Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen

doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P8637-cd

A Domino Effect-Driven Knowledge Graph for Large Language Model-Based Risk Identification in Natural Gas Pipeline Operations

Mingyuan Wu

College of Safety and Ocean Engineering, China University of Petroleum (Beijing), China. Key Laboratory of Oil and Gas Safety and Emergency Technology, Ministry of Emergency Management, China. E-mail: 1045913043@qq.com

Jinqiu Hu*

College of Safety and Ocean Engineering, China University of Petroleum (Beijing), China. Key Laboratory of Oil and Gas Safety and Emergency Technology, Ministry of Emergency Management, China. E-mail: hujq@cup.edu.cn,(corresponding author)

Yuhang Lu

College of Safety and Ocean Engineering, China University of Petroleum (Beijing), China. Key Laboratory of Oil and Gas Safety and Emergency Technology, Ministry of Emergency Management, China. E-mail: 2024211051@student.cup.edu.cn

In light of the hallucination issue frequently encountered by large language models (LLMs) in risk identification, a domino effect–based approach is introduced for constructing knowledge graphs that represent risk events, contributing factors, and corresponding mitigation strategies. These knowledge graphs serve as external knowledge bases for LLMs, supported by carefully designed prompt words to enhance retrieval and reasoning capabilities. A System-Theoretic Process Analysis (STPA) of natural gas pipeline operations was employed as a case study to evaluate the effectiveness of this method in improving the risk identification performance of LLMs. The findings indicate that the knowledge graph–based Retrieval-Augmented Generation (RAG) approach significantly reduces the occurrence of hallucinations in LLM outputs, thereby increasing the precision of STPA. This approach presents a novel avenue for utilizing LLMs in risk identification tasks for complex industrial systems.

Keywords: Risk identification, LLMs, Domino effect, Natural gas pipelines, STPA, Knowledge Graph.

1. Introduction

As safety requirements continue to grow in various industrial domains, large-scale, distributed, and highly automated system failures often lead to severe economic losses and environmental damage. (Gnoni et al., 2022; Viana et al., 2022). Although distribution pipeline networks are typically engineered with multiple safety safeguards to mitigate fault propagation, under certain specific conditions—such as equipment aging, inadequate maintenance, or design flaws—a fault may trigger a domino effect, resulting in a cascade of failures. (Gholamizadeh, Zarei, Yazdi, Ramezanifar, & Aliabadi, 2024; Khakzad, 2023; Xiao, Zayed, Meguid, & Sushama, 2024). With the widespread adoption of LLMs, some studies have attempted to apply these models for risk identification. However, the hallucination problem inherent in LLMs can render their risk identification outputs unusable. (Lavrinovics, Biswas, Bjerva, & Hose, 2024). Therefore, a critical challenge is how to minimize or even eliminate the interference and misjudgment caused by such hallucinations when using LLMs for risk identification. Complex industrial systems like natural gas pipelines involve numerous risk factors, and their fault modes and propagation paths are highly coupled. Relying solely on the linguistic reasoning capabilities of LLMs may fail to accurately capture potential risks. (Hong et al., 2023; X. Li, Wang, Abbassi,

& Chen, 2022; Mahmood, Chen, Yodo, & Huang, 2024). Moreover, LLMs often experience “hallucinations,” in which the model fabricates information or overlooks key factors in its responses, leading to deviations and even invalidation of risk identification results. (Metze, Morandin-Reis, Lorand-Metze, & Florindo, 2024). Hence, this paper proposes a method that combines LLMs with a structured knowledge base (such as a knowledge graph), leveraging the model’s language understanding and reasoning capabilities while restricting its output through structured data. This approach implements Retrieval-Augmented Generation (RAG) to reduce hallucinations and enhance the reliability of risk identification. (Arslan, Ghanem, Munawar, & Cruz, 2024; Arslan, Mahdjoubi, & Munawar, 2024).

