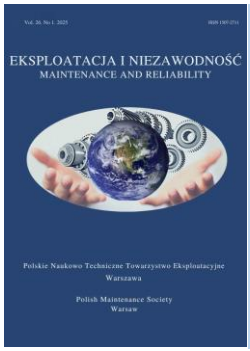

Eksploatacja i Niezawodnosc – Maintenance and Reliability

Volume 27 (2025), Issue 1

journal homepage: http://www.ein.org.pl

Article citation info:
Shi Z, Zhong Q, Zheng S, Wen J,. Peng L, Analysis of Fault Events in Rail Transit Vehicle Traction Systems Based on Knowledge Graph Reasoning, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2025: 27(1) http://doi.org/10.17531/ein/192171

Analysis of Fault Events in Rail Transit Vehicle Traction Systems Based on Knowledge Graph Reasoning

Indexed by: Web of Science Group

Zhao Shi[a], Qianwen Zhong[a,*], Shubin Zheng[a,b], Jing Wen[a], Lele Peng[a]

[a] School of Urban Railway Transportation, Shanghai University of Engineering Science, China
[b] Higher vocational and Technical College, Shanghai University of Engineering Science, China

Highlights

- Propose a novel method based on knowledge graph to handle fragmented fault text data capturing multi-level associations, conducting knowledge inference based on graph structure.
- RGCN-GAT adjusts node weights dynamically to uncover potential correlations, supporting preventive maintenance.
- BERT-BILSTM-CRF enables global semantic sharing, offering interpretable fault analysis by integrating knowledge graphs and Bayesian probability.

Abstract

Fault text records provide detailed information on faults and handling steps, which are valuable for fault analysis. However, the different individuals' recording styles can lead to ambiguities, and it is challenging to uncover potential fault associations in complex systems. To address these issues, this paper proposes a novel method for fault information extraction and analysis. Firstly, to tackle the problem of ambiguous boundaries between entities in fault texts, an integration algorithm is employed to accurately recognize fault entities considering contextual semantic features to establish fault knowledge graph (FKG). Then, a Relational Graph Convolutional Networks (RGCN) is improved with Graph Attention Networks (GAT) for sparse nodes caused by specific types of faults, to dynamically adjust the weight distribution of node learning, inferring potential links within the graph. The proposed method was validated using actual fault records from the traction system of rail transit vehicles, and contributes a reference for the mining and analysis of fault records in complex systems.

Keywords
rail transit vehicle traction system, fault analysis, knowledge graph, knowledge reasoning

1. Introduction

Urban rail transit vehicle operation modes have become increasingly complex, and traditional scheduling decision mechanisms based on human experience are no longer sufficient to meet the needs for rapid fault analysis and response in complex transportation networks[1]. As the power core of rail transit vehicles, faults in the traction system can affect train operation safety and reliability, causing significant losses to urban transportation systems and operating companies[2]. Scholars both domestic and international have summarized fault analysis methods in the field into three categories: analytical model-based[3], expert experience-based, and data-driven techniques[4]. The analytical model-based fault handling method was first proposed by BEARD at Massachusetts Institute of Technology (MIT) [5]. Although this method can accurately obtain fault-derived states, it is challenging to establish precise models due to the influence of uncertainties in complex systems. Expert experience-based methods include expert systems[6], Question&Answer (Q&A) systems[7], Failure

(*) Corresponding author.
E-mail addresses: Z. Shi szshizhao2000@163.com, Q. Zhong (ORCID: 0000-0002-7806-6558) qianwen.zhong@sues.edu.cn, S. Zheng (ORCID: 0000-0002-2327-4245) shubin.zheng@sues.edu.cn, J. Wen wenjing_jlu@126.com, L. Peng (ORCID: 0000-0003-2030-0986) lele.peng@sues.edu.cn,

Modes and Effects Analysis (FMEA)[8], and Case-Based Reasoning (CBR)[9]. These methods rely on the organization and input of a large number of fault samples, reducing the practicality of the models. The development of equipment condition detection system and sensor technology has laid the foundation for data-based fault diagnosis method. Among them, data-driven technology commonly uses statistics-based methods such as time series analysis[10], control chart [11] and regression analysis[12], and signal processing-based methods such as Fourier Transform (FFT)[13], wavelet transform [14] and Hilbert-Yellow transform (HHT) [15]. However, these methods face two main issues: they do not effectively utilize expert prior knowledge, with fault analysis and rich information stored in paper fault texts, causing inconvenience and data resource wastage; and fault analysis lacks interpretability, greatly reducing the credibility of the results.

Currently, fault reporting and reading methods used on-site for urban rail transit vehicle fault handling are manually summarized, with fault event analysis mainly relying on industry experts to extract information from reports[16]. This process is time-consuming, labor-intensive, and prone to errors due to human factors. With the advent of digitalization, text analysis in a big data environment has become an inevitable trend[17]. Furthermore, Knowledge Graph (KG), as an emerging network relationship visualization technology, have been widely applied to large-scale data organization, enhancing the utilization of engineering knowledge[18]. Compared to traditional tabular databases, KG offers several promising features, including: (1) graphical visualization interface functions, (2) integration of intrinsic and related information for more advanced representation and enhanced data understanding and utilization, (3) inference of potential relationships and discovery of implicit links through significant semantic relationships, and (4) effective propagation from initial results to related results based on the graph structure, increasing information dissemination and influence. Therefore, collecting increasingly accumulated domain fault event reports and employing machine learning methods to extract fault-related elements from large-scale, complex, unstructured texts to establish a corresponding KG database, utilizing KG queries[19] and reasoning[20] to identify fault causes[21], can enhance fault handling efficiency and interpretability, maximizing the value of historical data.

In the process of constructing KG, Named Entity Recognition (NER) is considered an advanced NER is considered an advanced technique in Natural Language Processing (NLP). It accurately extracts textual knowledge entities of unlimited length, identifies patterns of proper names in the text, and classifies them into appropriate categories[22]. Zhen et al. [23]proposed an integrated method of common words and syntactic contexts for discontinuous biomedical NER, designing a distance-independent co-occurrence feature mining method to enrich contextual semantic features with fine-grained syntax and long-distance co-occurrence information, solving long-distance dependency problems. Pathak et al. [24]introduced Assamese named entity recognition (AsNER), an annotated NER dataset for low-resource Assamese, with a baseline Assamese NER model. S. Silalahi et al. [25]proposed using deep learning-based NLP technology to extract information related to Unmanned aerial vehicle (UAV) accidents from UAV log messages. Liu et al. [26]completed NER for hazard-related entities in UK railway accident reports, forming a risk KG in railway safety and achieving a quantitative mapping between multi-level hazards and risks. However, due to limitations in the standardization of text records, terminology professionalism, and the scale of annotated corpora, research on fault text mining for rail transit vehicles has yet to be conducted. Additionally, considering the concentration of professional terms and blurred boundaries between entities in Chinese texts, adaptive modeling and optimization of Chinese features are effective and necessary for improving NER performance.

In terms of KG knowledge reasoning, current research mainly focuses on distance-based[27] and semantic-based[28] link prediction. The former has limited expressive power due to the lack of semantic information, while the latter, although considering node semantic information, ignores inter-node associations. FKG nodes contain valuable fault semantic knowledge, and their graph structure is constructed from historical fault handling experience. Therefore, both semantic and structural information should be preserved. Embedding-based link prediction can achieve this goal, creating more effective graph learning methods. Shi et al. [29]designed the project embeddings model (ProjE) to rank candidate entity sets

and select the most matching target entity. Dettmer et al. [30]proposed the Convolutional 2D Knowledge Graph Embeddings model (ConvE), which applies convolution operations to KG knowledge reasoning, capturing interactions between entities and relationships through convolution operations. The Graph Convolutional Networks (GCN) model[31] integrates relational information by merging neighborhood information of nodes. However, there is currently no research utilizing embedding-based knowledge reasoning in FKGs.

To address the research gap outlined above, this paper proposes a method for analyzing fault events in rail transit vehicle traction systems based on knowledge reasoning. The scientific novelty can be summarized as follows:

(1) In the context of fault analysis scenarios where confusion in fault record information hampers the reuse of expert prior knowledge and hinders exploration of potential inter-fault correlations, we establish a comprehensive, efficient, and accurate fault handling and analysis system through entity recognition, construction of fault knowledge graphs, graph neural network inference, and Bayesian probability quantitative analysis. Significant model improvements are made in knowledge graph construction and reasoning tailored to fault analysis domains.

(2) In response to the characteristics of Chinese fault text, we integrate Bidirectional Encoder Representations from Transformers (BERT), Bidirectional Long Short-Term Memory (BiLSTM), and Conditional Random Field (CRF) models. BERT is employed to extract contextual text representations, while the combination of BiLSTM and CRF optimizes global information processing, facilitating the extraction of structured knowledge and the elimination of textual ambiguities resulting from diverse descriptions of the same fault by different maintenance personnel.

(3) We propose a knowledge reasoning model, RGCN-GAT, based on graph embedding. This model dynamically adjusts node learning weight allocation by introducing attention networks, custom-tailored to address the sparsity and imbalanced node types characteristic of fault knowledge graphs.

The rest of the paper is organized as follows. Section 2 introduces the FKG architecture. Section 3 presents the method framework, highlighting improvements to NER and knowledge reasoning algorithms for fault analysis domain. Section 4 conducts experimental validation using actual fault data from the traction system of urban rail transit vehicles, providing application examples of the proposed method for traction system fault analysis. Section 5 concludes the paper and emphasizes future development directions.

2. Fault Knowledge Graph

Fault handling data from the operation and maintenance process can be coordinated in the form of a KG for the purpose of fault analysis. The FKG is intended to depict the fault analysis process. Therefore, it should include the nodes and edges outlined in Table 1.

Table 1.Nodes in FKG.

