Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/AP07.20260468
Published: March 25, 2026

RAG-Enhanced Cognitive Intelligent Recommendation Agent for Self-Evolving Failure Recovery in Manufacturing System Operations

Liuyang Shan, Mulang Song and Yiyun (Cindy) Fei
School of Mechanical Engineering
Georgia Institute of Technology, Atlanta, GA, USA

Shu Wang
TE Connectivity, Middletown
Pennsylvania, USA

Roger J. Jiao
School of Mechanical Engineering
Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Manufacturing systems in the industry 4.0 era are becoming increasingly complex and large-scale, creating growing demand for efficient and reliable failure recovery systems. However, traditional manual failure recovery processes often suffer from cognitive overload, inconsistent recovery decisions, and limited scalability. To address these challenges, this paper proposes a retrieval augmented generation (RAG) enhanced failure recovery framework that integrates large language model (LLM) with hierarchical knowledge graph–based knowledge management. The proposed system organizes machine failure data into a structured knowledge graph and performs semantic retrieval over failure metadata and domain specific knowledge to generate recommended recovery solutions. The framework enables scalable and context-aware troubleshooting for manufacturing failure recovery.

Keywords
Large-language model, Knowledge graph, Retrieval-augmented generation, Failure recovery, Industry 4.0

1. Introduction
Manufacturing systems are increasingly characterized by high levels of automation, interconnectivity, and data-driven operation under the paradigm of Industry 4.0. While these advances significantly improve productivity and operational flexibility, they also introduce greater system complexity, making production and equipment failures inevitable and often costly. Such failures may result in unplanned downtime, yield loss, and potential safety risks, highlighting the importance of effective failure recovery mechanisms in modern manufacturing environments.

Central to failure recovery is the ability to analyze failure reports, identify plausible root causes, and determine appropriate corrective actions in a timely and context-aware manner. As manufacturing systems become more interconnected and technologically sophisticated, the efficiency and reliability of failure handling increasingly influence overall system resilience and operational performance.

1.1 Knowledge-Intensity of Failure Recovery

Failure recovery is inherently a knowledge-intensive and context dependent process. When a failure occurs, engineers must interpret inspection or diagnostic reports, understand machines and process specific conditions, retrieve relevant historical cases, and determine suitable corrective strategies. In practice, this process remains largely manual, and experience driven, relying heavily on individual expertise and ad hoc searches through historical records.

Although enterprises routinely archive failure reports and solution records in structured formats such as spreadsheets or databases, these records are often isolated and lack explicit semantic relationships linking failure phenomena, causes, and corrective actions. Furthermore, critical troubleshooting knowledge frequently exists as informal “tribal knowledge” held by experienced engineers, which is valuable but difficult to formalize and systematically reuse (Vidyaratne et al. 2025). As manufacturing complexity increases, such practices lead to inconsistent decision making, delayed responses, and limited scalability of troubleshooting processes.

1.2 Emerging AI Approaches for Failure Reasoning

Recent advances in knowledge representation and generative artificial intelligence offer new opportunities to address challenges mentioned in paragraph above. Knowledge graphs provide a structured mechanism to model failure-related knowledge, including failure modes, causal relationships, and corrective strategies, while enabling explicit semantic linking across heterogeneous reports.

When combined with RAG, knowledge graphs can support contextualized reasoning by grounding LLM outputs in domain specific evidence (Alenjareghi et al. 2025). RAG powered LLMs have demonstrated promising capabilities in industrial troubleshooting and anomaly handling by retrieving relevant historical knowledge and generating context-aware recommendations (Liu et al. 2024). In addition, recent studies indicate that LLM based agents equipped with retrieval mechanisms can dynamically gather diagnostic evidence from structured data sources, improving factual accuracy in root cause analysis tasks (Roy et al. 2024).

1.3 Challenges in Intelligent Failure Recovery

Despite these advances, several challenges remain in deploying intelligent failure recovery systems in real manufacturing environments. First, domain specific knowledge modeling is required to transform structured failure reports into semantically meaningful representations. This requires mapping report fields to a failure ontology and constructing a hierarchical failure knowledge graph that explicitly captures relationships among failure types, root causes, equipment contexts, and corrective actions. Second, effective decision support requires context-aware retrieval and reasoning over historical failure knowledge, which necessitates close integration between a RAG-based retrieval module and an LLM based solution generation component. Third, manufacturing environments continuously generate new failure reports and validated corrective actions. Therefore, a practical failure recovery system must support continual learning so that newly verified operational experience can be assimilated into the knowledge base and improve future decision-making. However, such evolution must remain controlled and traceable, ensuring that human expert judgment remains central to knowledge validation and that automated learning mechanisms do not compromise reliability or interpretability (Vidyaratne et al. 2025).

To address these challenges, this paper proposes a RAG-Enhanced Cognitive Intelligent Recommendation Agent for manufacturing failure recovery. The proposed framework transforms heterogeneous failure records into a structured domain specific failure ontology and hierarchical knowledge graph, enabling explicit modeling of relationships among failure modes, root causes, equipment contexts, and corrective actions. Building upon this structured representation, an RAG mechanism grounds LLM reasoning in relevant historical cases and knowledge subgraphs, enabling context-aware and evidence supported solution recommendations. Furthermore, the framework incorporates a self-evolving learning mechanism that continuously assimilates validated failure–solution pairs and operational feedback. These updates refine both the semantic retrieval process and the solution generation capability over time.