Currently, preliminary attempts have been made to apply LLMs for risk identification across various fields. (Liu, Li, Ng, Han, & Feng, 2025). In industries like finance, education, healthcare, and transportation, numerous researchers have utilized LLMs as auxiliary tools to identify potential risks. (Al Faraby, Romadhony, & Adiwijaya, 2024; Pu, Yang, Li, & Guo, 2024; Shekhar et al., 2025; Zou et al., 2025). Meanwhile, knowledge graphs also exhibit significant value in the industrial safety domain. Research has shown that by constructing knowledge graphs for multi-source heterogeneous data in sectors such as power, construction, or manufacturing, it is possible to achieve more transparent fault tracing and causal analysis. (Bai, Wu, Ren, Jiang, & Cai, 2023; Z. Li et al., 2023; Zhang, Ruan, Si, & Wang, 2025). However, in the field of risk identification, studies combining knowledge graphs with LLMs remain at a relatively superficial stage of retrieval enhancement; there is still a lack of research on how to deeply support systematic safety analysis methods like System-Theoretic Process Analysis (STPA). (Wong, Zheng, Su, & Tang, 2024). As a result, the “hallucination” outputs generated by LLMs in analytical processes lead to inconsistent and lower-quality results. Currently, several approaches have been proposed to enhance the accuracy of LLMs and mitigate hallucination phenomena, including prompt engineering, fine-tuning with domain-specific datasets, and post-hoc verification techniques. Although each method offers unique advantages, they are often accompanied by challenges such as limited generalizability to new domains, high computational requirements, and a dependence on extensive, high-quality labeled data. To address this, this paper proposes a domino-effect-driven method for constructing the schema layer of a knowledge graph and realizes automatic generation of knowledge graphs based on an LLM. Subsequently, this knowledge graph is integrated into the LLM’s generative process as an external knowledge base. By leveraging a retrieval-augmented generation mechanism, it provides contextual support for STPA, thereby mitigating the shortcomings of traditional knowledge graphs in representing risk causality and propagation patterns, and enhancing the LLM’s capacity for understanding and reasoning about risk identification in industrial settings. A case study further validates the feasibility of this approach in the context of natural gas pipelines.

The structure of this paper is organized as follows: Chapter 2 provides a comprehensive introduction to the schema construction and generation methods for the domino-effect-driven knowledge graph. Chapter 3 focuses on a case study involving the development of a knowledge graph in the natural gas pipeline domain and explains how retrieval-augmented generation, based on the knowledge graph, is applied to support STPA during the operational phase of natural gas pipelines. Finally, Chapter 4 presents the conclusions and discusses potential directions for future research.

2. Method

2.1. *Domino Effect-Driven Knowledge Graph Schema Layer Construction*

To use the knowledge graph as an external knowledge base for LLMs and to enhance the STPA capabilities of LLMs, the schema layer of the knowledge graph must be designed to capture the causal relationships behind risk propagation, analysis, and mitigation. This involves clarifying the types and meanings of nodes as well as the types and logical relationships of edges in order to accurately depict the risk relationships between nodes. As a theory that describes the causal chain of events, the domino effect can reflect the entire progression of risk from its trigger through propagation to its eventual outcome.

(Gholamizadeh, Zarei, Yazdi, Ramezanifar, & Aliabadi, 2024).

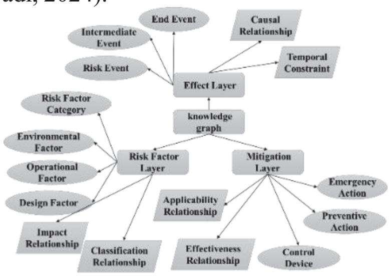

Fig. 1. Schema Layer of the Knowledge Graph.

Therefore, this paper integrates the concept of the domino effect into the construction of the knowledge graph schema layer so that it aligns with the requirements of STPA. The structure is shown in Figure 1.

In constructing the schema layer of the knowledge graph, different shapes are used to represent the types and hierarchical relationships of core elements. (Hogan et al. 2021). Ovals signify the node types in the knowledge graph; parallelograms denote the edge types; and rectangles represent the layers to which nodes and edges belong. To systematically illustrate the key elements of the domino effect and their mutual relationships, the knowledge graph is divided into three layers: the event layer, the risk factor layer, and the countermeasure layer. The event layer mainly describes the causal chain of risk events (e.g., equipment failure or pipeline leaks) and uses causal relationships and temporal constraints to depict the evolution of the domino effect. The risk factor layer analyzes the major drivers of events, including environmental, operational, and design factors, revealing the causes and characteristics of risks through influence and classification relationships. The countermeasure layer uses applicability and effectiveness relationships to represent the emergency and preventive measures aimed at events and risk factors.