Node	Description	Edge	Description
Unit	System top-level unit	Contain	Hierarchical relation between component
Component	The mechanical part within the unit	Occur	Component's failure mode
Mode	Fault behavior	Caused by	The reasons of failure mode
Cause	Rationale or justification behind an occurrence	Lead to	The effect of failure mode
Effect	Consequences of an event	Take	Solution for failure mode
Action	An approach to resolving an issue or managing a challenging circumstance	Located in	The train code experiencing the failure mode
Car model	Train code	-	-

As depicted in Figure 1, the constructed FKG in this paper consists of three main levels: component level, fault level, and action level. The component level comprises potential fault components, connections between components, and vehicle type information. The fault level encompasses all fault modes and their impacts, with logical connections between fault modes and impacts mapped to system state assessment criteria, facilitating the exploration of co-occurrence relationships. The

action level identifies fault causes and response strategies.

To supplement the fault attribute nodes in the KG, this paper adopts a combined top-down and bottom-up approach to construct the FKG. The specific steps are as follows: Firstly, through an analysis of the system structure, mechanical connections of components, fault types, and maintenance modes are determined to establish the primary model layer of the KG. Subsequently, building upon this model layer, a bottom-up approach is employed to identify and analyze entities from fault handling data, extracting structured knowledge to form high-quality knowledge representations. Finally, the extracted fault-related entities are incorporated as attribute nodes, completing the update and enrichment of the KG.

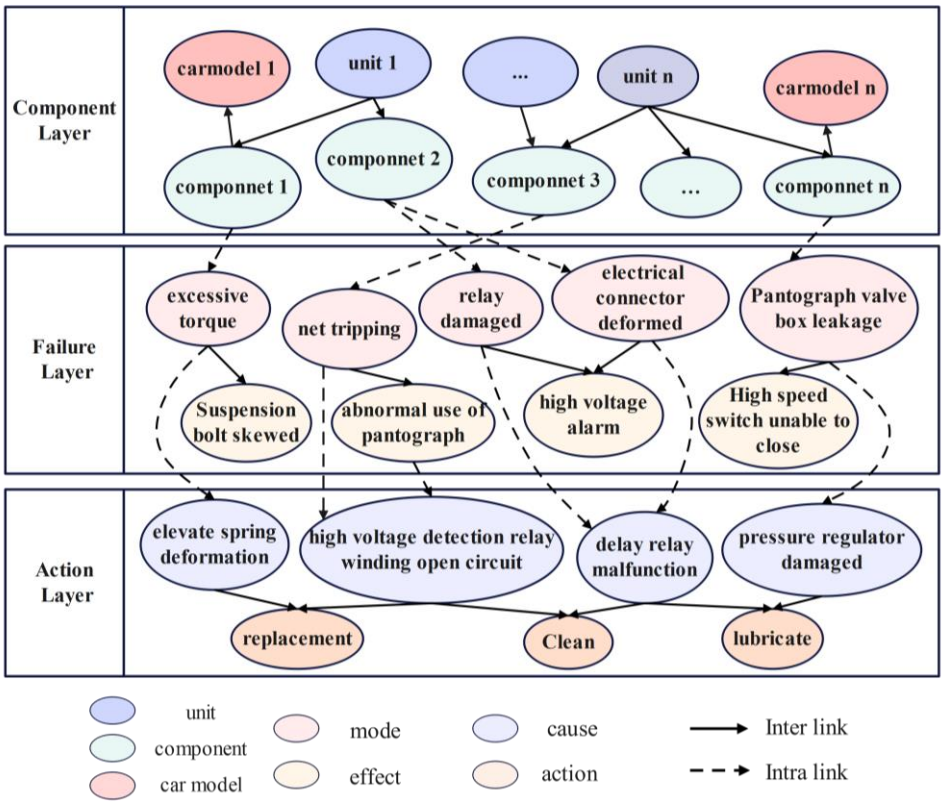

Figure 1. Architecture of the FKG.

3. Method

3.1. Framework

In the field of industrial automation and intelligent operation and maintenance, the use of historical valuable experience can excavate the deep fault connection and reduce the manual dependence of fault handling. Entity recognition technology is widely used to automatically extract key information from fault processing text to provide basic data for subsequent fault analysis and processing. On the basis of entity recognition, the fault knowledge map is further constructed to form a semantic network integrating fault entities, attributes and relationships. In order to realize in-depth mining and reasoning of fault knowledge Graph, this paper adopts Graph Neural Network (GNN) technology to capture complex dependence and interaction between nodes and edges in the graph, so as to reveal the potential pattern and law of fault occurrence. Quantitative analysis combined with Bayesian probability helps operation and maintenance personnel to quickly locate the root cause of the fault and the cause of the fault, and formulate an effective solution. The above method framework is shown in Figure 2.

A. Data Layer: This layer includes relationships and fault data of traction system components, derived from traction system records, fault analysis reports, and maintenance manuals.

B. Building Layer: Fault data is first annotated using the BMEO approach[26], followed by entity recognition using a BERT-BiLSTM-CRF model. Relationships between entities are established based on fault analysis logic rules, and the Neo4j graph database is used to visualize the resulting

knowledge graph.

C. Inference Layer: KG embedding techniques are employed to reveal fault correlations. The RGCN model is applied for deep feature learning within the knowledge graph, and GAT is integrated to flexibly learn dependencies between nodes. To infer new triplets, KG reasoning methods are used within the KG embedding space, identifying potential relationships that are then validated and added to the knowledge graph.

D. Calculating Layer: Bayesian inference is used to calculate fault-related probabilities. Smart search capabilities are implemented using Cypher queries within the Neo4j database, and the FKG, combined with the RGCN-GAT model, is used to enhance the accuracy and efficiency of fault analysis.

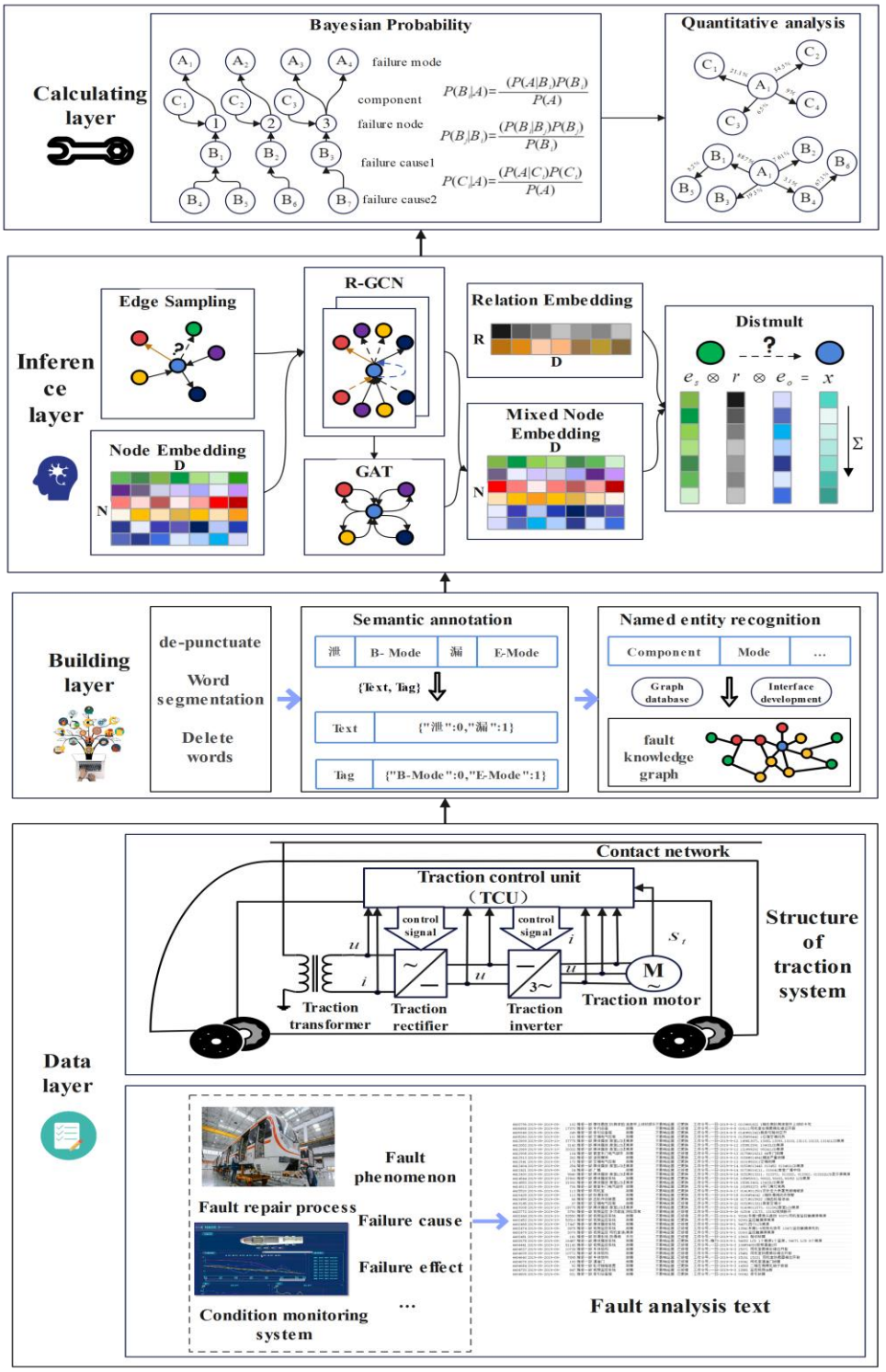

Figure 2. Methodological Framework.

3.2. Construction of the KG

3.2.1. Knowledge Extraction

The construction of the data layer depends on the type of data source. For structured data, methods such as graph mapping or Database to RDF (D2R) conversion can be directly applied. For semi-structured data, processors such as Python are needed, while specialized knowledge extraction methods are required for handling unstructured text data.