2. Relevant Work

2.1 Ontology Engineering and Hierarchical Knowledge Graph Construction

Knowledge graphs have become an important paradigm for representing structured industrial knowledge and enabling semantic integration across heterogeneous data sources. A key prerequisite for constructing domain knowledge graphs is ontology engineering, which defines domain entities, relations, constraints, and hierarchical taxonomies that guide knowledge acquisition and evolution. Ontology engineering methodologies emphasize that ontologies should function as evolving conceptual infrastructures that support iterative specification and refinement rather than static schemas (Kotis et al. 2020).

In manufacturing environments, structured ontologies have been developed to support knowledge reuse and standardization across design and production processes. For example, Chhim's team propose a product design and manufacturing ontology that connects design knowledge with manufacturing process knowledge for systematic knowledge reuse (Chhim et al. 2020). Such approaches highlight the importance of domain aligned conceptual structures for organizing engineering knowledge.

Practical frameworks for transforming industrial reports into knowledge graphs have also been explored. Yin's team present a knowledge graph construction framework for industrial safety reports using standardized report mapping and rule based triple generation stored in Neo4j (Yin et al. 2023). In safety critical environments, human-in-the-loop verification mechanisms are often introduced to ensure traceability and reliability of generated knowledge structures (Vidyaratne et al. 2025).

Earlier work on intelligent diagnostic systems also emphasized the importance of meta level coordination mechanisms for integrating heterogeneous knowledge sources. Jiao et al. proposed a meta system framework that manages multiple diagnostic modules through meta knowledge structures (Jiao et al. 1999). From this perspective, ontology driven hierarchical knowledge graphs can be viewed as a modern implementation of meta level knowledge organization. Knowledge graphs have also been applied to structure failure related knowledge in manufacturing environments, such as enriching failure analysis artifacts like FMEA through improved knowledge organization (Alenjareghi et al. 2025). However, many studies primarily focus on reasoning over pre-existing knowledge graphs, while less attention has been devoted to ontology guided pipelines that systematically transform operational failure reports into structured and evolving knowledge representations.

2.2 RAG and LLM Based Solution Recommendation

LLM has demonstrated strong capabilities in knowledge-intensive reasoning tasks, but their reliability is often limited by insufficient domain grounding and hallucination risks. RAG addresses this limitation by integrating neural retrieval mechanisms with generative models, allowing systems to access external knowledge sources during inference (Lewis et al. 2020).

In industrial contexts, LLM based approaches have been explored for troubleshooting and maintenance reasoning. Vidyaratne's team demonstrates that LLMs can generate troubleshooting trees from equipment manuals to support diagnostic reasoning (Vidyaratne et al. 2024). Similarly, LLM based agents equipped with retrieval tools have been proposed to gather diagnostic evidence from structured enterprise data sources for root-cause analysis (Roy et al. 2024).

Recent research also investigates how RAG frameworks can improve knowledge grounded recommendation systems. Multimodal RAG systems integrating textual and visual information have been shown to improve industrial question answering performance compared to text only retrieval (Riedler et al. 2024). Domain specific RAG pipelines have also been used to retrieve safety or failure knowledge from structured repositories to support troubleshooting recommendations (Alenjareghi et al. 2025).

Knowledge graphs have further been combined with LLM based reasoning. Liu's team propose a framework that integrates knowledge graph retrieval with LLM reasoning to support fault localization in aviation assembly systems (Liu et al. 2024). While these approaches demonstrate the value of grounding LLM reasoning in structured relational knowledge, many assume the existence of curated knowledge bases and do not address how operational failure reports can be continuously integrated and reused for troubleshooting support.

2.3 Knowledge Evolution and Continuous Learning

Industrial knowledge bases and knowledge graphs are inherently dynamic and must evolve as new operational data and expert insights become available. Polleres's team provides a comprehensive survey of knowledge evolution in open knowledge graphs, highlighting mechanisms such as temporal graph models and version-controlled knowledge management (Polleres et al. 2023).

In industrial intelligence systems, knowledge evolution often occurs through iterative integration of newly observed operational information. Multi-agent LLM workflows have been proposed to dynamically extract and integrate knowledge elements into evolving knowledge bases (Yang et al. 2025). Human expertise also plays a critical role in this process. Human-in-the-loop verification mechanisms are frequently used to validate extracted knowledge and capture tacit troubleshooting knowledge held by experienced engineers (Vidyaratne et al. 2025).

Knowledge graph evolution mechanisms have also been explored for industrial fault diagnosis. Han's team introduce reinforcement learning–based approaches to continuously enrich industrial fault diagnosis knowledge graphs with new fault cases and missing relations (Han et al. 2022). Beyond structural updates, continual improvement can also occur at the reasoning layer through techniques such as reinforcement learning from human feedback and iterative reasoning frameworks (Tadaka et al. 2024; Roy et al. 2024).

Despite these advances, existing studies typically focus on individual aspects of the problem, such as ontology design, knowledge graph construction, or LLM based reasoning. Practical manufacturing failure management systems, however, require an integrated framework that can simultaneously structure failure knowledge, retrieve relevant historical evidence, and continuously evolve through expert validation and feedback. Addressing this need motivates the integrated failure knowledge management framework proposed in this work.

3. Self-Evolving Machine Failure Recovery System Analysis and Design

3.1 Problem Context and Motivation

Manufacturing failure handling is often performed under time pressure and incomplete information. In many industrial environments, troubleshooting decisions rely heavily on the experience of engineers rather than systematically structured knowledge. Historical failure reports, operational records, and tacit engineering knowledge often remain fragmented, resulting in limited reuse of accumulated troubleshooting experience, as illustrated in Figure 1. To address these limitations, a structured knowledge management framework is required to organize heterogeneous failure knowledge, enable semantic retrieval across historical cases, and support continuous knowledge evolution.

