Systems & Control Transactions

**Original Research Article**
*Peer Reviewed Conference Proceeding*
ESCAPE 36 - European Symposium on Computer Aided Process Engineering
Sheffield, UK. 21-24 June 2026

PSE PRESS

# A Large Language Model Enhanced Fault Diagnosis Framework for Chemical Processes

**Jingkang Liang[a], Gürkan Sin[a]***

[a] Process and Systems Engineering Center (PROSYS), Department of Chemical and Biochemical Engineering, Technical University of Denmark (DTU), 2800 Kgs. Lyngby, Denmark
* Corresponding Author: gsi@kt.dtu.dk

## ABSTRACT

Fault diagnosis is essential for ensuring safety and efficiency in chemical process industries. Conventional diagnostic systems often generate raw numerical outputs that require extensive human interpretation, increasing the operator's workload and slowing decision-making during abnormal events. To overcome these limitations, this work introduces a model context protocol (MCP)-integrated fault diagnosis framework, where a Large Language Model (LLM) functions as the MCP client, coordinating multiple diagnostic tools through a unified protocol. Within the proposed framework, the LLM interacts with specialized diagnostic tools, including a convolutional neural network-based fault diagnosis model and an ensemble-based variant for uncertainty-aware analysis. The LLM synthesizes the outputs of these tools and generates operator-oriented natural-language reports that summarize diagnostic results and explicitly communicate uncertainty, thereby supporting more transparent and efficient decision-making.A benchmarking study based on the Tennessee Eastman (TE) process is conducted to evaluate the framework using multiple LLMs under identical settings. The evaluation assesses diagnostic accuracy, tool adherence, and operator report quality using an LLM-as-a-judge methodology. Experimental results show that LLM performance varies significantly across models, and that reliable tool calling and uncertainty-aware reasoning are more critical than model size for effective fault diagnosis assistance.Overall, the proposed framework demonstrates strong potential for enhancing fault diagnosis workflows in chemical processes by improving usability, interpretability, and decision-making efficiency without modifying existing diagnostic algorithms.

**Keywords**: Artificial Intelligence, Fault Detection, Large Language Model

## 1. INTRODUCTION

Fault diagnosis plays an important role in chemical process industries, where undetected or misdiagnosed faults can lead to severe safety incidents, environmental damage, and significant economic losses [1]. To address these challenges, machine learning and deep learning techniques have been widely applied for fault detection and diagnosis [2], enabling earlier identification of abnormal operating conditions and improving overall process safety and reliability [3, 4]. Despite their performance, most existing fault diagnosis systems remain algorithm-centric, providing outputs in the form of numerical scores or fault probabilities [5]. Such results often lack intuitive, human-friendly explanations, increasing the cognitive workload of operators and potentially slowing down decision-making during abnormal events.

In parallel, recent advances in natural language processing have led to the rapid development of large language models (LLMs) based on transformer architectures [6]. These models demonstrate strong capabilities in reasoning, contextual understanding, and natural-language generation [7]. Beyond standalone text generation, LLMs have evolved into agentic systems, where an LLM can autonomously select and invoke external tools, interact with real-world applications, and maintain contextual memory to support more adaptive and personalized interactions [8]. In chemical engineering, LLMs have begun to emerge as promising tools for different tasks. Caccavale et al. [9] introduced ChatGMP, an LLM-based chatbot to support student learning in chemical engineering education. Liang et al. [10] leveraged LLM agent in