The vehicle type field in the table is structured data, so it can be directly converted into RDF graph data format. This paper utilizes the commonly used RDB to RDF Mapping Language Schema (R2RML) to complete the mapping and applies named entity recognition tasks to handle the job content field in the Excel spreadsheet.

3.2.2. Entity Recognition Algorithm Based on BERT-BiLSTM-CRF

Due to the problems in the Chinese fault text, such as concentration of professional terms, blurred boundary between entities, short text content, and large amount of content, and the text ambiguity caused by different maintenance personnel's different description of the same fault, manual rule template is not suitable for entity recognition. To address these issues, this paper adopts the BiLSTM and introduces the BERT, which includes pre-trained models, to complete the modeling and improvement of the entity recognition algorithm based on CRF, as illustrated in Figure 3. The specific steps are as follows:

Step 1: Represent each word in the sentence x as a vector containing word and character embeddings. (taking the word “switch” ("开""关") and “tripped” ("跳") as an example).Character embeddings are randomly initialized, while word embeddings are typically imported from pre-trained word embedding files. All embedding files will be fine-tuned during training.

Step 2: The input of the BiLSTM-CRF model is these embeddings, and the output is the predicted labels for words in sentence x. In this paper, the BMEO tagging method is used to label the data, where B-Label represents the beginning part of a tagged entity, M-Label represents the middle part of a tagged entity, E-Label represents the ending part of a tagged entity, and O represents irrelevant information.

Step 3: All scores predicted by the BiLSTM layer are input into the CRF layer. In the CRF layer, the legal label sequence with the highest predicted score is selected as the best answer.

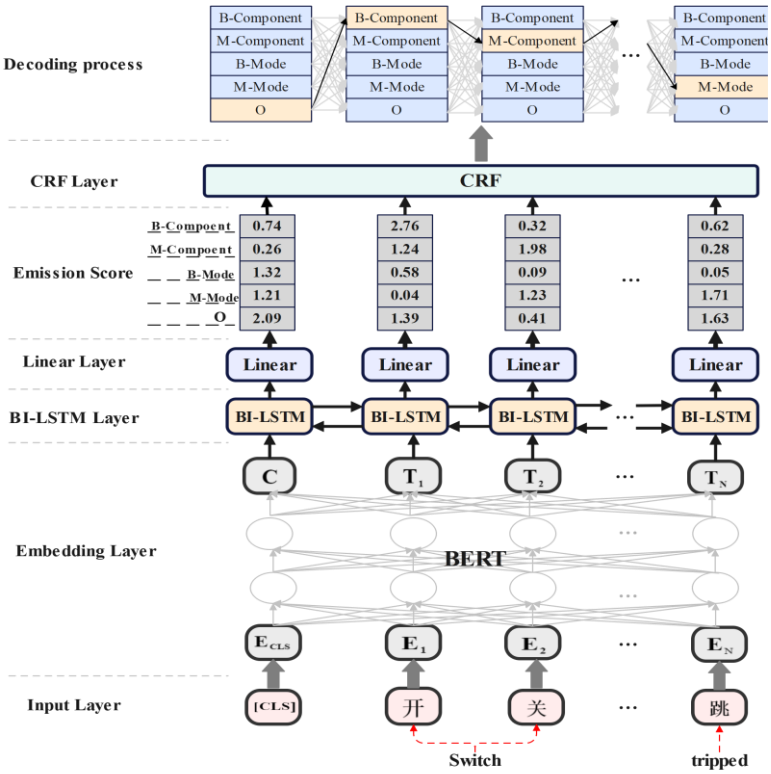

Figure 3. BERT-BiLSTM-CRF model.

BERT, based on transformer, is a bidirectional encoding representation pre-training model (Figure 4), designed to address the problem of long-distance dependencies in traditional Recurrent Neural Network (RNN) models. Each module of the encoder contains a multi-head self-attention mechanism, which can reduce the distance between two distant words to 1, directly calculating the relevance of words. The formula for self-attention is:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \qquad (1)$$

$Atention(Q,K,V)$calculates attention scores using query (Q), key (K), and value (V) vectors, determining the importance of each word in the sequence relative to others. These vectors, derived from word embeddings, are obtained by multiplying them with three weight matrices (w_q, w_k,and w_v) respectively. $softmax$ is applied to the scaled dot-product of the Q and K vectors, normalizing the scores to be between 0 and 1. d is the dimensionality of the input embeddings. The scaling factor, represented by $\sqrt{d_K}$ stabilizes gradients during training by adjusting for the dimensionality of the K vectors.

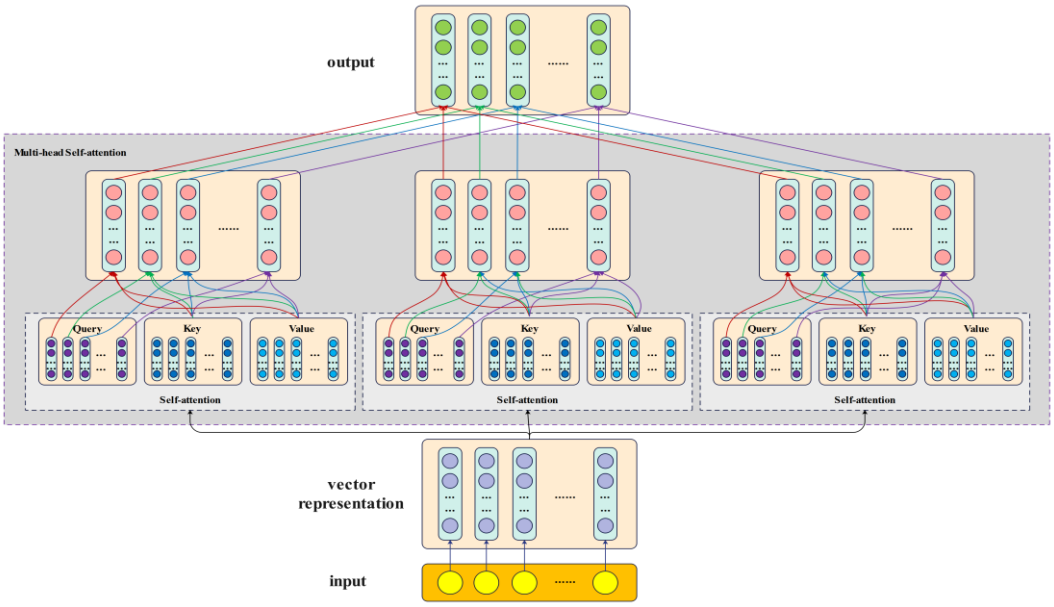

Figure 4. BERT Pre-trained Model.

The BILSTM model comprises two separate LSTM networks. The input sequence is fed into these two LSTM networks, one in a forward order and the other in a reverse order, enabling them to extract features from both directions. The final word feature expression is constructed by concatenating the output vectors from these two LSTMs. Following the BI-LSTM modeling approach, the feature values at time ' t ' not only retain information from both the past and future but also enhance recognition accuracy by making predictions related to the nearby sequence context.

CRF is a probabilistic model that deals with conditional probability distributions for one set of input sequences and another set of output sequences. In the context of linear chain conditional random fields, the characteristic functions can be primarily categorized into two main groups. The first type is a state characteristic function defined on node x, which is only relevant to the current node; The other is a transitive feature function created in the context of node y, which is only relevant to the current node and the previous node. For a given input sequence $X = x_1, x_2, \cdots, x_n$, can get the output tag sequence $Y = y_1, y_2, \cdots, y_n$. The scoring function of the tag sequence can be expressed as:

$$score(X,y) = \sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i) \qquad (2)$$

Where, t_k represents local feature function; s_l represents node characteristic function; λ_k and μ_l are the weight coefficients of t_k and s_l respectively. k represents the number of transition eigenfunctions; l stands for state characteristic function.

3.3. Fault Analysis Based on KG Reasoning

3.3.1. KG Reasoning

KG reasoning aims to uncover potential connections between entities. Traditional methods rely on rule matching or statistical

learning, which struggle to capture complex relationships. Therefore, this paper proposes a new knowledge reasoning method based on graph embedding. The steps are as follows:

Step 1: Utilize random initialization or pre-training methods to generate embeddings for the entities and relationships in the knowledge graph. Apply RGCN to perform convolution operations on the nodes in the knowledge graph, extracting local features of the nodes.

Step 2: Based on the local features extracted by RGCN, GAT is used to model the dependencies between nodes. GAT adaptively assigns attention weights according to the importance of nodes, thereby capturing complex global dependencies.

Step 3: Apply the DistMult model to the final node embeddings to compute the scores of all possible triple combinations in the knowledge graph. Set a score threshold and screen out all triples that exceed this threshold. Finally, add the new triples verified by experts to the knowledge graph.

(1) RGCN

GCN is a layer that operates from graph to graph, with its input comprising node representation vectors and the structure of the graph. For undirected graph $G = (V, \varepsilon)$, V represents finite nodes and ε is a set of edges. When there exists a directed edge from node V_i to node V_j, denoted as $\langle V_i, V_j \rangle \in \varepsilon$, the adjacency matrix A_{ij} is 1; otherwise, it is 0. The message passing rule for a single layer of GCN is as follows:

$$H^{(l+1)} = \sigma(AH^l W^l) \tag{3}$$

Where l is the layer of the graph. H^l represents the nodes at the l layer, W represents the weight parameters, and σ is the non-linear activation function.

