The Human Side of Service Engineering, Vol. 221, 2026, 344–354
https://doi.org/10.54941/ahfe1007738

Development of "WeaveBack": An Integrated System for Human Error Prevention

Hinata Nogi, Joohyun Lee, and Yusaku Okada

Graduate School of Science and Technology, Keio University, Japan

ABSTRACT

Near-miss reports capture frontline signals that can prevent accidents, yet many organizations stop at recording and sharing them, and struggle to translate narratives into implementable actions. This study proposes WeaveChain, an integrated framework for converting near-miss narratives into actionable knowledge through three stages—Factors, Mode, and Action—and focuses on the design and evaluation of its action-generation module, WeaveBack. WeaveBack takes as input Performance Shaping Factors (PSFs) extracted upstream and one of 20 human-error modes and generates candidate countermeasures anchored in PSF–mode combinations. To prevent generic, context-insensitive outputs and vigilance-only advice, WeaveBack enforces a structured protocol that crosses ten intervention domains (L/T/H/P/E/C/R/M/O/X) with two intents (human-error prevention; frontline improvement), thereby ensuring 20 comparable candidates per case. We further curated two reference datasets (software-oriented and hardware-oriented, 100 factors each) and refined them through an iterative improvement loop that alternates KPI scoring (11 KPIs, 0–10) and human revision. Rubric-based evaluation conducted by the authors showed that, across all factors in both datasets, the mean scores for KPI1–KPI5 exceeded the threshold (≥5), while KPIs related to external value remained relatively lower. These results suggest that the proposed design can operationalize near-miss learning for small teams by stabilizing a structured candidate set that supports comparison, selection, implementation, and continuous refinement.

Keywords: Near-miss reports, Human error prevention, Human–AI collaboration, Countermeasure generation, KPI-based evaluation

INTRODUCTION

Near-miss activities provide a foundation for preventing accidents and failures by capturing early weak signals and concerns on the frontline. However, many organizations lack the time and expertise to continuously translate reports into implementable countermeasures and to run a learning loop from analysis to decision-making, implementation, and reflection. As a result, the activity tends to stagnate at sharing or end with reminders to be careful. Recent work has explored methods using generative AI, automated root cause analysis (RCA), and human-in-the-loop support designs. Nevertheless, general-purpose generative AI often converges to generic recommendations that do not fit the input context, and the resulting countermeasures can

Received February 6, 2026; Revised April 11, 2026; Accepted April 28, 2026; Available online July 20, 2026

drift toward vigilance-only statements such as "pay attention" or "raise awareness." Moreover, when the output resembles a single recommended action, frontline teams have limited room to compare alternatives and build consensus.

We therefore position generative AI not as a one-off advisor but as a mechanism that reliably supplies a comparable set of candidate countermeasures to support frontline decision-making. Specifically, we decompose countermeasures into ten intervention domains (see Table 1 in Section 3) and require one item per domain for each of two intents—human-error prevention and frontline improvement—so that 20 candidates are produced in every case. This fixed protocol suppresses individual biases and drift toward generic answers and ensures a stable starting point for discussion. Countermeasure quality must be compared not only in terms of safety but also in operational impact, psychological aspects, and external value. We therefore introduce an 11-KPI evaluation framework and use the KPIs not as pass/fail criteria but as indicators that reveal weaknesses and guide rewriting. By combining protocol-level constraints with KPI-driven refinement, we aim to enable small teams to sustain a learning loop of analysis, decision-making, implementation, and reflection. Rather than linking factors directly to countermeasures, WeaveChain inserts an explicit diagnostic interpretation that supports decision-making. WeaveChain separates responsibilities across Factors, Mode, and Action, and presents options in a form that is easy for people to judge—thereby operationalizing Collaborative Orchestration Intelligence (COI). This paper focuses on WeaveBack, describing the protocol that guarantees 20 proposals via the ten domains × two intents design, the iterative refinement using reference datasets and KPIs, and the evaluation results.

RELATED WORK

LLM-based RCA support has progressed toward inferring causes from logs and records and integrating reasoning with tool use (Roy et al., 2024; Luo et al., 2025; Zhang et al., 2025; Wang et al., 2024; Chen et al., 2024). However, near-miss management requires outputs that explicitly account for performance shaping factors and organizational context and that support comparison, consensus building, and action planning on the frontline. Structured prompting and guardrails can improve generation reliability (Boit and Patil, 2025; Meynhardt et al., 2025; Sultan et al., 2024). Yet for near-miss use, protocol-level constraints are essential to prevent convergence to generic responses and drift toward vigilance-only advice.