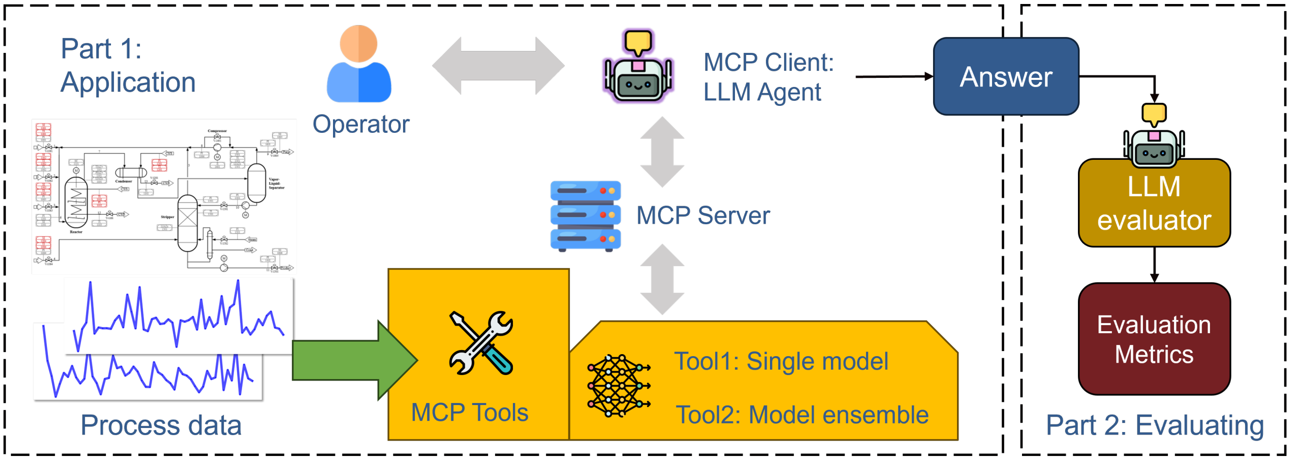


**Figure 1.** The proposed LLM-enhanced fault diagnosis framework with Model Context Protocol (MCP)

chemical simulations for simulation flowsheet analysis and synthesis. Zeng et al. [11] demonstrate an LLM-guided multi-agent optimization system that outperforms traditional methods on hydrodealkylation optimization . In fault diagnosis, FaultExplainer [12] demonstrates the potential of large language models to enhance fault diagnosis systems by generating interpretable, human-readable explanations alongside data-driven diagnostic results.

Motivated by the reasoning and tool-interaction capabilities of LLM agents, this work proposes a framework that integrates LLMs into chemical process fault diagnosis to enhance usability and decision support. Operators can query the system using natural language, while the LLM agent autonomously selects and invokes appropriate diagnostic tools using model context protocol (MCP), evaluates confidence in the results, and synthesizes the outputs into interpretable, operator-oriented explanations. By combining the reliability of established fault diagnosis algorithms with the flexibility and expressiveness of LLM agents, the proposed framework aims to reduce operator workload, improve situational awareness, and support faster and more informed decision-making during abnormal operating conditions.

In addition to proposing the framework, this work conducts a systematic benchmarking study using a dedicated evaluation dataset to compare the performance of different LLMs in terms of diagnostic accuracy, tool adherence and report quality. This study provides practical insights into LLM selection for fault diagnosis tasks that require both tool calling and operator-oriented report generation.

The main contributions of this paper are summarized as follows:

1) An MCP-based fault diagnosis framework is proposed, in which an LLM agent selects diagnostic tools based on prediction uncertainty and generate user-friendly report.
2) A structured evaluation experiment for LLM agents in fault diagnosis is carried out, enabling a comparative study of different LLMs under fair settings.

The remainder of this paper is organized as follows. Section 2 introduces the methodologies used in the proposed framework, including the CNN-based fault diagnosis algorithm, the MCP tools, and the LLM-as-a-judge evaluation strategy. Section 3 presents the experimental study of the proposed framework under different language models, using the Tennessee Eastman Process as a benchmarking dataset. Section 4 concludes the paper and outlines directions for future work.

## 2. METHODOLOGIES

### 2.1 Proposed workflow

As shown in Figure 1, the proposed framework defines an interaction between the operator, the LLM agent, the MCP server, diagnostic tools, and the chemical process. The operator interacts with the system by submitting natural-language queries to the LLM agent, which acts as an MCP client. The chemical process generates process data that are accessed by fault diagnosis algorithms exposed as MCP tools. These tools are managed by an MCP server, which provides standardized interfaces for tool discovery and invocation to the LLM. Upon receiving a query, the LLM agent interprets the user intent, selects appropriate diagnostic tool(s) via the MCP server, and invokes them using the relevant process data. The MCP tools perform fault diagnosis and return structured results to the LLM agent. The LLM agent aggregates and interprets the tool outputs and generates a user-facing fault diagnosis response for the operator.

In addition, an evaluation module is incorporated, where the generated responses are assessed using an LLM-as-a-judge [13] approach to compute multiple evaluation metrics. This evaluation provides insights into the usefulness, reliability, and decision-support quality of the LLM agent.