To address the drawback of ignoring self-features, a self-looping mechanism is added to each node. An identity matrix I_N is added to the original adjacency matrix A (N is the number of nodes in the graph), so that each node of the modified adjacency matrix $\tilde{A}$ has its own loop connected to it(as shown in formula 4).

$$\tilde{A} = A + I_N \tag{4}$$

During the information aggregation process, the enhancement of features for high-degree nodes and the reduction of features for low-degree nodes may lead to gradient vanishing or explosion. Therefore, a common approach is to consider the degree of neighboring nodes and perform symmetric normalization on the adjacency matrix:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l W^l\right) \tag{5}$$

The FKG is a heterogeneous graph with multiple types of nodes and relationships, surpassing the capacity of traditional GCN. Therefore, it is necessary to consider message passing for different relationships. By separately considering the direction of edges and handling message passing for different relationships, the graph convolution is extended to RGCN[32]. Its computational method is as follows:

$$H^{(l+1)} = \sigma\left(\sum_{r=1}^{R} \tilde{D}_r^{-\frac{1}{2}}\tilde{A}_r\tilde{D}_r^{-\frac{1}{2}}H^{(l)}W_r^{(l)}\right) \tag{6}$$

Where R represents the number of relationship types. Ar, Dr, and $W_r^{(l)}$ denote the adjacency matrix, degree matrix, and weight matrix corresponding to specific relationship types, respectively. In RGCN, different types of edges connected to nodes generate different edge aggregations, integrating the nodes themselves and optimizing node embeddings through fully connected layers. For bidirectional information propagation from the head entity s to the tail entity o, a new edge (o, r', s) is added to each edge (s, r, o), where r' represents the inverse relationship of r. Additionally, a self-loop edge (s, r_{self}, s) is added to each node, where r_{self} denotes a new relationship representing self-information. Figure 5 illustrates the structure of the directed graph.

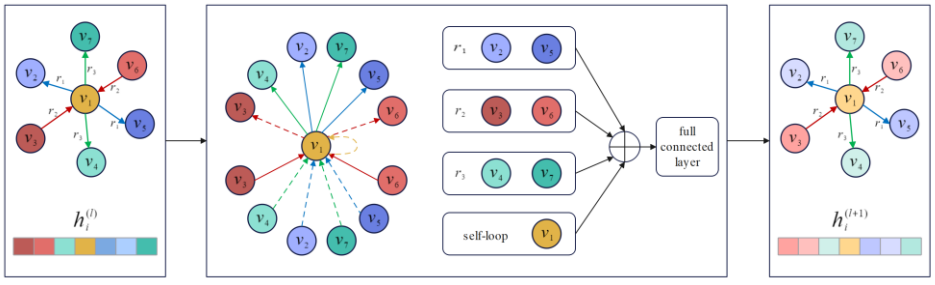

Figure 5. Message Passing Process in RGCN.

(2) GAT

In this paper, we integrate the traditional GAT [33]into the middle of the RGCN mechanism (Figure 6) to address the limitation of traditional heterogeneous graphs. Specifically, the data undergoes processing through the GAT layer post-encoder and pre-decoder. This integration allows for the modulation of learning weights for different nodes, enabling enhanced representation learning, with the equation as follows:

$$H_a^{(l)} = \sigma(\sum_{j \in N_i} \alpha_{ij} W_a v_j) \quad (7)$$

Where N_i represents the neighbors of the node i, W_a represents the weight matrix in the attention mechanism, and v_j denotes the node j. Additionally, the representation of α is as follows:

$$a_{ij} = \frac{\exp\left(LeakyReLU(\lambda^T [W_a v_i || W_a v_j])\right)}{\sum_{k \in Neigh(i)} \exp\left(LeakyReLU(\lambda^T [W_a v_i || W_a v_k])\right)} \quad (8)$$

Where λ represents the weight indicator in the $LeakyReLU$ function, and $||$ denotes the concatenation process. This attention mechanism allocates different weights to nodes, refining the final representation. Additionally, the $LeakyReLU$ function is defined as follows:

$$LeakyReLU(\lambda, x) = \begin{cases} \lambda, & if\ x \geq 0 \\ \lambda \times x & otherwise \end{cases} \quad (9)$$

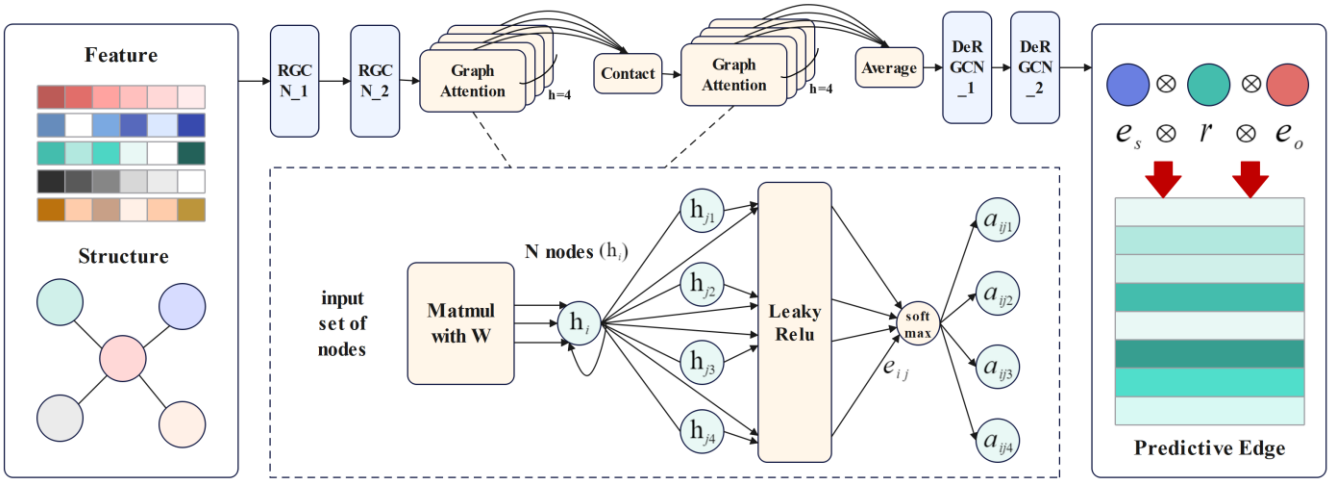

Figure 6. RGCN-GAT Model.

(3) DistMult

The output of the last module in the decoder has been well-trained and can be used to predict potential links in the KG. The likelihood of possible edges is computed using the scoring function[34]. Given a triple (s, r, o), where s, r, and o represent the head node, relation, and tail node respectively, well-trained node representations y_{n_i} can be obtained through the following method:

$$y_{e_i} = f(W_e X_{e_i}) \quad (10)$$

Where X_{e_i} represents the input vector of node e_i, f denotes a non-linear function, and W_e represents the corresponding matrix.

Based on embedded nodes, the combination of triplets (e_i, r, e_j) can be achieved as follows:

$$g_r(y_{e_i}, y_{e_j}) = A_r^T \begin{pmatrix} y_{e_i} \\ y_{e_j} \end{pmatrix} \quad (11)$$

Where A_r^T is a relation parameter. Based on the above output, the loss function is as follows:

$$L(\Omega) = \sum_{(e_i,r,e_j) \in T} \sum_{(e_i',r,e_j') \in T'} max\left\{S_{(e_i',r,e_j')} - S_{(e_i,r,e_j)} + 1, 0\right\} \quad (12)$$

Where T denotes the positive sample triplet, T' denotes the negative sample triplet. Additionally, S represents the scoring function, with the model utilizing matrix multiplication as the scoring function.

3.3.2. Fault Analysis

Using the fault phenomenon phrase matching knowledge graph, the same fault phenomenon nodes are found. Use this node to search for the occurrence times of faulty data nodes, fault causes, and faulty device nodes. The above operations can be implemented using Neo4j graph database query language Cypher.

This paper, based on the idea of Markov process, considers that the current fault occurrence is only related to primary fault causes. This paper calculates the probability $P(B_i|A)$ using the Bayesian formula:

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{P(A)} \quad (13)$$

In the equation, $P(A|B_i)$ represents the probability of fault phenomenon A occurring given the fault cause B_i. $P(B_i)$ represents the probability of the fault cause i occurring. $P(A)$ represents the probability of fault phenomenon A occurring.Using the relationship network of fault causes in the FKG, it is possible to infer the root causes leading to the occurrence of a certain fault phenomenon.

Similarly, the formula for calculating the probability $P(C_i|A)$ of each device component C_i experiencing a fault given the fault phenomenon A is as follows:

$$P(C_i|A) = \frac{n_{AC_i}}{n_A} \quad (14)$$

In the formula, n_A represents the number of occurrences of fault phenomenon A, and n_{AC_i} represents the number of occurrences of fault phenomenon A after device component C_i experiences a fault.

4. Application case: Traction System of Rail Transit Vehicles

4.1. Fault Data

According to the functional classification standards of rail vehicle equipment, the traction system is mainly composed of several key components, including the current receiving device, input circuit, inverter and chopper module, traction motor, traction control unit, and braking resistor. The structure of the traction system is shown in Figure 7. The pantograph supplies direct current to the traction inverter chopper module, which is then converted into three-phase alternating current output to drive the vehicle by the traction control unit, and the vehicle motion is achieved through the traction motor.

