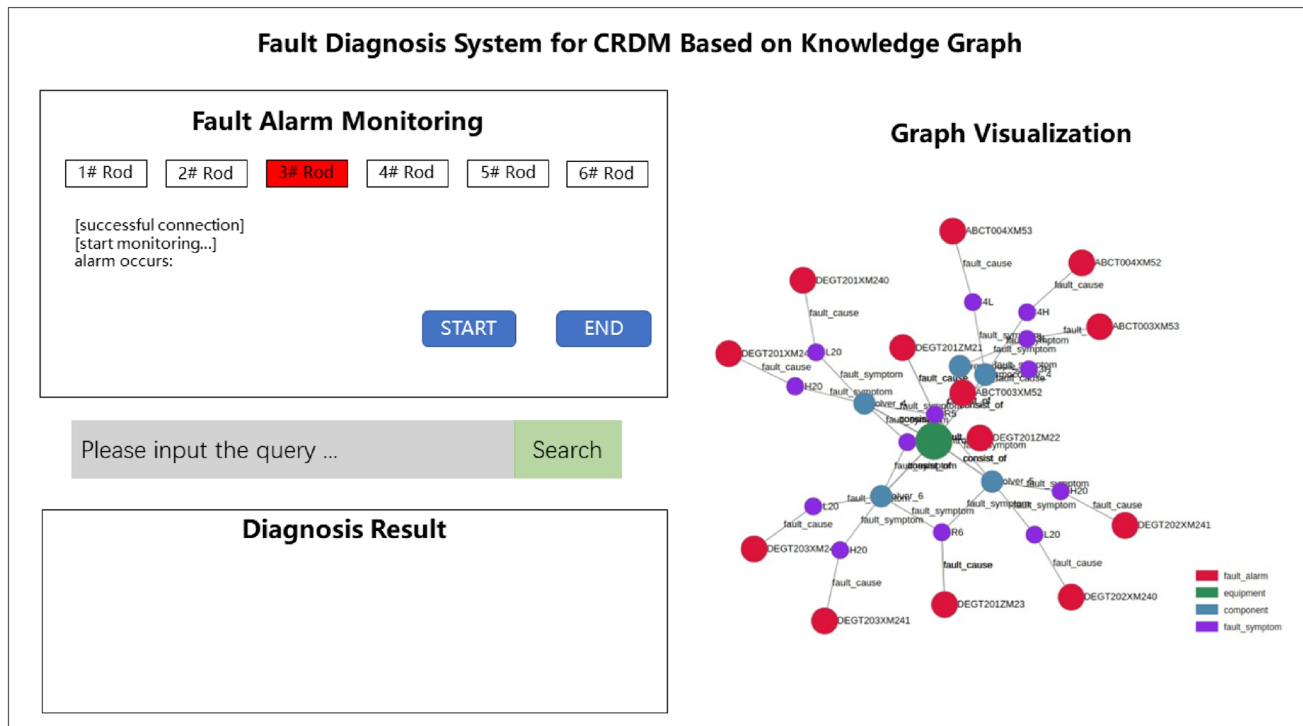


Fig. 12 (Color online) Initial implementation of the Web interface in the CRDM fault diagnosis system

3.4 Web interface implementation

The Web interface, designed to prompt the process of CRDM fault diagnosis intuitively and effectively, was integrated with fault alarm monitoring, querying fault event information, and visualization of the knowledge graph. Figure 12 shows the preliminary implementation of the Web interface in the CRDM fault diagnosis system. The operator can open and close the CRDM fault alarm monitoring at any time through the Web interface. The fault diagnosis system supports the function of natural language input for querying the knowledge about the CRDM fault event knowledge. Besides, the Web interface can display the knowledge graph corresponding to the CRDM directly. It is convenient for the operator to view the information related to the fault event in the diagram, including the entities and relationships.

4 Conclusion and future work

The construction of the CRDM fault diagnosis system using the knowledge graph is an effective technology for better utilizing the CRDM unstructured data generated during the processes of CRDM design, manufacturing, and functional testing. This prototype system, combined with fault

knowledge query, real-time fault alarm monitoring, and fault diagnosis based on Bayesian inference, is complete and easy to deploy in practice. In general, this work can be regarded as a pilot scheme for fault diagnosis based on the knowledge graph in the field of NPPs.