2.2. *Knowledge Graph Generation Method*

The generation of the knowledge graph proceeds in three steps. First, text information is input into the LLM, and the model is instructed to output the relevant information in a standardized format such as triples. Second, a tool is chosen to build the knowledge graph. This paper uses Neo4j, where the query language is known as Cypher. Third, tools like Python automatically convert the standardized output into Cypher statements, which are then fed into Neo4j to ultimately generate the knowledge graph. (Elapolu et al. 2024).

(i) Prompt Template Construction

In order to guide the LLM to generate content in the desired standardized format, a prompt template must be designed so that, based on the input text, the model outputs the formatted information—such as triples—that cover the specific domain. To ensure output quality, the prompt must explicitly require that all necessary information be included in the generated triples and prohibit omitting any details, as incomplete output would undermine its reliability. Additionally, any generated triple must be reviewed to avoid information errors. The core function of the prompt template is to specify the types and logical structure of the triples while ensuring the completeness and accuracy of information extraction. Prompt Template: Given the text below, construct knowledge graph triples relevant to the domain of natural gas pipelines. The triples should be categorized as either Entity-Attribute-Value or Entity-Relationship-Entity. Adhere strictly to these categories when generating the triples, ensuring completeness, logical consistency, and factual accuracy. Refer to Table 1 for examples of inputs and outputs based on this prompt template.

Table 1. Input and Output Examples for LLMs

Example Input	Example Output
Under high-temperature conditions, natural gas pipelines are prone to material fatigue, which may lead to pipeline leakage.	(Natural gas pipeline, high-temperature conditions, material fatigue); (Material fatigue, leads to, pipeline leakage);

(ii) Prompt Content

The prompt content primarily comes from previously conducted risk analysis reports, which may include but are not limited to results from safety risk analyses employing Hazard and Operability Analysis (HAZOP), Hazard Identification (HAZID), and other methods. Ultimately, the prompt content and the prompt template are combined to form a complete prompt and input into the LLM, which outputs the information in the form of triples or other standardized structures.

(iii) Knowledge Graph Generation

Using Python, the formatted text (e.g., triples) generated by the LLM is processed and converted into Cypher statements, which are then input into Neo4j. Two types of Cypher statements are employed in this process: one for creating nodes and another for adding relationships. (Elapolu et al. 2024).

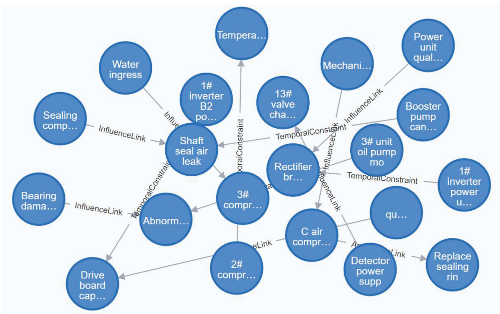

Fig. 2. The Generated Knowledge Graph.

2.3. *Knowledge Graph Retrieval-Augmented Generation*

(i) Matching Queries with Nodes

To enable RAG based on the knowledge graph, user queries posed to the LLM must be matched with corresponding nodes in the knowledge graph. A common approach for this process is vector-based matching. Specifically, a pre-trained language model (e.g., BERT) is used to encode the user query into a semantic vector, while the same encoding process is applied to the nodes of the knowledge graph, including entity names or descriptive text, to generate semantic vectors for the nodes. (Devlin et al. 2019). A mapping function is then defined to project both the query vectors and node vectors into the same semantic space, thereby enabling semantic matching.The primary objective of this mapping function is to optimize the semantic distance or similarity between query vectors and node vectors, ensuring high alignment within the semantic space. Due to the complex logic and semantic relationships involved in STPA, the mapping between query vectors and node vectors is often intricate. Simple linear mapping methods are insufficient to capture these deeper semantic relationships. To address this, a two-layer linear transformation with a ReLU activation function is employed, enabling nonlinear mapping to better model the complex semantic relationships between query vectors and node vectors. The formulation of this mapping function is provided in Equation (1).

$$f(v_q) = \mathrm{ReLU}(W_2 \cdot (\mathrm{ReLU}(W_1 \cdot v_q + b_1)) + b_2) \quad (1)$$

The semantic vector of the question is denoted as v_q, where W_1 and W_2 represent weight matrices, and b_1 and b_2 represent bias vectors. The mapped question vector is denoted as $f(v_q)$.