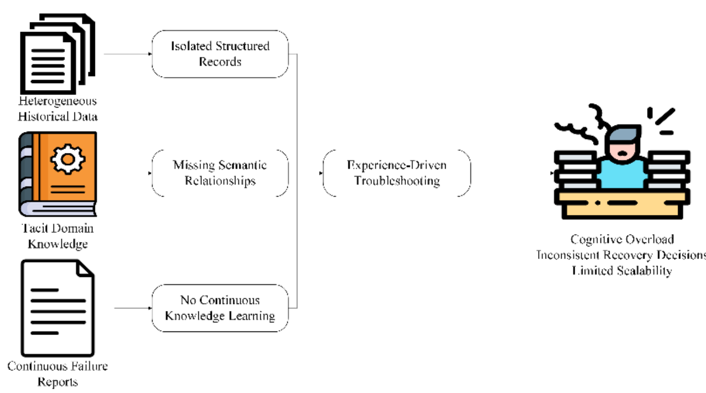

Figure 1. Illustration of challenges in traditional manufacturing failure handling

3.2 Knowledge Driven Failure Reasoning Architecture

Knowledge driven reasoning systems leverage accumulated historical experience to support decision making in complex and uncertain environments. In manufacturing failure handling, effective reasoning requires not only referencing historical cases but also governing how failure knowledge is represented, retrieved, validated, and incrementally evolved. From this perspective, intelligent failure recovery requires a meta level knowledge management architecture that regulates the lifecycle of failure knowledge across modeling, retrieval, and evolution stages.

Knowledge graphs provide a structured mechanism for representing relationships among failure symptoms, root causes, and corrective actions, supporting reasoning over interconnected industrial knowledge (Mao et al. 2024). At the same time, RAG enables LLM to generate recommendations grounded in retrieved external knowledge rather than relying solely on parametric reasoning (Lewis et al. 2020). Integrating knowledge graphs with LLM based reasoning further enables troubleshooting recommendations to be supported by structured relational evidence (Liu et al. 2024).

Based on these concepts, the proposed system establishes a knowledge governing reasoning architecture integrating ontology driven knowledge representation, embedding based retrieval, and LLM based solution generation. As illustrated in Figure 2, the architecture organizes the lifecycle of failure knowledge through coordinated modules including ontology construction, knowledge graph storage, semantic retrieval, and solution generation, enabling evidence grounded reasoning and controlled knowledge evolution.

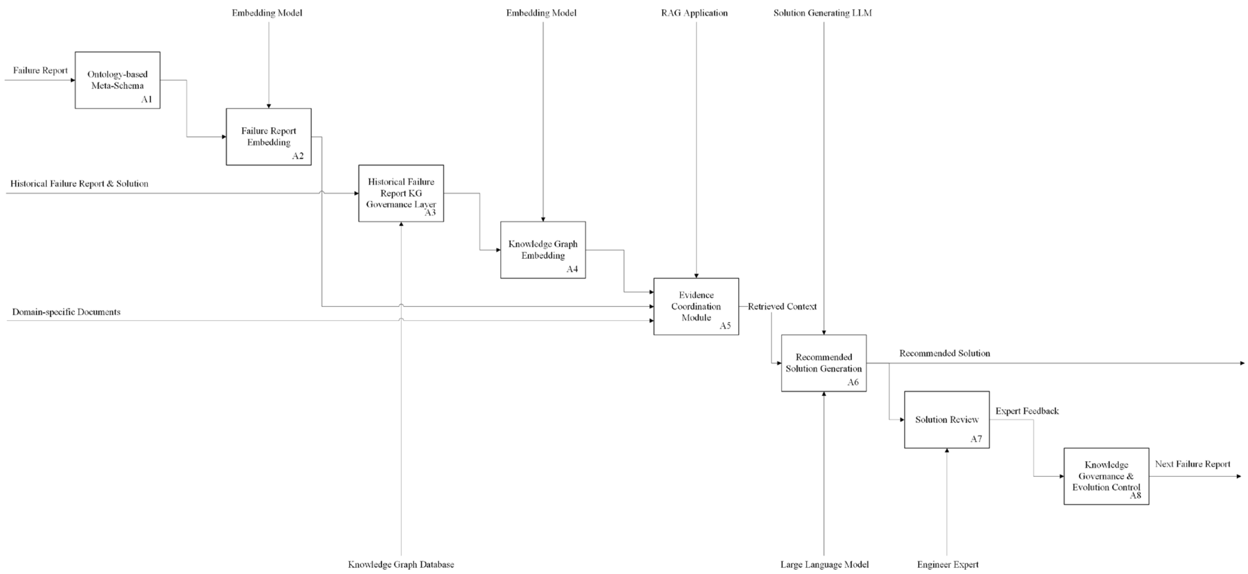

Figure 2. Knowledge driven failure reasoning architecture integrating knowledge graphs, embedding based retrieval, and LLM based solution generation

3.3 Operational Reasoning and Continuous Learning Workflow

The operational workflow of the proposed system is illustrated in Figure 3. When a new failure report is submitted, it is first mapped to the ontology based representation and embedded into the shared semantic space. The embedded report is then matched against historical failure cases and knowledge graph entities to retrieve contextually relevant knowledge fragments through a RAG based retrieval module.

An evidence coordination mechanism aggregates retrieved historical cases, relational subgraphs, and domain-specific documents into a structured contextual package that constrains the generation process. This contextual evidence is supplied to an LLM based solution generator, which synthesizes a recommended corrective action grounded in retrieved knowledge rather than relying solely on parametric reasoning.