The following sections describe the diagnostic tools, the LLM tool-calling strategy, and the evaluation methodology, which together form the key components of this work.

## 2.2 Fault diagnosis algorithms

In this work, a multi-scale convolutional neural network (MSCNN) [14] is adopted as the core fault diagnosis algorithm due to its demonstrated effectiveness in previous fault diagnosis studies. Two variants are employed: a single MSCNN model and an ensemble-based MSCNN approach designed to provide more robust predictions.

The single MSCNN model is trained using the training dataset of the Tennessee Eastman Process. For the ensemble approach, multiple MSCNN models with identical network architectures and hyperparameters are trained independently. Model diversity is introduced through random initialization of model parameters and by training each model on different subsets of the training data. The final ensemble prediction is obtained by aggregating the outputs of all individual models, resulting in improved robustness against uncertainty in individual model predictions.

Based on prior experimental results, the ensemble-based MSCNN achieves an overall diagnostic accuracy of 73.67% in difficult faults, outperforming the single-model MSCNN, which attains an accuracy of 68.04%. These results demonstrate the advantage of the ensemble approach in providing more reliable fault diagnosis outcomes for subsequent interpretation by the LLM agent.

## 2.3 MCP tools

By using the Model Context Protocol (MCP) [15], a standardized protocol designed to expose external tools and their interfaces to large language models, the LLM is provided with structured contextual information about the available tools, including their functionality, required input parameters, and expected outputs. The tools are defined in a MCP server, the LLM agent, is a MCP client that can connect to the MCP server and get the available tools. MCP enables LLMs to interact with domain-specific software in a controlled and interpretable manner without requiring direct access to the underlying code or execution environment.

With this information, the LLM can reason about when and how to invoke a tool to solve tasks that require interaction with external software or applications. In this way, MCP serves as a standardized interface between the LLM and domain-specific tools, significantly simplifying tool invocation and integration. In this work, two diagnostic tools are exposed to the LLM through MCP, along with their corresponding descriptions:

**Predict_sample_single_model():** Predicts the fault type and associated probabilities for a given sample index using a single diagnostic model.

**Predict_sample_ensemble_with_plot():** Predicts fault types and probabilities using a model ensemble and generates a probability distribution plot across fault classes.

By invoking these tools, the LLM gains access to the fault diagnosis results produced by the underlying algorithms. The LLM then interprets, summarizes, and presents these results in a human-readable form, enabling operators to understand both the predicted fault and the associated uncertainty.

## 2.4 Agent Reasoning and Response Strategy

A rule-based tool selection logic is defined to guide the reasoning behavior of the LLM agent. Let $p_{\max}$ denote the maximum fault probability predicted by the single-model fault diagnosis algorithm. The tool selection strategy is formulated as follows:

- If $p_{\max} \geq 0.95$, the LLM agent directly uses the output of the single-model diagnosis.
- Otherwise, the LLM agent invokes the ensemble-based diagnosis tool and analyzes alternative faults, as well as plotting the result to the operator.

The rationale behind this strategy is to balance efficiency and robustness in operator-facing fault diagnosis. When the prediction confidence is high, the LLM agent is expected to provide a concise and confident report that focuses on the primary fault, thereby minimizing unnecessary computation and cognitive load. In contrast, when the prediction confidence is low, the LLM agent prioritizes robustness and transparency by incorporating ensemble results into the report. In such cases, the agent presents the most likely fault together with plausible alternative faults and their corresponding vote counts, explicitly communicating diagnostic uncertainty to the operator. This design supports efficient yet transparent decision-making in uncertain fault conditions.

## 2.5 LLM-as-a-judge

To evaluate the quality of the responses generated by the LLM agent, human feedback is generally regarded as the most reliable approach, since the goal of the system is to support human operators. However, for large-scale and repeated evaluations, relying on human assessment can be time-consuming, costly, and inconsistent. As an alternative, this work adopts an automated evaluation strategy known as LLM-as-a-judge [13], where a well-defined LLM is used to assess the outputs of another LLM.