In addition to governance and human-in-the-loop operation frameworks and AI risk management (Umakanth, 2025; Basir, 2025; Bengio et al., 2024; Vassilev et al., 2025), designing learning loops that account for psychological safety and error management culture is critical (Conchie et al., 2012; Cusin and Goujon-Belghit, 2019). Our approach prioritizes COI: instead of fully automating conclusions, it makes the factors–mode–actions pipeline explicit to connect to practice. To bridge this gap, WeaveChain separates responsibilities across three layers: FactorWeave structures factors (PSFs),

ThreadMesh provides a diagnostic interpretation, and WeaveBack proposes candidate countermeasures. This design avoids treating generative output as the correct answer; instead, it offers a set of options linked to evidence so that teams can compare, select, and learn. Across prior work, key themes include

(1) automated diagnosis from logs and metrics
(2) systematized prompt design
(3) role design for human-in-the-loop operation
(4) AI risk governance
(5) text analytics for incident reports.

In frontline near-miss practice, however, the workflow must function end-to-end—covering factor coverage, interpretation, candidate generation, consensus building, implementation, and learning—yet designs that connect these components are often missing.

SYSTEM OVERVIEW: WEAVECHAIN AND WEAVEBACK

WeaveChain is an integrated pipeline that translates near-miss narratives into implementable action candidates by distributing responsibilities across factor structuring, diagnostic interpretation, and countermeasure generation. The input is a free-text near-miss report; the output is a set of candidate countermeasures that teams can compare and select, together with the underlying factors and interpretations that justify them. FactorWeave extracts Performance Shaping Factors (PSFs) from free-text narratives and organizes them into ten domains (shown as Table 1) derived from m-SHEL and extended with practical perspectives for frontline management. During extraction, it distinguishes contextual background conditions from intervenable factors and rewrites factor statements at a granularity suitable for downstream decision-making. This process absorbs individual wording differences and yields a comparable set of factors.

Table 1: 10 intervention domains for human error countermeasures.

Category	Focus	Key Requirement
L: Individual Level (Cognitive, Judgmental, and Psychological Operations)	*Focus:* allocation of attentional resources, decision deferral, prioritization, and sense-making processes.	*Key requirement:* elucidating mechanisms of delayed decision-making under multitasking conditions and the inherent difficulty of intuitively grasping urgency.
T: Team Coordination (Interaction among Command, Field Operators, and Other Units)	*Focus:* shared prioritization, granularity of communication, strength and tone of language, and ambiguity of role boundaries.	*Key requirement:* deepening the structural analysis of situations in which information is nominally shared, yet the "action temperature" across actors remains misaligned.

(Continued)

Table 1: Continued.

Category	Focus	Key Requirement
H: **Hardware and User Interface (Equipment, Displays, Protective Functions)**	*Focus:* alarm design, display prioritization, operational guidance, and unintended side effects of fail-safe mechanisms.	*Key requirement:* identifying structural conditions under which critical information becomes obscured during emergencies due to excessive information density.
P: **Manuals and Procedures (Procedure Design, Sequencing, and Exception Handling)**	*Focus:* insufficient clarification of procedural order, missing branches for exceptional cases, and internally inconsistent rules.	*Key requirement:* evaluating procedures not by their readability, but by whether they enable execution without hesitation in real situations.
E: **Work Environment (Workload, Concurrency, and Field Conditions)**	*Focus:* simultaneous task demands, communication congestion, passenger/customer interactions, visibility constraints, and noise.	*Key requirement:* clarifying how environmental conditions actively *induce* delays in judgement rather than merely accompany them.
C: Safety Education and Training (Training Design, Learning, and Scenario Imagination)	*Focus:* forgetting of low-frequency events, insufficient training for compound scenarios, and lack of adaptive capability.	*Key requirement:* assessing whether training fostered autonomous judgement, rather than mere execution of predefined "correct" actions.
R: Rules and Regulations (Authority, Standards, Exceptions, and Responsibility Boundaries)	*Focus:* impracticable rules, formalistic compliance, and unclear discretionary authority.	*Key requirement:* examining paradoxical situations in which rule compliance itself amplifies risk.
M: On-Site Management (Supervision, Command, and Priority Declaration)	*Focus:* resource allocation, global situational awareness, and triage directives.	*Key requirement:* identifying the adverse effects of management practices that prioritize surveillance and control over genuine operational support.
O: Organizational Design (Departmental Structure, Culture, and Learning Pathways)	*Focus:* inter-departmental fragmentation, safety culture, and organizational learning from failure.	*Key requirement:* determining how organizational structures themselves render certain errors effectively unavoidable.
X: Other Factors (Frequency, External Disturbances, Institutional and Social Conditions)	*Focus:* extremely low-frequency events and exogenous constraints beyond routine operational assumptions.	