The generated recommendation is subsequently subjected to expert validation. Human engineers review the suggested solution, providing approval, refinement, or rejection based on practical operational considerations. The validated solution, together with expert feedback and the observed outcome of the corrective action, is then forwarded to the knowledge governance and evolution module.

This module regulates how newly generated failure–solution pairs are integrated into the knowledge graph and determines whether retrieval representations or reasoning components require updating. Through this controlled human-in-the-loop feedback mechanism, the system continuously accumulates operational knowledge while maintaining traceability, interpretability, and domain reliability. Over time, the knowledge graph evolves to reflect newly discovered failure patterns and corrective strategies, enabling progressively more accurate and consistent troubleshooting recommendations.

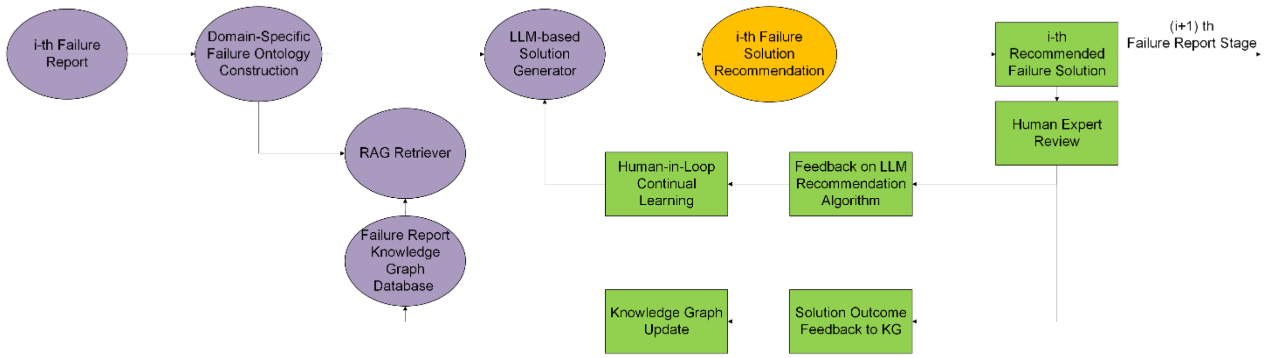

Figure 3. Operational workflow and continuous learning mechanism of the proposed failure reasoning system

4. Hierarchical Knowledge Graph Construction

4.1 Ontology Governed Failure Knowledge Representation

Effective failure knowledge management requires not only transforming structured failure records into machine readable formats, but also establishing a governing structure that defines how failure knowledge is organized, linked, validated, and incrementally evolved. In the proposed framework, hierarchical knowledge graph construction is treated as a structured knowledge representation mechanism guided by a domain specific ontology.

Within this framework, ontology defined semantic constraints regulate how heterogeneous failure information is represented and connected across cases. Each failure report is modeled as a central knowledge entity, while descriptive attributes and contextual information are represented as linked sub-entities. This hierarchical structure enables explicit semantic relationships between failure cases and relevant contextual elements such as machines, components, temporal information, textual descriptions, and solution records.

4.2 Failure Knowledge Graph Schema Design

As illustrated in Figure 4, the failure knowledge graph schema defines each failure report as a core node labeled Failure_Report. This node is connected to attribute nodes through predefined semantic relations. Specifically, the relation hasMachine links the failure report to the corresponding machine identifier (Machine_Number), while hasFailurePart connects the report to the failed component identifier (Failure_Part_Number).

Additional contextual attributes are represented through relations such as hasDescription, which links the report to the textual failure description (Description), and hasFailureDate, which records the occurrence time (Failure_Date). When corrective actions are documented, the relations hasSolution and hasFixedDate connect the failure report to the corresponding repair action (Solution) and repair completion timestamp (Fixed_Date).

The schema is designed to support incomplete failure records frequently encountered in industrial datasets. Optional attributes such as solution and repair time nodes allow partially documented failure cases to be incorporated without violating schema constraints, enabling practical integration of heterogeneous operational data.

Figure 4 also illustrates an example subgraph generated from a failure report. The central Failure_Report node connects to contextual attributes including machine identifier, failure component, description, occurrence time, and corrective solution. This example demonstrates how structured failure records are transformed into semantically connected graph representations for downstream retrieval and reasoning.

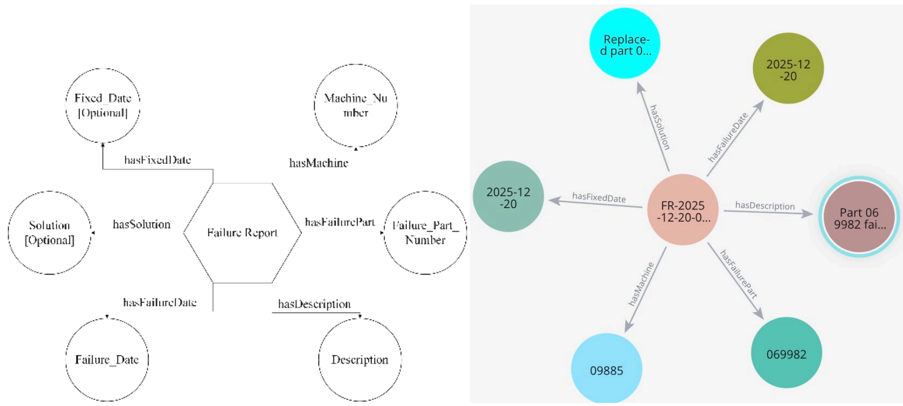

Figure 4. Failure report knowledge graph schema & a failure report knowledge graph example

4.3 Automated Failure Report–to–Graph Transformation Pipeline

To operationalize this schema in industrial environments, the system implements an automated transformation pipeline that converts structured failure reports into knowledge graph entities and relations. Historical failure reports and maintenance records are typically stored in structured spreadsheet formats or database tables.