The evaluation workflow using LLM-as-a-judge is illustrated as follows. First, a user submits a query to the LLM agent, which then performs reasoning and tool invocation to complete the fault diagnosis task and generate a final response. All intermediate information, including the user query, tool calls, and the final answer, is logged and provided as input to the LLM evaluator. In addition, auxiliary information from the evaluation dataset—such as the true fault label and the predefined uncertainty level of the sample—is included. A system prompt is also supplied to guide the judge LLM on the evaluation criteria and scoring procedure.

Based on the provided information, the judge LLM performs its own reasoning and assigns scores according to the specified metrics. This process closely resembles human evaluation, while offering improved efficiency and consistency across repeated experiments. The specific evaluation metrics used in this study are detailed in the following section.

## 3. EXPERIMENTAL STUDIES

### 3.1 Tennessee Eastman Process Dataset

The Tennessee Eastman process is used as a case study for the proposed LLM-enhanced fault diagnosis task. It is used to simulate fault scenarios in a chemical process where an operator queries an LLM agent to analyze process data, invoke diagnostic algorithms, and generate fault diagnosis reports. The TE process is a widely used benchmark in the literature for fault detection and fault diagnosis in chemical processes. In this work, its usage is extended to evaluate how LLMs can enhance fault diagnosis workflows through tool calling and natural-language interaction.

The dataset consists of simulation running with 21 different fault types [16]. In total 52 process variables are input to the algorithm. The output of the algorithm is typically the fault number with the highest probability, which is usually regarded as the predicted fault. However, in the uncertainty aware method, we take the probability distribution of all faults, so that we can see how confident it is for the model to predict the primary fault. The training, validation and test sets are separated by the ratio of 6:2:2, and the test set will contribute to the LLM evaluation dataset in section 3.2.

### 3.2 LLM-fault diagnosis evaluation dataset

To evaluate the behavior of the LLM agent in a controlled and quantitative manner, an evaluation dataset with ground-truth labels is constructed. This dataset enables systematic assessment of the LLM's reasoning, tool usage, and response quality under different levels of diagnostic uncertainty. A total of 100 samples are selected, consisting of 50 low-uncertainty samples and 50 high-uncertainty samples. All samples are drawn from the test set of the Tennessee Eastman process dataset, ensuring that they are not seen by the diagnostic models during training.

Each data instance in the evaluation dataset is formulated as a natural-language query and is accompanied by additional labels used for evaluation. The structure of the dataset is illustrated in Table 1.

**Table 1:** Example of the LLM Evaluation Dataset

| Question | True label | Uncertainty level |
|---|---|---|
| What is the fault type of sample 761? | 13 | Low |
| ... | ... | ... |
| What is the fault type of sample 4201? | 21 | High |

It is important to emphasize that this work focuses on evaluating the role of LLMs in enhancing fault diagnosis workflows, rather than improving the predictive accuracy of the underlying diagnostic algorithms. Accordingly, this study does not aim to develop or benchmark new fault diagnosis models. Instead, the emphasis is on exploring an LLM-centered framework that improves usability, transparency, and decision support in fault diagnosis by integrating existing diagnostic models with LLM agents.

### 3.3 Evaluation Metrics

To comprehensively evaluate the performance of the proposed LLM-enhanced fault diagnosis framework, three complementary metrics are employed. These metrics assess technical correctness, rule compliance, and operator-oriented response quality.

**1) Diagnostic Accuracy**

Diagnostic accuracy is evaluated using an LLM-as-a-judge approach with a tiered scoring scheme of $\{0, 0.5, 1\}$, based on the true fault label and the response generated by the LLM agent. A score of 1 is assigned if the primary fault reported by the agent matches the true label. If the true fault is mentioned as an alternative fault, a score of 0.5 is assigned. Otherwise, a score of 0 is given. This tiered scheme accounts for uncertainty-aware responses and avoids penalizing the agent for explicitly acknowledging diagnostic ambiguity.

**2) Tool Adherence**

Tool adherence is evaluated using a rule-based evaluator that checks whether the LLM agent follows the tool selection strategy defined in Section 2.4. The evaluation is performed based on the labeled uncertainty level of each sample and the corresponding expected tool invocation. An LLM agent is considered compliant if it consistently invokes the appropriate diagnostic tool (single model or ensemble) according to the predefined uncertainty threshold.