After designing the mapping function, it must be trained. Hence, a high-quality training set is manually constructed, containing “question–node” pairings. The “question” refers to a specific user need stated in natural language, and the “node” is a knowledge graph entity related to the question’s semantics. For instance, for the question “Which loss-of-control scenarios might pipeline corrosion lead to?”, relevant graph nodes might include “failure of circumferential welds” or “pipeline cracking under stress corrosion.” Expert knowledge and manual vetting are required to ensure semantic consistency and reasonable matching. The format of the training data is shown in Equation (2).

$$\{(v_q^{(1)}, v_n^{(1)}), (v_q^{(2)}, v_n^{(2)}), \ldots, (v_q^{(N)}, v_n^{(N)})\} \quad (2)$$

The semantic vector of the question is denoted as $v_q^{(i)}$, and the semantic vector of the knowledge graph node is denoted as $v_n^{(i)}$.

The cosine similarity formula is utilized to train the model using paired data consisting of question vectors and their corresponding node vectors, with the objective of maximizing their similarity. This approach facilitates the accurate retrieval of relevant nodes for a given query. During the training process of the mapping function, it is crucial to select an appropriate loss function as the evaluation metric to assess the quality of the mapping results. Depending on the specific task objectives, the choice of loss function can be categorized as follows: when the goal is to maintain directional consistency in the semantic space, cosine similarity loss is preferable as it directly optimizes the directional similarity between vectors; (Gao et al. 2021) when focusing on numerical matching or directly minimizing distance, Euclidean distance loss can be chosen to minimize numerical errors between vectors; when further distinction between positive and negative samples is required, contrastive loss or triplet loss can be employed to enhance the model's discriminative ability and generalization performance. (Reimers and Gurevych 2019). In retrieval-augmented generation tasks, the degree of vector direction matching directly impacts the precision and efficiency of semantic retrieval. (Lewis et al. 2020). Therefore, ensuring semantic directional consistency between question vectors and node vectors is critical for efficient retrieval. Additionally, whether the retrieved knowledge nodes can provide high-quality contextual support for the generative model depends on the accuracy of semantic matching and the contextual quality generated by the model. As a result, cosine similarity loss is selected as the core evaluation metric. Given a question vector v_q and a node vector v_n, the mapped question vector is $f(v_q)$. The cosine similarity is defined as Equation (3), and the loss function is given as Equation (4).

$$\mathrm{CoS}(f(v_q), v_n) = \frac{f(v_q) \cdot v_n}{\| f(v_q) \| \ \| v_n \|} \tag{3}$$

The loss function L is defined as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \mathrm{CoS}(f(v_q^{(i)}), v_n^{(i)}) \tag{4}$$

Training is performed using paired data (question vectors and their corresponding node vectors) based on the cosine similarity calculation formula, aiming to maximize the similarity between the two. This ensures precise retrieval of questions and their related nodes.

(ii) Searching for Related Nodes

After the most relevant node is matched, the triples connecting this node to its neighboring nodes are extracted. These triples encapsulate semantic associations and logical relationships between nodes, offering extensive contextual support for subsequent generation tasks.

To improve the interpretability of these triples, a prompt template must be designed. The LLM then gives an initial explanation, generating a text description suited to the scenario. The output is combined with the user's original question to form a complete prompt, which is then input into the LLM.

3. Case Analysis

3.1. *Knowledge Graph Construction Driven by the Domino Effect in the Natural Gas Pipeline Domain*

In this case study, the data are sourced from documents such as natural gas pipeline design reports, risk assessment reports, and construction specifications. Python 3.9 is used for data processing. Based on these documents, a preliminary analysis of the risk-related relationships present during natural gas pipeline operations was conducted. Subsequently, relationships among nodes were determined according to the knowledge graph schema described in Section 2.1, as illustrated in Figure 3. The first, second, and third columns in the figure represent the source node, relationship, and target node, respectively.

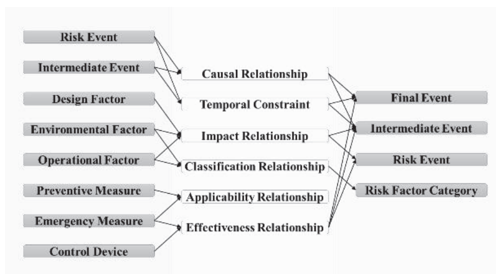

Fig. 3. Relationships Between Nodes in the Knowledge Graph.