As shown in Figure 5, failure report files are first parsed using a Python-based data extraction script. Column headers are interpreted according to the predefined ontology schema, and corresponding entity–relation–entity triples are generated automatically. The transformation script dynamically constructs Cypher queries that create or update nodes and relationships within a Neo4j graph database through the official driver interface. After the pipeline, an excel-based failure report will be transformed into a neo4j-based knowledge graph, as shown in Figure 4.

This automated pipeline ensures that newly ingested failure records follow the ontology defined schema and relational constraints. Consequently, knowledge graph expansion becomes a consistent and repeatable data ingestion process rather than an ad hoc manual operation.

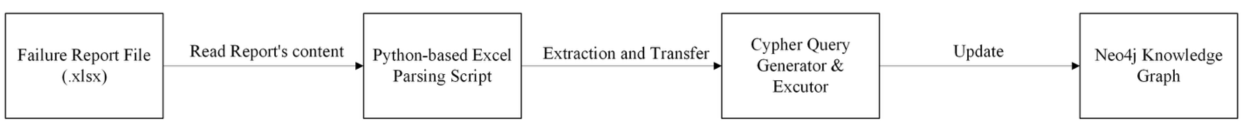

Figure 5. Automated excel-to-knowledge graph construction pipeline

5. Context-Aware Solution Generation & Recommendation

5.1 Multi-Source Knowledge Integration

After constructing the hierarchical failure knowledge graph, the system generates solution recommendations for newly observed failure reports. In practical manufacturing environments, effective troubleshooting often requires consulting not only historical failure cases but also domain specific technical documentation, such as maintenance manuals, troubleshooting guidelines, and engineering records.

To enable context-aware reasoning, the proposed system integrates two complementary knowledge sources: the hierarchical failure knowledge graph database and domain specific technical documents. These knowledge sources collectively provide both empirical failure–solution experience and formal engineering knowledge, forming the foundation of the retrieval augmented solution generation framework.

5.2 Structured Representation and Embedding

As illustrated in Figure 6, both the hierarchical knowledge graph and domain specific documents are first transformed into structured textual representations. For the knowledge graph, each Failure_Report entity and its associated attribute nodes are serialized into structured text that describes machine context, failure components, descriptive information, temporal attributes, and solution records. Similarly, domain specific documents are processed by extracting relevant passages and organizing them into structured textual segments. A newly submitted failure report is also converted into the same structured format.

All structured texts are then processed using a shared embedding model, which converts historical failure cases, domain documents, and the new failure report into vector representations. By mapping heterogeneous knowledge sources into a unified embedding space, the system enables consistent semantic comparison across different types of industrial knowledge.

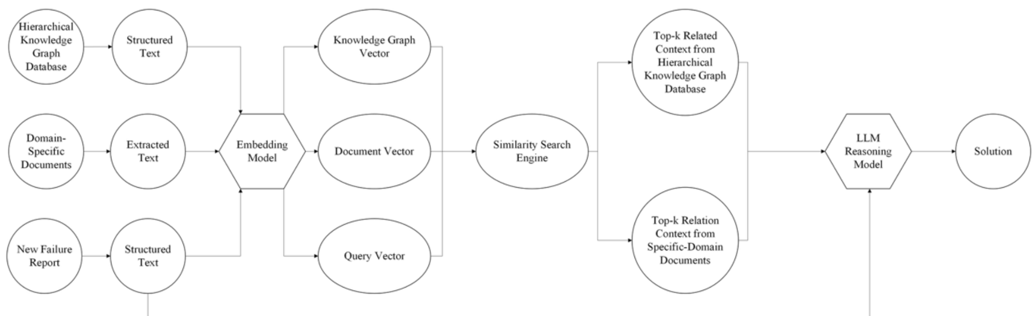

Figure 6. Multi-source embedding & retrieval

5.3 Unified Similarity Based Context Retrieval & Retrieval Augmented Solution Generation

Based on the embedded representations, a similarity search engine (as shown in Figure 6) performs unified retrieval over both the embedded hierarchical knowledge graph database and the embedded domain document corpus.

Given the query vector derived from the new failure report, the retrieval module identifies the most relevant contextual information using similarity matching. In the current implementation, the system retrieves the top historical failure reports from the knowledge graph and the top document passages from the domain specific document corpus. The retrieved failure cases provide concrete historical repair precedents, while the document passages supply complementary technical explanations and procedural knowledge that may not be explicitly stored within the knowledge graph.

As illustrated in Figure 7, the retrieved contextual information is assembled into a structured prompt that serves as grounding context for an LLM based solution generator. Conditioned on both the new failure report description and the retrieved contextual knowledge, the language model synthesizes a corrective action recommendation tailored to the current failure case. This RAG mechanism ensures that generated solutions are grounded in verified historical experience and domain specific technical knowledge rather than relying solely on free-form generative reasoning.