Concretely, FactorWeave separates prerequisite conditions (e.g., equipment status, environmental conditions, time constraints) from intervenable design elements (e.g., information presentation, procedural design, role allocation, supervision). It then rewrites them into concise factor statements so that downstream modules can link them to interventions. When multiple factors interact within a case, FactorWeave enumerates them without redundancy and with clear causal direction, at a level that makes potential intervention points explicit. Assigning factors to the ten domains is not merely labeling; it converts narratives into discussion units aligned with how teams deliberate (individual, team, device/interface, procedure, environment, training, regulations, frontline management, organizational design). For example, even when an incident is described as a missed check, FactorWeave avoids reducing it to individual vigilance (L) and instead decomposes it into candidate domains such as interface design (H), procedural design (P), and supervision/instruction design (M), thereby supporting countermeasure search beyond reminders. ThreadMesh uses the extracted PSF set and situational description to classify the likely human-error mode into one of 20 categories, providing a diagnostic interpretation that the frontline can scrutinize. It outputs not only the mode label but also a textual rationale that explains which PSFs contributed to the judgment, thereby ensuring traceability from factors to interpretation to countermeasures. This allows teams to explore multiple improvement directions even for the same event, depending on the interpretation. The 20 error modes are a taxonomy of recurring failure patterns, organized around perspectives such as misjudgment, missing information, and breakdowns in preparation. They function as an intermediate representation that links causes (factors) to intervention points (countermeasures). Because ThreadMesh outputs the supporting PSFs and situational cues alongside the predicted mode, teams can examine why a given interpretation is proposed and proceed with discussion under an explicit explanation. This prevents leaps from factors to countermeasures and improves explainability needed for consensus building.

WeaveBack generates candidate countermeasures by using PSF–mode combinations as anchors and retrieving similar examples from reference datasets. Each candidate specifies the target of change, the mechanism of effect, implementation conditions, required resources, and anticipated trade-offs, steering away from vigilance-dependent advice. Outputs are structured by the ten domains × two intents protocol, guaranteeing 20 candidates per case. During generation, WeaveBack retrieves examples with similar PSF–mode patterns from the reference datasets and reuses their intent, target, and conditions while adapting them to the case context. This aims to balance contextual fit with reproducibility, avoiding mere template repetition. By treating software (operations, training, coordination) and hardware (fixtures, signage, automation) interventions symmetrically, the candidate set is less likely to collapse into person-dependent measures.

Overall, this design operationalizes COI by providing the evidence (factors and interpretation) and a stable candidate set that enables the frontline to compare, select, implement, and reflect, rather than having the model output a single conclusion. Figure 1 presents the WeaveChain pipeline.

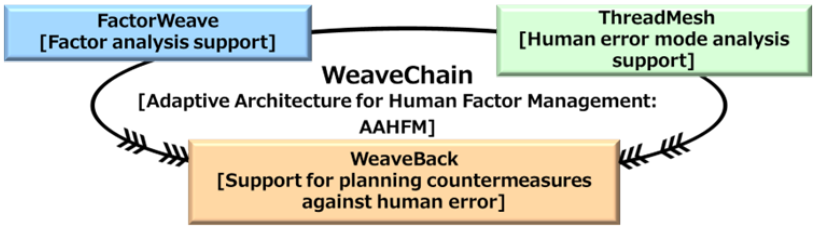

Figure 1: Overall concept of WeaveChain.