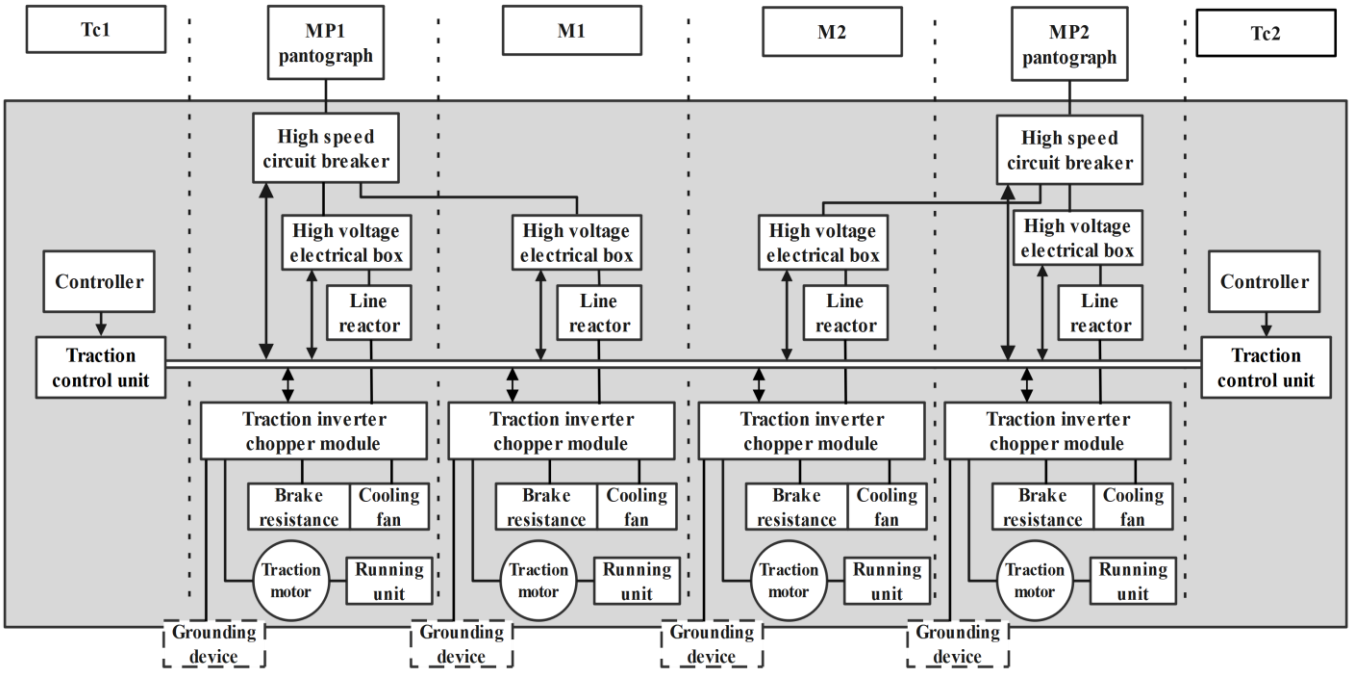

Figure 7. Structure of the Traction System.

This paper conducts an analysis of the key components of the traction system based on the statistical analysis of fault data and an understanding of its operational principles. By analyzing the fault disposal data of the traction system from 2019 to 2021, the overall fault distribution is calculated. During the statistical process, significant faults in the main operating systems of the vehicles can lead to service interruptions, instrument replacement, passenger evacuation, and other serious disruptions. Therefore, analyzing the key components is crucial. The distribution of faults within the traction system is as follows: receiver devices (33%), traction control units (22%), traction equipment boxes (15%), traction motors (13%), input circuits (7%), inverters and chopper modules (4%), braking resistors (3%), and heat dissipation devices (3%). Based on these statistical results, combined with the operational principles of the traction system and expert input, eight types of components including receiver devices, traction control units, traction equipment boxes, traction motors, input circuits, inverters and chopper modules, braking resistors, and heat dissipation devices are selected for in-depth analysis.

4.2. FKG Construction

4.2.1. KG construction model ablation experiment

First, preprocess the text data, including tokenization, removing meaningless words, and semantic annotation. BRAT is an open-source text annotation tool used for labeling named entities. Subsequently, the annotated dataset is divided into training and testing sets for training and evaluating the NER model. For preprocessed documents, NER algorithms based on different models are proposed. Table 2 shows the corresponding settings of optimized training parameters.

Table 2. Main hyperparameter of NER model.

NER model	hyperparameter
BERT	-
BiLSTM	learning rate 1e-4 epochs 700 batch size 50
BiLSTM-CRF	epochs 700
BERT- BiLSTM-CRF	epochs 700

$precision$, $recall$, and $f1_{score}$. They are defined by equations (15)-(17) respectively. Here, TP represents the number of true positives, FP represents the number of false positives, and FN represents the number of false negatives.

$$precision = \frac{TP}{TP + FP} \tag{15}$$

$$recall = \frac{TP}{TP + FN} \tag{16}$$

$$f1_{score} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{17}$$

Figure 8 displays the corresponding loss curves. It can be concluded that the convergence curve of the BERT-BiLSTM-CRF model is the smoothest, indicating a faster convergence speed. According to the results in Table 3, the BERT-BiLSTM-CRF model outperforms other models in terms of entity recognition and average performance. Since most entities in production reports are short words, the BERT model with multi-head self-attention mechanism demonstrates better performance.

Table 3. NER model performance evaluation(%)

NER model	$precision$	$recall$	$f1\_score$
BERT	94.43	94.43	94.42
BiLSTM	84.37	85.32	84.84
BiLSTM-CRF	89.69	90.65	90.17
BERT-BiLSTM-CRF	98.51	97.89	98.21

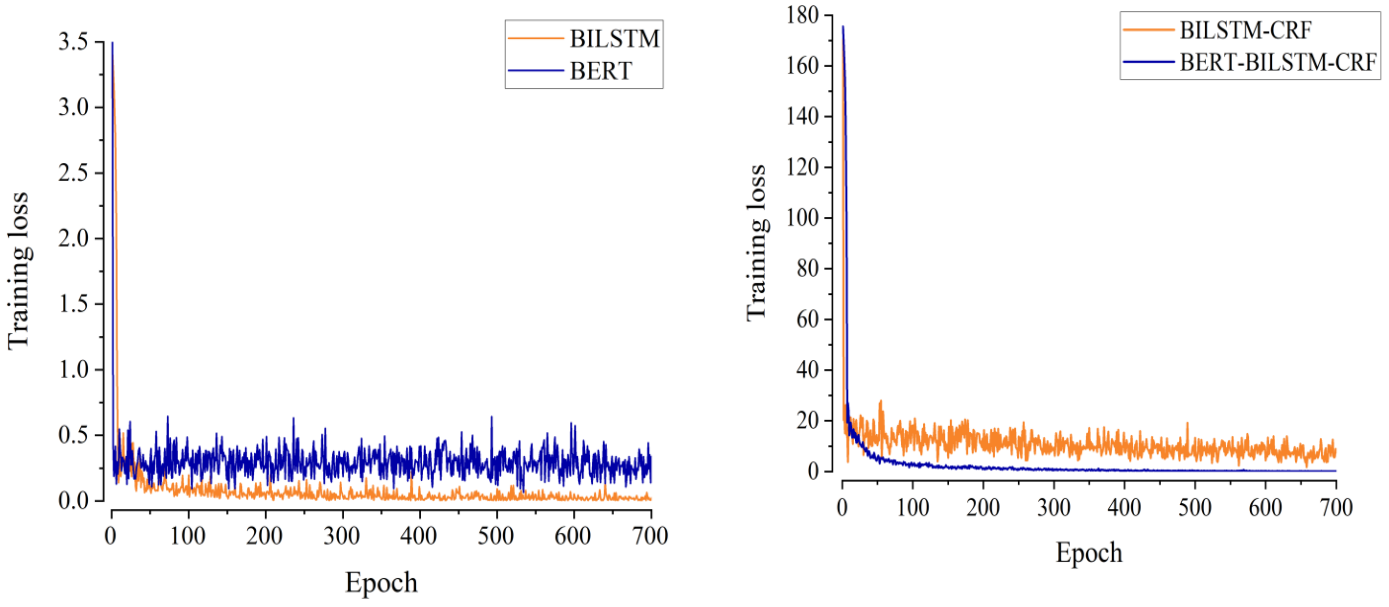

Figure 8. Comparison of Loss Curves.

In the model introduced in this paper, a confusion matrix is used to visually represent the performance of the algorithm, as shown in Figure 9. Clearly, most of the predicted results are consistent with the actual labels, confirming the effectiveness and accuracy of the NER algorithm. However, entities labeled as "B-mode", "E-mode", and "M-effect" exhibit a certain prediction error rate. These errors can be attributed to the unbalanced distribution of annotated data, which affects the prediction accuracy of these specific entity categories.

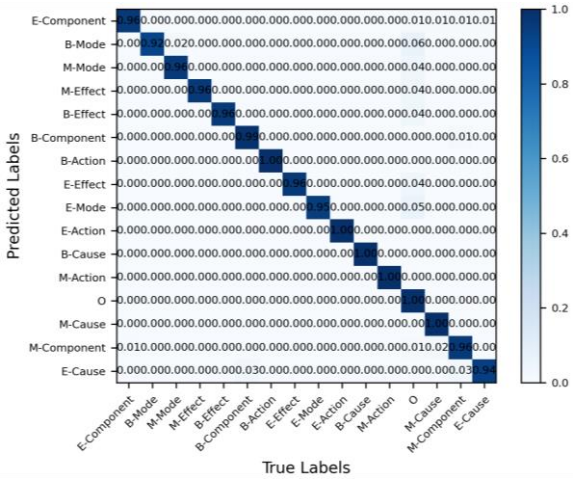

Figure 9. Comparison of Loss Curves.

Additionally, applying the NER model to the test set for classification prediction yielded results as depicted in Figure 10. It can be concluded that BERT-BiLSTM-CRF effectivelyhandles the fault disposal data of the traction system.

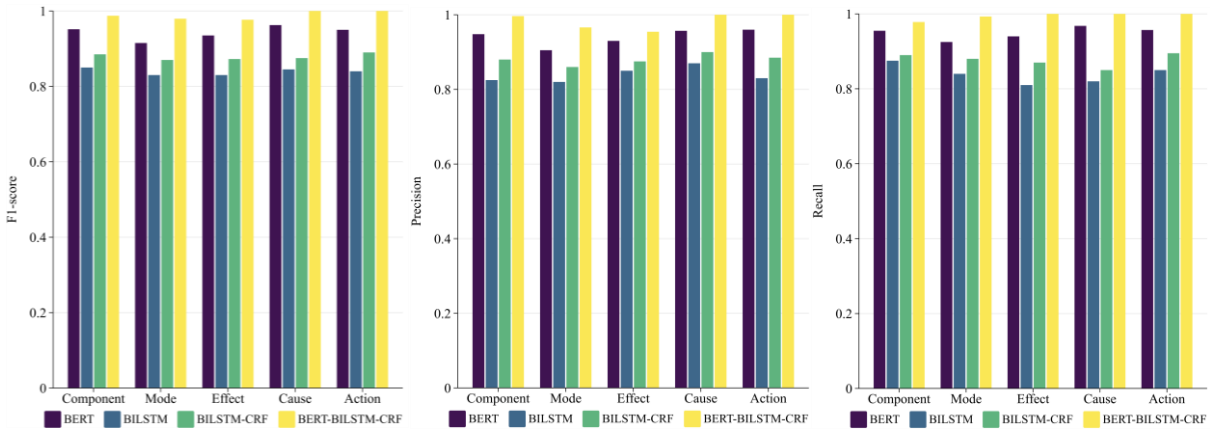

Figure 10. Classification Prediction Results.