Although the prototype of the CRDM fault diagnosis system based on the knowledge graph has initially been constructed, there are still limitations. At present, the established knowledge graph is still in its infancy and only located in the key device layer. The stored fault information is subject to the size of the constructed fault knowledge graph. In the next step, the fault knowledge corpus of the CRDM needs to be expanded, and fine granularity fault knowledge and device mechanism knowledge need to be updated. In addition, the prior probabilities of fault symptoms mainly rely on the expertise of the domain expert. Its accuracy still has limitations as the actual transformation of operation conditions. Thereby, future work should focus on making a comprehensive test and analysis method in practice and give a specific evaluation criterion to optimize the substantial probabilities of fault symptoms occurrence.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection, methodology, experiment, and analysis were performed by Xue-Jun Jiang. Method improvement and critical commentary were performed by Wen Zhou. Supervision and project administration were performed by Jie Hou.

The first draft of the manuscript was written by Xue-Jun Jiang and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

1. L. Zhang, Q. Li, J. Luo et al., A novel method for diagnosis and de-noising of control rod drive mechanism within floating nuclear reactor. *Ocean Eng.* **244**, 110398 (2022). <https://doi.org/10.1016/j.oceaneng.2021.110398>
2. Y. Wilhelm, P. Reimann, W. Gauchel et al., Overview on hybrid approaches to fault detection and diagnosis: combining data-driven, physics-based and knowledge-based models. *Procedia CIRP* **99**, 278–283 (2021). <https://doi.org/10.1016/j.procir.2021.03.041>
3. W. Li, H. Li, S. Gu et al., Process fault diagnosis with model-and knowledge-based approaches: advances and opportunities. *Control Eng. Pract.* **105**, 104637 (2020). <https://doi.org/10.1016/j.conengprac.2020.104637>
4. H. Qian, C.C. Ma, Research on belief rule base of intelligent diagnosis expert system based on event-triggering mechanism in nuclear power plant. *Atomic Energy Sci. Technol.* **51**(3), 485 (2017). <https://doi.org/10.7538/yzk.2017.51.03.0485>. (in Chinese)
5. W.L. Wang, M. Yang, P.H. Seong, Development of a rule-based diagnostic platform on an object-oriented expert system shell. *Ann. Nucl. Energy* **88**, 252–264 (2016). <https://doi.org/10.1016/j.anucene.2015.11.008>
6. Y.K. Liu, X. Ai, A. Ayodeji et al., Enhanced graph-based fault diagnostic system for nuclear power plants. *Nucl. Sci. Tech.* **30**, 174 (2019). <https://doi.org/10.1007/s41365-019-0708-x>
7. Y. Zhao, Q. Zhang, C.L. Dong, Design and implementation of nuclear power plant fault diagnosis expert system based on DUCG. *Atom. Energy Sci. Technol.* **50**(8), 1474 (2016). <https://doi.org/10.7538/yzk.2016.50.08.1474> (in Chinese)
8. X. Ma, J. Wu, D.Z. Li, Fault diagnosis of nuclear power plant based on simplified signed directed graph with principal component analysis and support vector machine. in *2019 Chinese Automation Congress (CAC)*. 3082–3087 (2019). <https://doi.org/10.1109/CAC48633.2019.8997001>
9. L.L. Shi, H.G. Jia, Z.W. Zhou et al., Intelligent maintenance design of nuclear power system based on PHM. in *2018 Prognostics System. Health Manage Conference (PHM-Chongqing)*. 797–801 (2018). <https://doi.org/10.1109/PHM-Chongqing.2018.00142>
10. A. Hogan, E. Blomqvist, M. Cochez et al., Knowledge graphs. *ACM Comput. Surv.* **54**(4), 1–37 (2021). <https://doi.org/10.1145/3447772>
11. L.Q. Su, Z.Y. Wang, Y.D. Ji et al., A survey based on knowledge graph in fault diagnosis, analysis and prediction: key technologies and challenges. in *2020 International Conference Artificial Intelligence and Computer Engineering (ICAICE)*. 458–462 (2020). <https://doi.org/10.1109/ICAICE51518.2020.00096>
12. J.F. Deng, T. Wang, Z.W. Wang et al., Research on event logic knowledge graph construction method of robot transmission system fault diagnosis. *IEEE Access.* **10**, 17656–17673 (2022). <https://doi.org/10.1109/ACCESS.2022.3150409>
13. H. Liu, R. Ma, D. Li et al., Machinery fault diagnosis based on deep learning for time series analysis and knowledge graphs. *J. Signal Process. Syst.* **93**, 1433–1455 (2021). <https://doi.org/10.1007/s11265-021-01718-3>