The next step involves inputting the existing textual data into the LLM using the designed prompt template. The model extracts information such as events, risk factors, and mitigation measures described in the text. It is important to note that, in this case, the content output by the LLM includes only source nodes and target nodes. The relationship types between the nodes are pre-defined within the knowledge graph generation model. A total of 879 relationships are generated. The content output by the LLM is shown in Table 2.

Table 2. Structured Text Output by the LLM.

Index	Structured Text
1	{"risk_event": ["Unit 1 Fault Shutdown"], "intermediate_event": ["Inverter Output Grounding Fault"]}
…	…
879	{"risk_event": ["Unit 3 Oil Pump Motor Bearing Damage"], "intermediate_event":

["Abnormal Oil Pump Noise"]}

After the output is proofread, Python's py2neo library is utilized to input the structured text into Neo4j, resulting in the generation of the knowledge graph. The graph comprises 1,205 nodes and 879 edges, with the distribution of node types and edges aligning with the actual characteristics of risk propagation during the operational phase of natural gas pipelines.

3.2. *STPA Based on Knowledge Graph Retrieval-Augmented Generation*

STPA is a systematic safety analysis method that requires a clear definition of analysis objectives and a structured approach to guide the analysis process. This paper illustrates how to generate STPA analysis text using a retrieval-enhanced mechanism based on a knowledge graph through a case study involving a natural gas pipeline compressor component. To guide the analysis, a prompt template is designed as follows: "*I am conducting a systematic STPA safety analysis. Below is the system description: [System Description]. Please help me complete the STPA analysis by following these steps: 1. Identify potential hazardous events in the system and briefly explain their possible impacts. 2. Based on the system description, draw the control structure or list the key control components and their interactions in text form. 3. Identify potential unsafe control actions (UCAs) in the system. 4. For each unsafe control action, analyze the potential causal factors and propose improvement suggestions or safety constraints. Please provide detailed step-by-step answers.*" The "System Description" section is adaptable to meet specific requirements.To achieve alignment between the prompts and knowledge graph nodes, keywords must first be extracted from the prompts. This involves tokenizing the input text and vectorizing all tokenized words as well as the nodes in the knowledge graph using BERT model. After vectorization, a mapping function is employed to align the keywords from the prompts with nodes in the knowledge graph, thereby achieving prompt-to-node matching. To facilitate this, a domain-specific training dataset for natural gas pipelines was designed to train the mapping function. Once trained, the mapping function accurately aligns input information with knowledge graph nodes. Following the alignment, neighboring nodes are retrieved to form triples.

The system description used in this case is as follows: "*The compressor monitors operating pressure, temperature, and flow rate through sensors. The system includes an automatic overload protection mechanism that shuts down the compressor in cases of abnormal pressure or overheating. If a sensor fails or the overload protection mechanism malfunctions, the compressor may operate under overload conditions or exhibit abnormal temperatures.*" The node matching results are presented in Table 3.

Table 3. Node Matching Results.

Matched Nodes	Triples
Risk Events: Compressor Operating Pressure Abnormally High, Sensor Failure.	(Compressor Operating Pressure Abnormally High, Causal Relationship, Overload Protection Device Not Triggered) (Sensor Failure, Causal Relationship, Abnormal Temperature Data Not Uploaded)

Subsequently, these triples and the prompt template are fed into the LLM, which provides an interpretation of the triples. Finally, the generated explanation is embedded into the system description section of the initial prompt template and input back into the LLM. Based on this new input, the model produces a more complete and contextually relevant STPA.

3.3. *Results Analysis*

A comparison was conducted between the original STPA results and the enhanced STPA analysis results generated using knowledge graph-based retrieval, as shown in Table 4. The STPA results under the two sets of prompts exhibit distinct emphases and characteristics.The first set of prompts is more general, addressing overarching system safety requirements such as "compressor overload operation or abnormal temperature." This set is suitable for comprehensive, system-level preliminary safety assessments. It identifies events like overload, overheating, sensor failures, and protection device malfunctions, providing a broad analysis that is relatively thorough but lacks specific examples. The analysis of unsafe control actions (UCAs) focuses on a macro-level perspective, such as "the control system failing to shut down in time" or "sensor failures going unnoticed."