**3) Operator Report Quality**

Operator report quality is assessed using an LLM-as-a-judge approach, with scores ranging from 1 to 5. A system prompt is provided to guide the judge LLM in scoring the responses based on multiple criteria, including alignment with the labeled uncertainty level, clarity and directness of the explanation, effectiveness in supporting operator decision-making, and the actionability of the generated report.

To help readers better understand the LLM's reasoning in the fault diagnosis task, Table 2 presents an example output generated by *gemini-2.5-flash* for sample 4201.

**Table 2:** Answer from gemini-2.5-flash of sample 4201

| Question: What is the fault type of sample 4201? |
|---|
| Answer: For sample 4201, the primary fault type is 15. However, the confidence in this prediction is low (around 30%), indicating significant uncertainty.<br>Alternative fault types to consider are:<br>▪ **Fault type 21** (with a mean probability of 29.17%)<br>▪ **Fault type 9** (with a mean probability of 11.84%)<br>To distinguish between these, the operator should check for symptoms related to both fault type 15 and fault type 21, as they have very similar probabilities. Additionally, a review of indicators for fault type 9 would be prudent. |

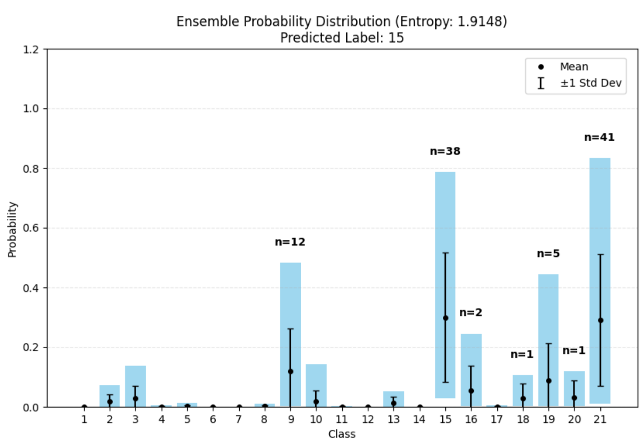


**Figure 2.** Diagnostic result visualization of sample 4201 by calling tool Predict_sample_ensemble_with_plot()

For this sample, the generated response achieves a fault accuracy score of 0.5, a report quality score of 5, and a tool adherence score of 1. These scores reflect that, although the prediction is uncertain, the LLM correctly follows the intended diagnostic strategy. Specifically, the LLM invokes the ensemble-based diagnostic tool, which produces a visualization of the fault probability distribution and exposes the uncertainty across fault classes. The LLM then incorporates this information into a structured and cautious explanation, discussing multiple plausible fault hypotheses instead of over-committing to a single class.

Overall, this example illustrates how the LLM leverages tool outputs and uncertainty information to generate an interpretable, decision support-oriented diagnosis, highlighting its potential role in enhancing fault diagnosis beyond raw classification results.

## 3.4 Results and Discussion

To evaluate the effectiveness of different LLMs within the proposed MCP-based fault diagnosis framework, experiments were conducted using multiple LLMs under identical settings. The evaluated models span multiple providers, architectures, and deployment types, including both open-source and closed-source models with varying parameter sizes. All models were evaluated under the same environment with identical prompts and tools, ensuring a fair comparison. The experimental results are summarized in Table 3.

**Table 3:** Experimental results from the LLM-enhance fault diagnosis report

| Model name | Fault Accuracy | Tool Adherence | Report Quality |
|---|---|---|---|
| openai/gpt-4o-mini | 0.83 | 1.00 | 5.00 |
| x-ai/grok-4-fast | 0.82 | 1.00 | 5.00 |
| anthropic/claude-3.5-haiku | 0.82 | 0.99 | 5.00 |
| google/gemini-2.5-flash | 0.82 | 1.00 | 4.94 |
| openai/gpt-4o | 0.82 | 1.00 | 5.00 |
| deepseek/deepseek-chat-v3.1 | 0.79 | 0.97 | 4.85 |
| meta-llama/llama-3.1-8b-instruct | 0.75 | 0.85 | 4.60 |
| google/gemini-2.5-flash-lite | 0.70 | 0.69 | 4.89 |
| qwen/qwen3-235b-a22b-2507 | 0.62 | 0.86 | 4.47 |
| meta-llama/llama-3.3-70b-instruct | 0.60 | 0.64 | 3.20 |

From these results, several conclusions can be drawn.