4.2.2. KG Construction Results

Entities for KG Construction are extracted from the dataset using the aforementioned NER steps. A partial list of the extracted results is shown in the table 4.

Table 4. Partial NER results

Node attribute	Node name	The node name corresponds to Chinese characters
Component	Bow spring assembly	升弓弹簧组件
	Draw-arch machine	降弓电机
	Drop indicator	落弓指示器
	Electrical control box	电气控制箱
	Air valve box	气阀箱
Mode	Excessive torque	扭力超标
	Pantograph lifting fault	受电弓升起故障
	Contact network tripping	触网跳闸
	Electrical connector deformation	电气连接器变形
	Air leakage in the pantograph air valve box	受电弓气阀箱漏气
Cause	Deformation of lifting spring	升举弹簧变形
	Motor rod twisted	电机光杆扭曲
	Damaged bow indicator	落弓指示器损坏
	Bow lowering delay relay malfunction	降弓延时继电器故障
	Pressure regulating valve damaged	管路漏气

Node attribute	Node name	The node name corresponds to Chinese characters
Effect	Bow bolt deviation	受电弓螺栓偏斜
	Loss of lifting function	丧失升降功能
	Vehicle disconnected	车辆掉线
	Vehicle High Voltage Alert	车辆高压警惕
	Pantograph failure	受电弓故障
Action	Disassembly	拆卸
	Clean and replace	清洁更换
	Clean, lubricate	清洁、润滑

The FKG is constructed using the Neo4j platform (Figure 11). In the graph, brown nodes represent "Unit", pink nodes represent "Component", orange nodes represent "Mode", blue nodes represent "Causes", red nodes represent "Effect", green nodes represent "Action" and purple nodes represent "Car Model". For better understanding, specific examples of some nodes are shown in Table 4.

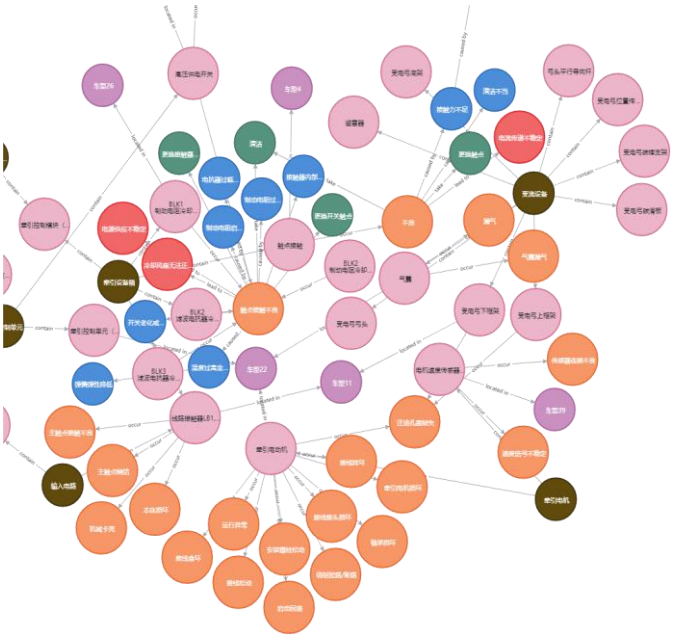

Figure 11. Traction system FKG (local).

4.3. Fault Analysis Based on Knowledge Reasoning

4.3.1. Performance analysis of knowledge reasoning model

This paper employs GNN to infer potential links in the Traction System FKG. To validate the link prediction performance, the evaluation metrics are as follows:

$$Hit@n = \frac{1}{N}\sum_{i=1} N(rank_i \le n) \tag{18}$$

Where $rank$ refers to the ranking of a specific triple, $(.)$ indicates whether the expected triple is present in the selected list. This paper selects $Hit@1$, $Hit@3$, and $Hit@10$ for evaluation. Additionally, Mean Reciprocal Rank (MRR) represents the overall performance, emphasizing the ranking order:

$$MRR = \frac{1}{N}\sum_{i=1} N\left(\frac{1}{rank_i}\right) \tag{19}$$

The main model parameters are set as shown in Table 5. The ratio of training dataset to testing dataset is 5:2. Existing edges serve as positive samples, while non-existent edges serve as negative samples. To demonstrate its superiority, the proposed RGCN-GAT model needs to be compared with other state-of-the-art models, namely: RGCN, GCN, TransE, GAT. Figure 12 comprehensively illustrates the comparison results of MRR, $Hit@1$, $Hit@3$, and $Hit@10$. Overall, the model achieves an MRR score of 0.258, demonstrating a more balanced performance across the entire ranking and better identification of higher-ranked correct links. In terms of $Hit@k$ metrics, RGCN-GAT achieves more than half accuracy in the top 10 predictions. When considering stricter metrics such as $Hit@3$ and $Hit@1$, RGCN-GAT outperforms other models. These metrics collectively reflect the capability of RGCN-GAT in capturing and leveraging graph structural information, as well as its efficiency and accuracy in handling knowledge inference tasks.

Table 5. Main hyperparameter of RGCN-GAT.

Hyperparameter	Value	Hyperparameter	Value
Learning rate	0.01	Batch size	64
Drop rate	0.1	Epoch	1000
Optimize function	Adam	Attention head number	4

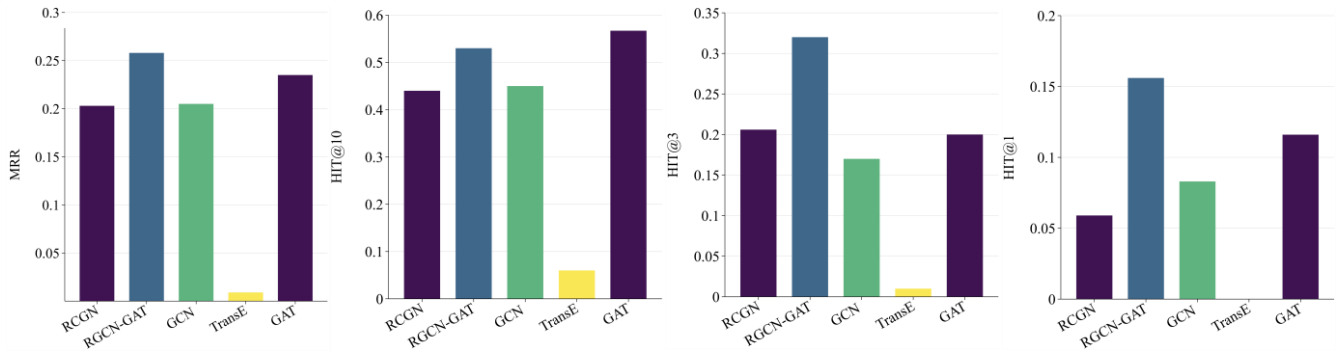

Figure 12. Comparison of Knowledge Inference Models.

Additionally, the node embeddings in the model were visualized to intuitively demonstrate the effectiveness of the proposed model. As shown in Figure 13, the visualization indicates that the proposed RGCN-GAT can cluster nodes of the same type in the same region. This axis represents the coordinates of high-dimensional data mapped to a two-dimensional space using dimensionality reduction techniques, where their relative positions reflect the similarity of data points in the high-dimensional space. While other models are capable of clustering nodes of the same type, they exhibit more dispersion and larger overlaps, demonstrating that the proposed model can successfully generate representative node embeddings.