Correspondingly, the improvement suggestions are also high-level measures, such as increasing redundancy, enhancing maintenance, or optimizing algorithms. In contrast, the second set of prompts is more targeted and scenario-specific, focusing on concrete situations such as "failure to relieve abnormal pressure increases in time" or "temperature sensor failure causing data loss." This specificity makes the description of hazardous events more precise and enables UCA analysis to be more actionable. For instance, it identifies issues such as "the overload protection device failing to respond during abnormal pressure increases" or "temperature sensor failure leading to an inability to monitor overheating." The improvement suggestions are also more detailed and practical, such as automatically entering a safe mode upon detecting temperature data loss or executing protective actions during abnormal pressure increases.

Table 4. Comparison of Original STPA Results and Knowledge Graph Retrieval-Augmented STPA Results.

Original STPA Results

Step 1: Identify Potential Hazardous Events and Their Impacts

1. **Compressor Overloading**
   Potential Impact: Overloading may lead to mechanical fatigue, part failure, and damage to critical components, disrupting operations and posing safety risks.
2. **Compressor Overheating**
   Potential Impact: Overheating can degrade lubricants, cause seal damage, and risk fire or explosion, leading to catastrophic failures.
3. **Excessive Pressure Without Relief**
   Potential Impact: Uncontrolled pressure buildup may rupture pipes or storage tanks, resulting in severe equipment damage and safety hazards.
4. **Sensor Failure**
   Potential Impact: Failure in pressure, temperature, or flow sensors can lead to inaccurate monitoring, delaying protection measures and increasing accident risk.
5. **Overload Protection Malfunction**
   Potential Impact: If the protection mechanism fails, the system cannot shut down or release pressure during abnormal conditions, escalating risks to equipment and personnel.

Knowledge Graph Retrieval-Augmented STPA Results

Step 1: Identify Hazardous Events

1. **Pressure Rise Without Shutdown**
   Potential Impact: Sustained high pressure may lead to component overloading, ruptures, or catastrophic failure.
2. **Sensor Failure Preventing Temperature Monitoring**
   Potential Impact: Lack of real-time monitoring leads to undetected overheating, equipment wear, or fire risks.
3. **General Risk Escalation from Combined Failures**
   Impact: Overloading and overheating amplify operational and safety risks, potentially affecting nearby personnel and environment.

4. Conclusion

4.1. *Summary*

This paper proposes a knowledge graph construction method based on the domino effect, integrating the knowledge graph into a retrieval-augmented generation framework as an external knowledge base. This approach provides contextual enhancement for STPA and facilitates the generation of more precise and detailed safety analyses. Additionally, the feasibility and effectiveness of utilizing a domino-effect-driven knowledge graph as an external knowledge base for LLMs are validated in the areas of risk identification, causal analysis, and safety strategy development.

4.2. *Outlook*

Although the proposed method has yielded promising results in practical applications, there is still room for improvement. Future work could further enrich the structure and content of the knowledge graph by integrating additional domain-specific risk analysis data, thereby enhancing its adaptability to complex system safety scenarios. Moreover, more advanced nonlinear mapping and semantic reasoning algorithms could be employed in retrieval and generation processes to further improve retrieval accuracy and the overall quality of the generated results.

Acknowledgment

This work was supported by the Natural Science Foundation of China (52074323).

References

Al Faraby, S., Romadhony, A., & Adiwijaya. (2024). Analysis of LLMs for educational question classification and generation. *Computers and Education: Artificial Intelligence, 7*, 100298.

Arslan, M., Ghanem, H., Munawar, S., & Cruz, C. (2024). A Survey on RAG with LLMs. *Procedia Computer Science, 246*, 3781-3790.

Arslan, M., Mahdjoubi, L., & Munawar, S. (2024). Driving sustainable energy transitions with a multi-source RAG-LLM system. *Energy and Buildings, 324*, 114827.

Bai, Y., Wu, J., Ren, Q., Jiang, Y., & Cai, J. (2023). A BN-based risk assessment model of natural gas pipelines integrating knowledge graph and DEMATEL. *Process Safety and Environmental Protection, 171*, 640-654.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *In Proceedings of the 2019 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171-4186.

Elapolu, M. S. R., Rai, R., Gorsich, D. J., Rizzo, D., Rapp, S., & Castanier, M. P. (2024). Blockchain technology for requirement traceability in systems engineering. *Information Systems*, 123, 102384.