First, models from major commercial providers consistently demonstrate strong performance across all three metrics, with relatively small performance differences among them. This indicates that advanced LLMs are generally capable of strictly following the proposed tool selection strategy and generating high-quality, operator-oriented fault diagnosis reports. These results highlight the effectiveness of leveraging LLMs to enhance fault diagnosis workflows in chemical processes.

Second, differences between lightweight and full-scale models are model-dependent rather than universal. For example, Gemini-2.5-Flash-Lite performs noticeably worse than Gemini-2.5-Flash, suggesting that model compression may negatively impact tool adherence and diagnostic reasoning. In contrast, gpt-4o-mini achieves performance comparable to its full-scale counterpart (gpt-4o), demonstrating that it is possible to deploy lightweight models while still achieving high diagnostic accuracy and report quality when tool-calling behavior is well aligned with the task requirements.

Third, open-source models are generally less competitive in this setting, exhibiting lower fault accuracy, weaker tool adherence, and reduced report quality. Among them, deepseek-chat-v3.1 performs the best, while llama-3.1-8b, qwen3-235b, and llama-3.3-70b show significantly poorer performance. Notably, the wide variation in parameter sizes among these models further indicates that larger model size does not necessarily lead to better performance in MCP-based fault diagnosis tasks.

Finally, it can be observed that the upper bound of fault diagnosis accuracy across all evaluated models is approximately 0.83, even when tool adherence reaches 100%. This suggests that the accuracy of the LLM agent is fundamentally constrained by the performance of the underlying diagnostic tools. This behavior is intuitive: by mitigating hallucinations through strict tool usage, the LLM's outputs are limited to the information provided by the invoked algorithms.

Overall, these results demonstrate that LLMs can effectively assist fault diagnosis in chemical processes by leveraging tool-calling capabilities and natural-language report generation. The findings further indicate that LLM performance varies across models, and that selecting models with strong tool-calling reliability and reasoning ability is more important than choosing models solely based on parameter size. Rather than establishing a permanent ranking of LLMs, these results highlight the importance of a systematic benchmarking process for evaluating LLM-based agents in fault diagnosis tasks.

## 4. CONCLUSION AND FUTURE WORK

This work proposes an LLM-enhanced fault diagnosis framework that integrates large language models with fault diagnosis algorithms through structured tool calling. By enabling the LLM to invoke diagnostic tools and generate operator-oriented reports, the proposed framework improves the usability and efficiency of fault diagnosis in chemical processes. Experimental results across multiple LLMs demonstrate that many models are capable of correctly following the proposed strategy, while the achievable diagnostic accuracy is fundamentally constrained by the performance of the underlying fault diagnosis algorithms. Advanced LLMs, however, are effective in interpreting diagnostic results and communicating them to operators in a clear and actionable manner.

Overall, the results highlight the potential of applying LLMs to enhance fault diagnosis workflows in chemical engineering, particularly by improving user experience, transparency, and decision-making efficiency rather than by replacing existing diagnostic models.

Future work will focus on extending the proposed framework toward more advanced multi-agent systems and scaling up the validation to a broader range of industrial process systems. In addition, the observed performance gap between open-source and commercial LLMs motivates further research into improved system design and agent strategies to narrow this gap. Advancing the capability of open-source LLM-based systems is particularly important for addressing data privacy and deployment constraints in industrial environments.

Specifically, many critical chemical facilities operate on strictly air-gapped control networks without internet access to mitigate cybersecurity risks and protect sensitive intellectual property. Cloud-based commercial LLMs cannot be deployed in these settings, making locally deployed open-source models a strict industrial necessity. Furthermore, because chemical process monitoring is a continuous task, relying on commercial APIs would incur significant ongoing token costs, whereas locally deployed open-source models incur zero variable query costs once established. Therefore, enhancing the performance of local, open-source LLMs remains a critical priority for practical industrial deployment.

## ACKNOWLEDGEMENTS

This research was funded by the European Union Horizon Europe 2022 Research and Innovation Programme under the Marie Sklodowska-Curie Grant Agreement No. 101119358 (ProSafe).