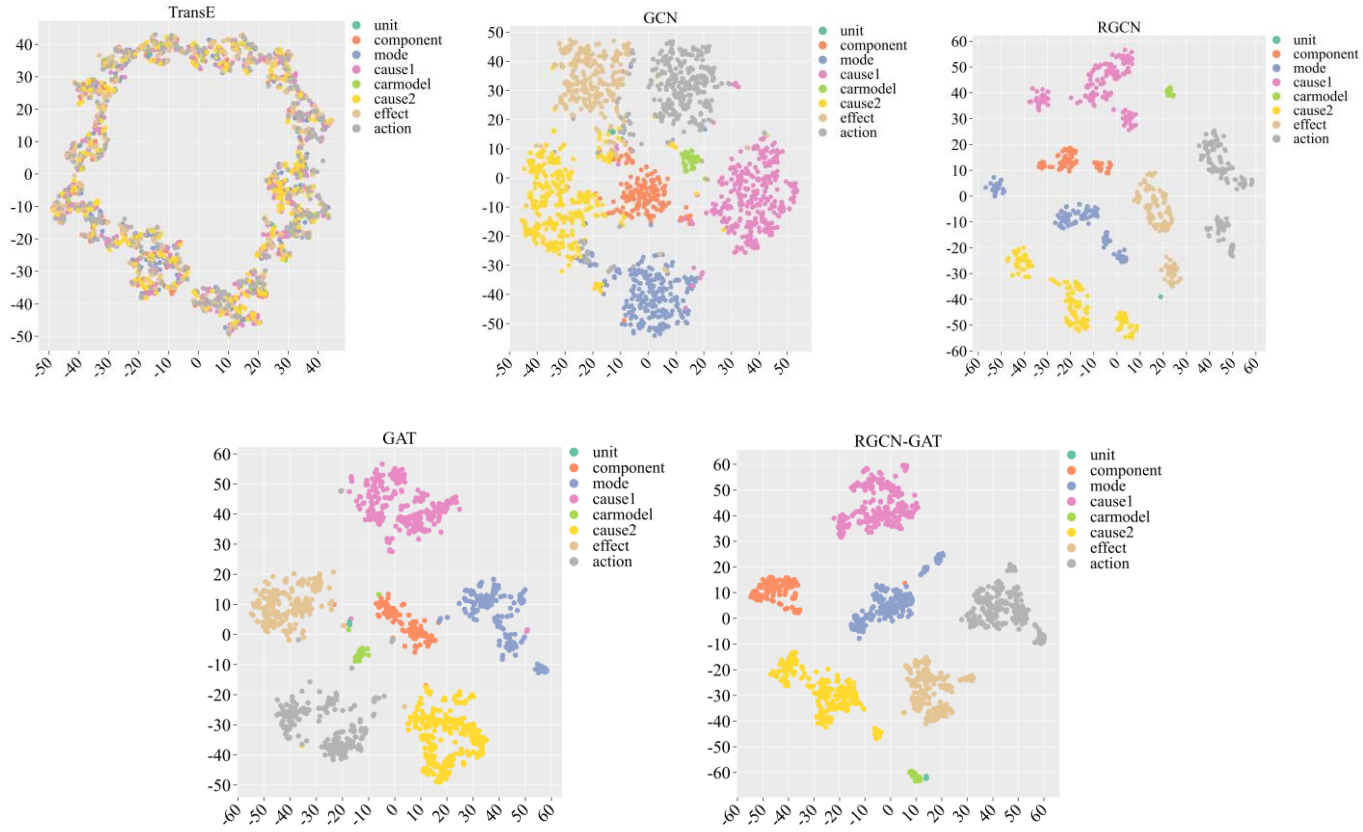

Figure 13. Visualization of Node Embeddings.

4.3.2 Fault Analysis

Neo4j graph database is flexible, interpretable and extensible, and uses the declarative graph query language Cypher. For the traction system fault analysis business scenario, the fault analysis sample frame is shown in the figure 14, which is roughly divided into four steps: fault consistency judgment, knowledge query, knowledge reasoning, and probability calculation.

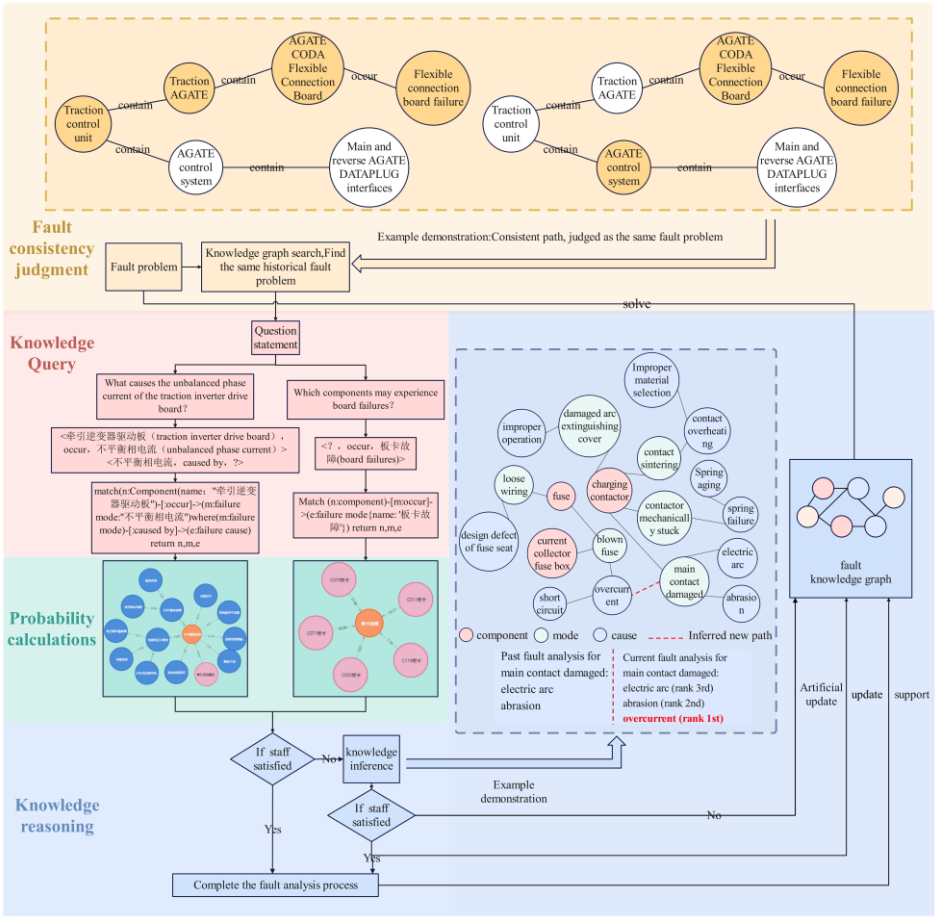

Figure 14. Fault Analysis Sample frame.

Retrieval Fault handling relies on the knowledge and experience of engineers, and similar fault cases can be identified through text retrieval. However, literal similarity may not fully reflect consistency, and it is necessary to understand the relationships in the text information more deeply. The method of determining whether a new fault is consistent with an existing fault through path analysis is more interpretable and simpler. Table 6 fault phenomena are selected for illustration.

Table 6. A set of fault phenomena

Identification Number	Fault Phenomenon
Scenario 1	Work Order Number: On June 17th, 2020, during the fault diagnosis download process, a communication fault was discovered in the traction control unit of MP1 vehicle (080022). The traction AGATE fault cannot be downloaded due to a fault in the AGATE EXT flexible connector board.
Scenario 2	Work Order Number: On January 19th, 2020, the M1 vehicle's AGATE control system experienced loosening of the handle fixing screws, along with a fault in the AGATE EXT flexible connector board, resulting in the inability to connect to the auxiliary inverter computer.

In the table, Scenario 1 and Scenario 2 represent the same fault phenomenon, but describe the fault objects differently. Situation 2 lacks the description of the top-level unit "traction control unit" and does not specify whether the "AGATE control system" controls the traction unit. Merely utilizing NER to structure the fault phenomenon description text cannot determine consistency. However, if the node connection paths are the same, it can be inferred that the two fault phenomena are consistent. As shown in the yellow example section of figure 14, yellow nodes are symptom markers.

The querying process involves problem analysis and interaction with the graph database. The logical form of problem analysis is translated into Cypher query statements, which are then used to retrieve information from the graph database according to the specified query language(as shown in the pink part of figure 14). The table 7 provides examples of query statements. By integrating Bayesian probability, fault causes or faulty components can be accurately identified(as shown in the green part of figure 14).

Table 7. Examples of Query Statements

Question statement	Semantic Analysis	Cypher statement
What causes the unbalanced phase current (不平衡相电流) of the traction inverter drive board(牵引逆变器驱动板)?	<牵引逆变器驱动板，occur，不平衡相电流><不平衡相电流，caused by，?>	match(n:Component(name："牵引逆变器驱动板")-[:occur]->(m:failure mode:"不平衡相电流")where(m:failure mode)-[:caused by]->(e:failure cause) return n,m,e
Which components may experience board failures(板卡故障)?	<？，occur，板卡故障>	Match (n:component)-[m:occur]- >(e:failure mode{name: '板卡故障'}) return n,m,e

When the required knowledge cannot be queried directly, inference of entity links can be made by analyzing the nodes and their interrelationships in the graph, such as the potential connection between nodes like 'main contact damaged' and 'overcurrent' (as shown in the blue sample section of figure 14). This inference is not only based on existing data but also on the model's learning and understanding of complex relationships among fault modes. When engineers and operators apply these inference results, they need to combine them with professional experience and on-site practical situations.

4.3.3. Assessment of Fault Analysis Results

This paper randomly selects 50 pieces of data from the fault text as Test Set 1, and another 50 pieces of new data as Test Set 2. A total of 100 pieces of data are tested to determine whether the fault causes and faulty equipment parts appear correctly in the fault analysis results. Define n_{top} and $n_{top-three}$ as the number of data records where the fault causes appear in the analysis results in the test set and the number of data records where the fault causes rank in the top three in the analysis results in the test set, respectively. Similarly, define the fault equipment analysis parameters n_{eqp} and $n_{eqp-three}$. The table below shows the test results.

Table 8. Fault Analysis Result

Test Set 1	n_{top} 50	$n_{top-three}$ 38	n_{eqp} 50	$n_{eqp-three}$ 42
Test Set 2	n_{top} 35	$n_{top-three}$ 27	n_{eqp} 40	$n_{eqp-three}$ 30
	accuracy			
Test Set 1	100%	76%	100%	84%
	accuracy			
Test Set 2	70%	54%	80%	60%

As the data in Test Set 1 is stored in the KG, the accuracy of fault cause analysis reaches 100%. For Test Set 2, which consists of new fault data, the accuracy of diagnosing the top-level fault causes is 70%, indicating that the fault knowledge stored in the KG is incomplete and requires further exploration of potential associations through knowledge inference. The analysis of faulty equipment parts is relatively effective, suggesting that although there are numerous types of fault causes, the faulty equipment parts are mostly the same, with some being rare occurrences. These rare fault components can be added to the FKG to enhance the knowledge repository.

5. Conclusion

Given the challenges posed by differences in recording styles and potential fault correlations in complex systems, this paper proposes a fault event analysis method for rail transit vehicle traction system based on knowledge graph inference. The research results show that:

(1) In view of the domain specificity and semantic complexity of Chinese fault text, BERT model, as a pre-trained language model, extracts the context representation of the text, which can eliminate the text ambiguity caused by different descriptions of the same fault by different maintenance personnel. Through BiLSTM modeling of context information and global optimization of CRF, domain specific terms and abbreviations can be well handled. The BERT-BiLSTM-CRF model has better applicability to the long word recognition of the traction system fault data set, with an accuracy of 98.51%, which indicates that the model has good performance and robustness in the entity recognition task of Chinese fault text.