14. L.Q. Liu, B. Wang, F.Q. Ma et al., A concurrent fault diagnosis method of transformer based on graph convolutional network and knowledge graph. *Front. Energy Res.* **10**, 837553 (2022). <https://doi.org/10.3389/fenrg.2022.837553>
15. A. Pakonen, T. Mätäsniemi, Ontology-based approach for analyzing nuclear overall I&C architectures. in *IECON 2021—47th Annual Conference IEEE Industrial Electronics Society*. 1–7 (2021). <https://doi.org/10.1109/IECON48115.2021.9589078>.
16. X.Y. Liu, H.W. Wang, Knowledge graph construction and decision support towards transformer fault maintenance. in *2021 IEEE 24th International Conference Computer Supported Cooperative Work Design (CSCWD)*. 661–666 (2021). <https://doi.org/10.1109/CSCWD49262.2021.9437847>
17. X.G. Ma, Knowledge graph construction and application in geosciences: a review. *Comput. Geosci.* **16**, 105082 (2022). <https://doi.org/10.1016/j.cageo.2022.105082>
18. V.R.S. Kumar, A.M. Khamis, S.R. Fiorini et al., Ontologies for industry 4.0. *Knowl. Eng. Rev.* **34**, e37 (2019). <https://doi.org/10.1017/S0269888919000109>
19. L.R. Daniel, N.F. Natalya, A.M. Mark, Protégé: a tool for managing and using terminology in radiology applications. *J. Digit. Imaging* **20**, 34–46 (2007). <https://doi.org/10.1007/s10278-007-9065-0>
20. H. Guan, D. Xie, Research on building of acupuncture domain ontology. in *IEEE International Conference e-Health Networking, Applications Service, Healthcom*. pp. 87–90 (2017). <https://doi.org/10.1109/HealthCom.2017.8210821>
21. Q. Chen, Z. Zhuo, W. Wang, BERT for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909* (2019). <https://doi.org/10.48550/arXiv.1902.10909>
22. J. Devlin, M.W. Chang, K. Lee et al., Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018). <https://doi.org/10.48550/arXiv.1810.04805>
23. Y. Kim, Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014). <https://doi.org/10.48550/arXiv.1408.5882>
24. Y.M. Cui, W.X. Che, T. Liu et al., Pre-training with whole word masking for Chinese BERT. *IEEE/ACM Trans. Audio Speech Lang. Process.* **29**, 3504–3514 (2021). <https://doi.org/10.1109/TASLP.2021.3124365>
25. L. Srinivasan, J. Scharnagle, K. Schilling, Analysis of WebSockets as the new age protocol for remote robot tele-operation. *IFAC Proc.* **46**(29), 83–88 (2013). <https://doi.org/10.3182/20131111-3-KR-2043.00032>
26. T. Zhou, T. Han, E.L. Drogue, Towards trustworthy machine fault diagnosis: a probabilistic Bayesian deep learning framework. *Reliab. Eng. Syst. Saf.* **224**, 108525 (2022). <https://doi.org/10.1016/j.res.2022.108525>
27. X.Z. Li, Q.X. Zhang, H.Y. Tan et al., Fast nuclide identification based on a sequential Bayesian method. *Nucl. Sci. Tech.* **32**, 143 (2021). <https://doi.org/10.1007/s41365-021-00982-z>
28. E. Alhassan, D. Rochman, A. Vasiliev et al., Iterative Bayesian Monte Carlo for nuclear data evaluation. *Nucl. Sci. Tech.* **33**, 50 (2022). <https://doi.org/10.1007/s41365-022-01034-w>
29. J.D. Wang, Z.J. Yang, J.J. Su et al., Root-cause analysis of occurring alarms in thermal power plants based on Bayesian networks. *Int. J. Electr. Power Energy Syst.* **103**, 67–74 (2018). <https://doi.org/10.1016/j.ijepes.2018.05.029>
30. V. Agrawal, B.K. Panigrahi, P.M.V. Subbarao, Intelligent decision support system for detection and root cause analysis of faults in coal mills. *IEEE Trans. Fuzzy Syst.* **25**(4), 934–944 (2017). <https://doi.org/10.1109/TFUZZ.2016.2587325>
31. H.M. Zhao, J. Liu, H.Y. Chen et al., An extension to the noisy-or function to resolve the ‘Explaining Away’ deficiency for practical

- Bayesian network problems. *IEEE Trans. Knowl. Data Eng.* **31**(12), 2441–2445 (2019). <https://doi.org/10.1109/TKDE.2019.2891680>
32. Y.J. Gu, N. Yang, D.C. Chen et al., Study on intelligent fault diagnosis of steam turbines using fault causality information. *Noise Vib. Control.* **39**(4), 12–19 (2019). <https://doi.org/10.3969/j.issn.1006-1355.2019.04.003>. (in Chinese)
33. C. Bielza, P. Larranaga, Bayesian networks in neuroscience: a survey. *Front. Comput. Neurosci.* **8**, 1662–5188 (2014). <https://doi.org/10.3389/fncom.2014.00131>

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.