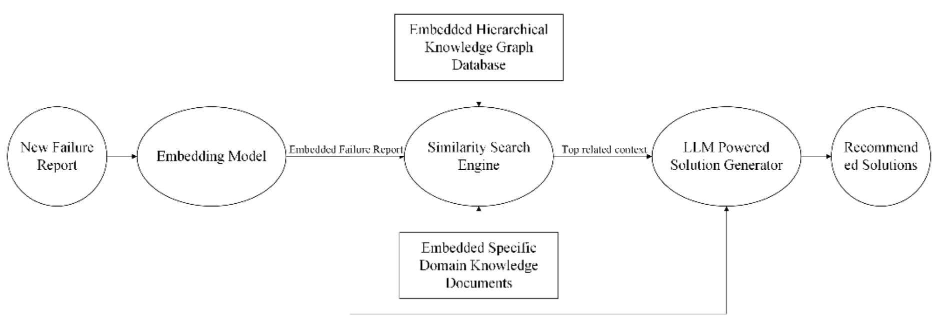

Figure 7. Retrieval augmented solution generation

6. Self-Evolving from Feedback

Once a solution has been generated for a newly observed failure case, expert evaluation is incorporated to enable continuous improvement of both the knowledge base and the solution generation model. As illustrated in Figure 8, for the i-th failure report case, the LLM based solution generator first produces a recommended solution. This recommendation is then examined by a domain expert, who provides structured feedback indicating the validity, effectiveness, and potential modifications of the proposed solution.

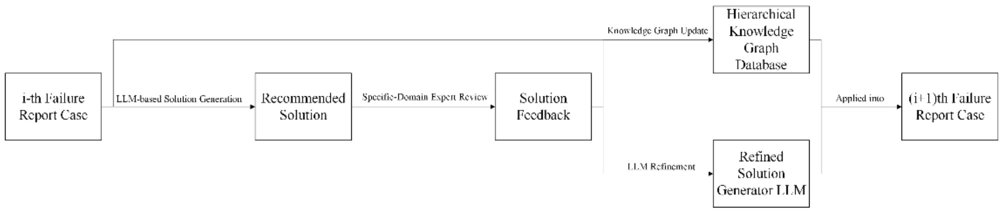

Figure 8. Self-evolving loop pipeline

The collected solution feedback serves two parallel purposes. Firstly, validated failure–solution pairs are appended to the hierarchical knowledge graph database through a knowledge graph update process. By continuously inserting newly confirmed cases, the knowledge graph incrementally expands its coverage of failure modes, machine contexts, and corresponding solution strategies. This ensures that future retrieval operations can reference an increasingly rich and up-to-date historical case repository. Secondly, the same expert feedback is used to refine the LLM based solution generator through feedback driven model refinement. Rather than retraining the model from scratch, lightweight fine-tuning or reward-based adjustment is applied to bias the generator toward producing solutions that better align with expert knowledge and practical operational constraints. This refinement process progressively improves the quality and reliability of generated recommendations over successive cases.

Through the joint evolution of the hierarchical knowledge graph and the refined solution generator, the updated components are subsequently applied to the (i+1)-th failure report case. In this manner, the proposed system realizes a closed-loop learning mechanism that continuously accumulates domain knowledge, adapts its reasoning behavior based on expert feedback, and incrementally enhances decision support performance in real industrial failure handling scenarios.

7. Case Study of Machine Failure Analysis

To validate the proposed meta level knowledge management architecture in a real manufacturing environment, a Machine Failure Analysis project was implemented within an operational production system. Due to data sensitivity and privacy, the detailed machine failure data and recommended solution information can't be shared publicly, however based on manufacture center engineer's estimation, this project will save their 3900 engineer hours per year.

7.1 Ontology Constrained Knowledge Interaction

As shown in Figure 9, the system provides a search and management interface through which engineers can access historical failure reports and maintain the knowledge base. All report entries and modifications are constrained by the predefined ontology schema. Key attributes such as part number, mold number, machine identifier, and report date correspond directly to entity types and relations defined in the hierarchical failure knowledge graph.

By enforcing ontology aligned data entry and modification, the interface functions as a controlled gateway to the knowledge governance layer. Newly inserted or updated reports therefore remain semantically consistent with the knowledge graph structure, ensuring that operational data can be reliably integrated into the evolving failure knowledge base.

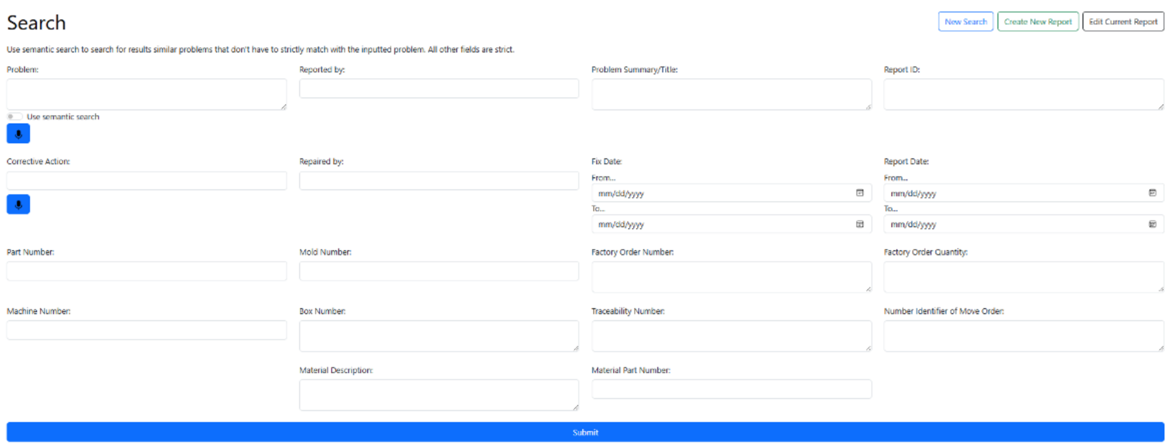

Figure 9. Machine failure analysis search interface for database maintenance and relative report retrieval

7.2 Evidence Grounded Retrieval and Solution Generation

When a new failure report is submitted, the system performs semantic retrieval over the hierarchical failure knowledge graph and associated domain documents. As illustrated in Figure 10, the retrieval module returns a ranked list of historically related failure cases based on embedding similarity within a unified representation space.