Gao, T., Yao, X., & Chen, D. (2021). Simcse: Simple contrastive learning of sentence embeddings. *arXiv Preprint arXiv:2104.08821.*

Gholamizadeh, K., Zarei, E., Yazdi, M., Ramezanifar, E., & Aliabadi, M. M. (2024). A hybrid model for dynamic analysis of domino effects in chemical process industries. *Reliability Engineering & System Safety, 241*, 109654.

Gnoni, M. G., Tornese, F., Guglielmi, A., Pellicci, M., Campo, G., & De Merich, D. (2022). Near miss management systems in the industrial sector: A literature review. *Safety Science, 150*, 105704.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., De Melo, G., Gutierrez, C., Kirrane, S. et al. (2021). *Knowledge graphs. ACM Computing Surveys (CSUR),* 54, no. 4, 1-37.

Hong, B., Shao, B., Guo, J., Fu, J., Li, C., & Zhu, B. (2023). Dynamic Bayesian network risk probability evolution for third-party damage of natural gas pipelines. *Applied Energy, 333*, 120620.

Khakzad, N. (2023). A goal programming approach to multi-objective optimization of firefighting strategies in the event of domino effects. *Reliability Engineering & System Safety, 239*, 109523.

Lavrinovics, E., Biswas, R., Bjerva, J., & Hose, K. J. J. o. W. S. (2024). Knowledge Graphs, Large Language Models, and Hallucinations: An NLP Perspective. 100844.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H. et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems,* 33, 9459-9474.

Li, X., Wang, J., Abbassi, R., & Chen, G. (2022). A risk assessment framework considering uncertainty for corrosion-induced natural gas pipeline accidents. *Journal of Loss Prevention in the Process Industries, 75*, 104718.

Li, Z., Huang, Z., Guo, L., Shan, L., Yu, G., Chong, Z., & Engineering, E. (2023). Cognitive knowledge graph generation for grid fault handling based on attention mechanism combined with multi-modal factor fusion. *111*, 108855.

Liu, Q., Li, F., Ng, K. K. H., Han, J., & Feng, S. (2025). Accident investigation via LLMs reasoning: HFACS-guided Chain-of-Thoughts enhance general aviation safety. *Expert Systems with Applications*, 126422.

Mahmood, Y., Chen, J., Yodo, N., & Huang, Y. (2024). Optimizing natural gas pipeline risk assessment using hybrid fuzzy bayesian networks and expert elicitation for effective decision-making strategies. *Gas Science and Engineering, 125*, 205283.

Metze, K., Morandin-Reis, R. C., Lorand-Metze, I., & Florindo, J. B. (2024). Bibliographic research with large language model ChatGPT-4: instability, hallucinations and sometimes alerts. *Clinics, 79*, 100409.

Pu, H., Yang, X., Li, J., & Guo, R. (2024). AutoRepo: A general framework for multimodal LLM-based automated construction reporting. *Expert Systems with Applications, 255*, 124601.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv Preprint arXiv:1908.10084.*

Shekhar, A. C., Kimbrell, J., Saharan, A., Stebel, J., Ashley, E., & Abbott, E. E. (2025). Use of a large language model (LLM) for ambulance dispatch and triage. *The American Journal of Emergency Medicine, 89*, 27-29.

Viana, F. F. C. L., Casado, R. S. G. R., da Silva, L. B. L., Alencar, M. H., Ferreira, R. J. P., & de Almeida, A. T. (2022). A hybrid multicriteria decision model for selecting a portfolio of risk-based maintenance actions in natural gas pipelines. *Journal of Natural Gas Science and Engineering, 103*, 104655.

Wong, S., Zheng, C., Su, X., & Tang, Y. (2024). Construction contract risk identification based on knowledge-augmented language models. *Computers in Industry, 157-158*, 104082.

Xiao, R., Zayed, T., Meguid, M. A., & Sushama, L. (2024). Dynamic risk assessment of natural gas transmission pipelines with LSTM networks and historical failure data. *International Journal of Disaster Risk Reduction, 112*, 104771.

Zhang, J., Ruan, X., Si, H., & Wang, X. (2025). Dynamic hazard analysis on construction sites using knowledge graphs integrated with real-time information. *Automation in Construction, 170*, 105938.

Zou, Y., Shi, M., Chen, Z., Deng, Z., Lei, Z., Zeng, Z., & Zhou, W. (2025). ESGReveal: An LLM-based approach for extracting structured data from ESG reports. *Journal of Cleaner Production, 489*, 144572.