(2) To solve the problem of inaccurate knowledge inference caused by some rare nodes in the graph (such as specific types of faults, etc.), GAT is introduced into RGCN, so that the model can focus on the neighbor nodes or relations that are most relevant to rare nodes. For HIT@k, the accuracy rate of the first 10 inference results of the proposed model reaches 76.7%. This model fully explores the hidden fault correlation in the knowledge graph to reduce the failure rate and improve the availability.

Through the organic integration of multiple stages including

entity recognition, construction of fault knowledge graphs, inference using graph neural networks, and Bayesian probability quantitative analysis, this paper establishes an effective and accurate method for fault handling and analysis. Advanced entity recognition algorithms provide foundational data for constructing fault knowledge graphs. These fault knowledge graphs integrate scattered fault information into structured knowledge, forming a knowledge network that intuitively reflects the overall picture of faults. Graph neural networks achieve deep understanding and exploration of fault knowledge through knowledge inference based on graph embedding. Bayesian probability is combined to precisely estimate fault probabilities. This approach enhances system reliability and maintainability, providing strong support for fault prediction and prevention.

While the proposed method allows for intelligent fault analysis, it does not focus on the issue of subsequent maintenance planning, which needs to be further explored. At the same time, further research can be carried out in the following aspects: (1) multimodal heterogeneous KG (2) Node classification methods can be developed on FKG.

References

1. Zhang F, Li S, Zhou P. Experimental Platform for Intelligent Application of Rail Transit. In: 2021 33rd Chinese Control and Decision Conference (CCDC) [Internet]. Kunming, China: IEEE; 2021. p. 912–4. Available from: https://ieeexplore.ieee.org/document/9602767/
2. Chen H, Jiang B. A Review of Fault Detection and Diagnosis for the Traction System in High-Speed Trains. IEEE Trans Intell Transport Syst. 2020 Feb;21(2):450–65. https://doi.org/10.1109/TITS.2019.2897583
3. Gao Z, Cecati C, Ding SX. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches. IEEE Trans Ind Electron. 2015 Jun;62(6):3757–67. https://doi.org/10.1109/TIE.2015.2417501
4. Rezaeianjouybari B, Shang Y. Deep learning for prognostics and health management: State of the art, challenges, and opportunities. Measurement. 2020 Oct;163: https://doi.org/10.1016/j.measurement.2020.107929.
5. BEARD R V. Failure Accomodation in Linear Systems through Self-reorganization[R]. USA:Massachusetts Institute of Technology, 1971.
6. Tan H. A brief history and technical review of the expert system research. IOP Conf Ser: Mater Sci Eng. 2017 Sep 1;242(1):012111. https://doi.org/10.1088/1757-899X/242/1/012111
7. Li Q, Zhang Y, Wang H. Knowledge Base Question Answering for Intelligent Maintenance of Power Plants. In: 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD) [Internet]. Dalian, China: IEEE; 2021. p. 626–31. Available from: https://ieeexplore.ieee.org/document/9437724/
8. Huang J, Li Z, Liu HC. New approach for failure mode and effect analysis using linguistic distribution assessments and TODIM method. Reliability Engineering & System Safety. 2017 Nov;167:302–9. https://doi.org/10.1016/j.ress.2017.06.014
9. Smiti A, Elouedi Z. Dynamic maintenance case base using knowledge discovery techniques for case based reasoning systems. Theoretical Computer Science. 2020 May;817:24–32. https://doi.org/10.1016/j.tcs.2019.06.026
10. Billuroglu B, Livina VN. Full-Cycle Failure Analysis Using Conventional Time Series Analysis and Machine Learning Techniques. J Fail Anal and Preven. 2022 Jun;22(3):1121–34. https://doi.org/10.1007/s11668-022-01381-1
11. Mohammed JS, Abdulhady JA. Rolling bearing fault detection based on vibration signal analysis and cumulative sum control chart. FME Transactions. 2021;49(3):684–95. https://doi.org/10.5937/fme2103684M
12. Wang Y, Han Y. Gearbox Fault Diagnosis Based on a Sparse Principal Component-Generalized Regression Neural Network. Guo K, editor. Mathematical Problems in Engineering. 2022 Sep 20;2022:1–6. https://doi.org/10.1155/2022/1406676
13. Fu Y, Gao Z, Liu Y, Zhang A, Yin X. Actuator and Sensor Fault Classification for Wind Turbine Systems Based on Fast Fourier Transform and Uncorrelated Multi-Linear Principal Component Analysis Techniques. Processes. 2020 Sep 1;8(9):1066. https://doi.org/10.3390/pr8091066
14. Yan R, Shang Z, Xu H, Wen J, Zhao Z, Chen X, et al. Wavelet transform for rotary machine fault diagnosis:10 years revisited. Mechanical Systems and Signal Processing. 2023 Oct;200: https://doi.org/10.1016/j.ymssp.2023.110545.
15. Aqamohammadi AR, Niknam T, Shojaeiyan S, Siano P, Dehghani M. Deep Neural Network with Hilbert–Huang Transform for Smart Fault Detection in Microgrid. Electronics. 2023 Jan 18;12(3): https://doi.org/10.3390/electronics12030499.
16. Wei W, Zhao X. Fault text classification of on-board equipment in high-speed railway based on labeled-Doc2vec and BiGRU. Journal

of Rail Transport Planning & Management. 2023 Jun;26: https://doi.org/10.1016/j.jrtpm.2023.100372.

17. ing X, Wu Z, Zhang L, Li Z, Mu D. Electrical Fault Diagnosis From Text Data: A Supervised Sentence Embedding Combined With Imbalanced Classification. IEEE Trans Ind Electron. 2024 Mar;71(3):3064–73. https://doi.org/10.1109/TIE.2023.3269463
18. Peng C, Xia F, Naseriparsa M, Osborne F. Knowledge Graphs: Opportunities and Challenges. Artif Intell Rev. 2023 Nov;56(11):13071–102. https://doi.org/10.1007/s10462-023-10465-9
19. Liu P, Wang X, Fu Q, Yang Y, Li YF, Zhang Q. KGVQL: A knowledge graph visual query language with bidirectional transformations. Knowledge-Based Systems. 2022 Aug;250: https://doi.org/10.1016/j.knosys.2022.108870.
20. Liu X, Mao T, Shi Y, Ren Y. Overview of knowledge reasoning for knowledge graph. Neurocomputing. 2024 Jun;585: https://doi.org/10.1016/j.neucom.2024.127571.
21. Mao Z, Wang H, Jiang B, Xu J, Guo H. Graph Convolutional Neural Network for Intelligent Fault Diagnosis of Machines via Knowledge Graph. IEEE Trans Ind Inf. 2024 May;20(5):7862–70. https://doi.org/10.1109/TII.2024.3367010
22. Zhong X, Cambria E, Hussain A. Does semantics aid syntax? An empirical study on named entity recognition and classification. Neural Comput & Applic. 2022 Jun;34(11):8373–84. https://doi.org/10.1007/s00521-021-05949-0
23. Zhen Y, Li Y, Zhang P, Yang Z, Zhao R. Frequent words and syntactic context integrated biomedical discontinuous named entity recognition method. J Supercomput. 2023 Aug;79(12):13670–95. https://doi.org/10.1007/s11227-023-05224-0
24. Pathak D, Nandi S, Sarmah P. AsNER - Annotated Dataset and Baseline for Assamese Named Entity recognition.
25. Silalahi S, Ahmad T, Studiawan H. Transformer-Based Named Entity Recognition on Drone Flight Logs to Support Forensic Investigation. IEEE Access. 2023;11:3257–74. https://doi.org/10.1109/ACCESS.2023.3234605
26. Liu C, Yang S, Cui Y, Yang Y. An improved risk assessment method based on a comprehensive weighting algorithm in railway signaling safety analysis. Safety Science. 2020 Aug;128:104768. https://doi.org/10.1109/ACCESS.2017.2759139
27. Chang L, Zhu M, Gu T, Bin C, Qian J, Zhang J. Knowledge Graph Embedding by Dynamic Translation. IEEE Access. 2017;5:20898–907. https://doi.org/10.1109/ACCESS.2017.2759139
28. Mohammadhassanzadeh H, Raza Abidi S, Raza Abidi SS. Plausible reasoning over large health datasets: A novel approach to data analytics leveraging semantics. Knowledge-Based Systems. 2024 Apr;289: https://doi.org/10.1016/j.knosys.2024.111493.
29. Shi B, Weninger T. ProjE: Embedding Projection for Knowledge Graph Completion. AAAI [Internet]. 2017 Feb 12;31(1). Available from: https://doi.org/10.1609/aaai.v31i1.10677
30. Dettmers T, Minervini P, Stenetorp P, Riedel S. Convolutional 2D Knowledge Graph Embeddings. AAAI [Internet]. 2018 Apr 25;32(1). Available from: https://ojs.aaai.org/index.php/AAAI/article/view/11573
31. Wan Y, Chen Z, Hu F, Liu Y, Packianather M, Wang R. Exploiting Knowledge Graph for Multi-faceted Conceptual Modelling using GCN. Procedia Computer Science. 2022;200:1174–83. https://doi.org/10.1016/j.procs.2022.01.317
32. Schlichtkrull M, Kipf TN, Bloem P, Berg R van den, Titov I, Welling M. Modeling Relational Data with Graph Convolutional Networks [Internet]. arXiv; 2017. Available from: https://doi.org/10.1007/978-3-319-93417-4_38
33. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph Attention Networks [Internet]. arXiv; 2018. Available from: http://arxiv.org/abs/1710.10903
34. Yang B, Yih W tau, He X, Gao J, Deng L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases [Internet]. arXiv; 2015. Available from: http://arxiv.org/abs/1412.6575