## AUTHOR IDENTIFIERS

Author ORCIDs:
Liang Jingkang: 0000-0003-0815-3135
Sin Gürkan: 0000-0003-0513-4502

## REFERENCES

1. Bi X, Qin R, Wu D, Zheng S, Zhao J. One step forward for smart chemical process fault detection and diagnosis. Computers & Chemical Engineering 164:107884 (2022). https://doi.org/10.1016/j.compchemeng.2022.107884

2. Taqvi SAA, Zabiri H, Tufa LD, Uddin F, Fatima SA, Maulud AS. A review on data?driven learning approaches for fault detection and diagnosis in chemical processes. ChemBioEng Reviews 8:239-259 (2021). https://doi.org/10.1002/cben.202000027
3. Deng X, Zhang Z. Nonlinear chemical process fault diagnosis using ensemble deep support vector data description. Sensors 20:4599 (2020). https://doi.org/10.3390/s20164599
4. Lomov I, Lyubimov M, Makarov I, Zhukov LE. Fault detection in tennessee eastman process with temporal deep learning models. Journal of Industrial Information Integration 23:100216 (2021). https://doi.org/10.1016/j.jii.2021.100216
5. Shi T, Yang A, Jin Y, Ren J, Shen W, Dong L, Man Y. Artificial intelligence in process systems engineering. Applications of Artificial Intelligence in Process Systems Engineering :1-10 (2021). https://doi.org/10.1016/b978-0-12-821092-5.00010-3
6. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need 2023. https://doi.org/10.48550/arXiv.1706.03762
7. Cao Y, Griffiths T, Narasimhan K, Shafran I, Yao S, Yu D, Zhao J. Tree of thoughts: deliberate problem solving with large language models. Advances in Neural Information Processing Systems 36 :11809-11822 (2023). https://doi.org/10.52202/075280-0517
8. Ferrag MA, Tihanyi N, Debbah M. From LLM Reasoning to Autonomous AI Agents: A Comprehensive Review 2025. https://doi.org/10.48550/arXiv.2504.19678
9. Caccavale F, Gargalo CL, Kager J, Larsen S, Gernaey KV, Krühne U. Chatgmp: a case of AI chatbots in chemical engineering education towards the automation of repetitive tasks. Computers and Education: Artificial Intelligence 8:100354 (2025). https://doi.org/10.1016/j.caeai.2024.100354
10. Liang J, Groll N, Sin G. Large Language Model Agent for User-friendly Chemical Process Simulations 2026. https://doi.org/10.48550/arXiv.2601.11650
11. Zeng T, Badrinarayanan S, Ock J, Lai C-K, Farimani AB. LLM-guided Chemical Process Optimization with a Multi-Agent Approach 2025. https://doi.org/10.48550/ARXIV.2506.20921
12. Khan A, Nahar R, Chen H, Constante Flores GE, Li C. Faultexplainer: leveraging large language models for interpretable fault detection and diagnosis. Computers & Chemical Engineering 199:109152 (2025). https://doi.org/10.1016/j.compchemeng.2025.109152
13. Gu J, Jiang X, Shi Z, Tan H, Zhai X, Xu C, Li W, Shen Y, Ma S, Liu H, Wang S, Zhang K, Lin Z, Zhang B, Ni L, Gao W, Wang Y, Guo J. A survey on llm-as-a-judge. The Innovation :101253 (2026). https://doi.org/10.1016/j.xinn.2025.101253
14. Liang J, Sin G. Enhancing fault diagnosis for chemical processes via MSCNN with hyperparameters optimization. Systems and Control Transactions 4:1712-1717 (2025). https://doi.org/10.69997/sct.181373
15. What is the Model Context Protocol (MCP)? Model Context Protoc n.d. https://modelcontextprotocol.io/docs/getting-started/intro (accessed December 3, 2025).
16. Melo A, Câmara MM, Clavijo N, Pinto JC. Open benchmarks for assessment of process monitoring and fault diagnosis techniques: a review and critical analysis. Computers & Chemical Engineering 165:107964 (2022). https://doi.org/10.1016/j.compchemeng.2022.107964

© 2026 by the authors. Licensed to PSEcommunity.org and PSE Press. This is an open access article under the creative commons CC-BY-SA licensing terms. Credit must be given to creator and adaptations must be shared under the same terms. See https://creativecommons.org/licenses/by-sa/4.0/