The retrieved cases contain contextual attributes such as machine context, failed components, timestamps, and previously validated solutions. These structured evidence blocks are assembled into a prompt for the LLM based generator, enabling the model to produce context-aware troubleshooting recommendations grounded in historical operational knowledge.

An example of the generated solution recommendation is shown in Figure 11. Rather than returning categorical fault labels, the system synthesizes narrative corrective guidance supported by retrieved evidence. Engineers can inspect the referenced historical cases and edit the generated report if necessary, maintaining human oversight in the troubleshooting process.

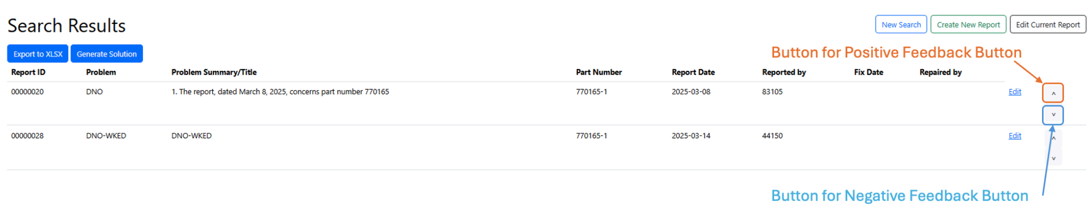

Figure 10. Semantic search for top-related historical reports with positive/negative feedback button

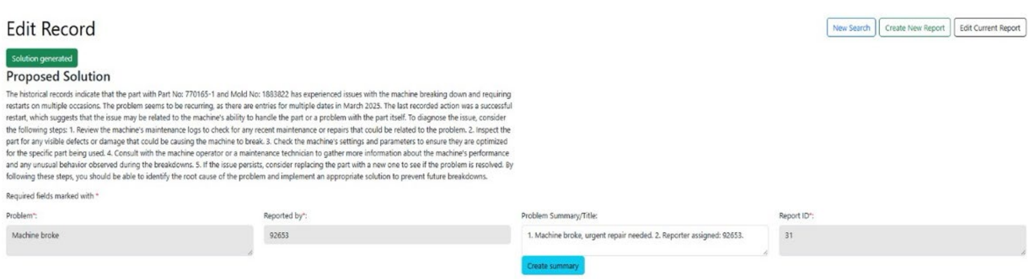

Figure 11. Recommended troubleshooting solution for a specific machine failure report

7.3 Expert Validation and Knowledge Evolution

Expert review acts as a validation gate within the knowledge lifecycle. Only after a recommended solution is confirmed by a domain expert is the corresponding failure–solution association incorporated into the knowledge graph. The validated case is then ingested through the ontology aligned report-to-graph transformation pipeline described earlier, ensuring that knowledge graph expansion remains schema consistent and traceable.

In addition to solution validation, the system records engineer feedback on the relevance of retrieved historical cases through the interface shown in Figure 10. Positive and negative feedback signals are logged together with query context and document identifiers. These records are periodically used to refine the dense retrieval model through offline training based on pairwise ranking objectives.

To preserve system stability, updated retrieval models are deployed only when evaluation results demonstrate improved retrieval quality on validation datasets or through expert assessment. Importantly, this feedback driven refinement is restricted to the retrieval ranking layer, while the ontology schema and LLM based solution generator remain unchanged. This architectural separation ensures that adaptive learning improves retrieval relevance without compromising structural knowledge integrity.

7.4 Meta Level Lifecycle Validation

The deployment demonstrates that the proposed framework functions as a governed knowledge lifecycle within a real manufacturing environment. The ontology driven interaction interface ensures that failure reports are consistently structured and integrated into the hierarchical knowledge graph. Embedding based retrieval enables engineers to identify relevant historical cases and contextual evidence, supporting evidence grounded troubleshooting recommendations generated by the LLM based module.

The expert validation mechanism further ensures that newly generated solutions are verified before incorporation into the knowledge base, maintaining traceability and reliability of the evolving knowledge graph. In addition, the feedback logging mechanism provides a practical means for refining the semantic retrieval component based on engineer interactions. Through repeated operational cycles of report submission, retrieval, solution generation, expert validation, and feedback driven refinement, the system demonstrates its capability to support structured failure knowledge management and continuously improve retrieval relevance in practical manufacturing failure analysis workflows.

8. Conclusion

This paper presents a RAG-enhanced intelligent failure recovery system framework for cognitive manufacturing decision support. The proposed approach has been applied through a real-world machine failure analysis project in an industrial manufacturing environment, validated its ability to retrieve relevant historical failure cases and generate actionable solution recommendations for newly reported failures.

A real-world machine failure analysis deployment demonstrates that the proposed framework can effectively assist engineers in retrieving relevant historical cases and generating corrective recommendations. Over successive operational cycles, the system transforms historical failure records into a continuously evolving knowledge infrastructure for manufacturing decision support.

References

Alenjareghi, M. J., Keivanpour, S., Chinniah, Y. A. and Jocelyn, S., LLM-driven FMEA for safe human-robot collaboration in disassembly, *IEEE International Conference on Human-Machine Systems*, 2025.

Chhim, P., Chinnam, R. B. and Sadawi, N., Product design and manufacturing process based ontology for manufacturing knowledge reuse, *Journal of Intelligent Manufacturing*, vol. 30, 2019.

Han, H., Wang, J., Wang, X. and Chen, S., Construction and evolution of fault diagnosis knowledge graph in industrial process, *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022.

Jiao, J., Zhang, W., Zhao, Z. et al., An integrated intelligent approach to process diagnosis in process industries, *International Journal of Production Research*, vol. 37, no. 11, 1999.

Kotis, K. I., Vouros, G. A. and Spiliotopoulos, D., Ontology engineering methodologies for the evolution of living and reused ontologies: status, trends, findings and recommendations, *The Knowledge Engineering Review*, vol. 35, 2020.

Lewis, P., Perez, E., Piktus, A. et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, *Advances in Neural Information Processing Systems*, vol. 33, 2020.
Liu, P., Qian, L., Zhao, X. and Tao, B., Joint knowledge graph and large language model for fault diagnosis and its application in aviation assembly, *IEEE Transactions on Industrial Informatics*, vol. 20, no. 6, 2024.
Mao, Z., Wang, H., Jiang, B., Xu, J. and Guo, H., Graph convolutional neural network for intelligent fault diagnosis of machines via knowledge graph, *IEEE Transactions on Industrial Informatics*, vol. 20, no. 5, 2024.
Polleres, A., Pernisch, R., Bonifati, A., Dell'Aglio, D., Dobriy, D., Dumbrava, S., Etcheverry, L., Ferranti, N., Hose, K., Jiménez-Ruiz, E., Lissandrini, M., Scherp, A., Tommasini, R. and Eachs, J., How does knowledge evolve in open knowledge graphs?, *Transactions on Graph Data and Knowledge*, vol. 1, no. 1, 2023.
Riedler, M. and Langer, S., Beyond text: optimizing RAG with multimodal inputs for industrial applications, *arXiv preprint*, 2024.
Roy, D., Zhang, X., Bhave, R., Bansal, C., Las-Casas, P., Fonseca, R. and Rajmohan, S., Exploring LLM-based agents for root cause analysis, *ACM International Conference on the Foundations of Software Engineering*, 2024.
Tadaka, K. B., Cassandro, R., Zhang, W. and Li, Z. S., ChatReliability: a customized copilot based on LLM to facilitate design for reliability, *IEEE Technology & Engineering Management Conference*, 2024.
Vidyaratne, L., Lee, X. Y., Kumar, A., Watanabe, T., Farahat, A. and Gupta, C., Generating troubleshooting trees for industrial equipment using large language models (LLM), *IEEE International Conference on Prognostics and Health Management*, 2024.
Yang, Y., Wang, D. D. L., Chiclana, F. and Chen, Z. S., A multi-agent LLM framework for severity classification of complaint events: probabilistic reasoning with scene uncertainty, *Advanced Engineering Informatics*, vol. 69, 2026.
Yin, Z., Shi, L., Yuan, Y. et al., A study on a knowledge graph construction method of safety reports for process industries, *Processes*, vol. 11, no. 1, 2023.

Biographies

Liuyang Shan is a Mechanical Engineering student from Georgia Institute of Technology in Atlanta, GA, USA. He has participated several real-world industrial projects related to AI powered manufacturing and automation process. More info is available in his LinkedIn: https://www.linkedin.com/in/liuyang-shan-898526335/

Shu Wang is a applied machine learning engineer at TE Connectivity. His expertise spans in industrial AI and data analytics, engineering design decision-making, human-automation interaction, and operations optimization. He holds a Ph.D. in Mechanical Engineering from Georgia Tech, along with dual master's degrees in Mechanical Engineering and Electrical and Computer Engineering. He earned his bachelor's degree in Electrical engineering from Wuhan University. More info about his research: https://scholar.google.com/citations?user=_R1ldZkAAAAJ&hl=en

Mulang Song is a Ph.D. in the School of Mechanical Engineering at the Georgia Institute of Technology (USA). He received both his B.S. and M.S. degrees in Mechanical Engineering with a focus on Computer Science, also from Georgia Tech. He has led and contributed to engineering projects through direct collaboration with industry, applying research in real-world production environments. More information about his research: https://scholar.google.com/citations?user=jLL4nycAAAAJ

Yiyun (Cindy) Fei is a Ph.D. candidate in Mechanical Engineering at Georgia Institute of Technology, Georgia, USA, where she also earned her bachelor's and master's degree in the same field. Her research focuses on the modeling, analysis, and optimization of complex operational systems to support data-driven design and decision-making. Yiyun worked as a Manufacturing and Process Development Engineer at TE Connectivity, where she led and contributed to multiple projects involving AI-based machine vision, process automation, and discrete event simulation of manufacturing systems. More information is available on her LinkedIn profile: www.linkedin.com/in/yiyun-fei

Roger Jiao is the editor-in-chief of Journal of Engineering Design and an associate professor of mechanical and industrial engineering at Georgia Tech, USA. Prior to joining the School of Mechanical Engineering at Georgia Tech in December 2008, he was an Assistant Professor and then Associate Professor in the School of Mechanical and Aerospace Engineering at Nanyang Technological University, Singapore. Before his career in Singapore, he was a Visiting Scholar in the Department of Industrial Engineering and Engineering Management at Hong Kong University of Science and Technology from 1998 to 1999. From 1993 to 1994, he was a Lecturer of Industrial Engineering in the School of Management at Tianjin University, China, and from 1988 to 1990, he worked as an Associate Lecturer in the Department of Industrial Design at Tianjin University of Science and Technology, China. More info about his research: https://scholar.google.com/citations?user=9yikEHAAAAAJ&hl=en&oi=ao.