METHOD: WEAVEBACK GENERATION AND EVALUATION PROTOCOL

WeaveBack decomposes countermeasures into ten domains and, for each domain, outputs one item for each of two intents—human-error prevention and frontline improvement. This yields 20 candidates per case while preventing domain and intent bias. To avoid drift toward vigilance-only statements such as "be careful" or "pay attention," each candidate must include five elements: (1) target, (2) mechanism of change, (3) implementation conditions, (4) required resources, and (5) trade-offs. We also impose constraints that steer the model toward structural improvements (work design, information design, environmental preparation, organizational procedures) rather than over-relying on recovery checks. To ensure that teams can decide feasibility—when, who, and what to change—we prohibit vague modifiers (e.g., "as appropriate," "sufficiently") and standardize wording so feasibility can be judged concretely. With the domain and intent fixed, candidates are written in a consistent order (target → intervention point → conditions → resources → side effects), aligning granularity and comparison axes so that review and iterative improvement are tractable. Table 2 shows representative examples of how countermeasure descriptions were rewritten through the refinement process.

Table 2: Examples of rewriting countermeasure descriptions: Ver.1 / Ver.3 / Ver.5.

Stage	Example Countermeasure Wording (Except)	KPI Feedback From AI
Ver.1	"Thoroughly check your footing and work carefully."	⚠ AI: No concrete action is specified; it remains a vigilance-only message. (KPI:
Ver.3	"Before starting work, check for wet areas on the floor and clean them."	⚠ AI: It is unclear who performs this and when; added workload to existing tasks is a concern.
Ver.5	"Five minutes before work begins, during the point-and-call time, the team leader verifies the floor using."	✓ AI: The intervention point, actor, and timing are explicit. (KPI: Implementability—High)

Table 3: Definitions and evaluation criteria for the 11 KPIs.

KPI	Name	Intent (Key Definition Points)
KPI1	Deterrence effect	Directness to recurrence prevention, causal fit, sustainability
KPI2	Operational impact	Low additional workload, frontline acceptability, continuity of operation

(Continued)

Table 3: Continued.

KPI	Name	Intent (Key Definition Points)
KPI3	Quality improvement	Secondary improvement of quality, productivity, or service level
KPI4	Psychological safety	Ease of reporting and consultation; contribution to a blameless culture
KPI5	Engagement	Ownership, willingness to participate, and buy-in for improvement
KPI6	Reputation	Trust from customers/regulators and internal/external stakeholders; explainability
KPI7	Non-financial visualization	Ability to quantify qualitative outcomes; ease of recording and monitoring
KPI8	Community impact	Spillover to local community/society; room for collaboration with stakeholders
KPI9	Recursive learning	Mechanisms to feed post-implementation learning back into the next design (feedback)
KPI10	Translation fitness	Applicability across sites/units; ease of adaptation to context
KPI11	Sustained dialogue	Design that sustains dialogue (facilitating meetings, reviews, and consensus building)

To evaluate quality from multiple perspectives, we defined 11 KPIs (0–10 points; $A \geq 5$, $S \geq 7$) and used them to compare candidate countermeasures. The KPIs cover deterrent effect, operational impact, quality improvement, psychological safety, engagement, reputation, visibility of non-financial value, community impact, recursive learning, translation fitness, and sustained dialogue. For the initial validation, the authors conducted rubric-based expert scoring; the first author served as the primary rater. For RAG/few-shot prompting, we curated two reference datasets—software-oriented and hardware-oriented—with 100 factors each, and revised them iteratively. Items whose factor-level mean on KPI1–KPI5 fell below 5 were prioritized for rewriting. Quality was improved by clarifying the target, intervention point, conditions, and resources. Here, KPIs are used not as pass/fail judgments but as indicators that guide refinement. Importantly, what the generative model evaluates here is not the real-world correctness of whether a countermeasure will physically prevent accidents. Instead, it audits the structural completeness of the description—whether it is specific, implementable, and multi-perspective, with explicit actor, intervention point, timing, and conditions. Real-world validity (applicability, prioritization, and side effects) must be ensured by humans, while the model stabilizes the quality of the codified description. With this division of labor, we examine the impact of dataset revisions using KPI scores.

EVALUATION DESIGN

We evaluated the initial and final versions of the reference datasets (software $N = 100$; hardware $N = 100$; total $N = 200$) by scoring each item on the 11 KPIs. Scoring was conducted as rubric-based expert assessment by the authors, with the first author as the primary rater. To verify whether the improvement

loop worked as intended, we compared the mean of KPI1–KPI5 (deterrent effect, operational impact, quality improvement, psychological safety, and engagement) between the initial and final versions. Figure 2 shows that the means increased after revision for both software and hardware datasets. In particular, rewriting to clarify targets, intervention points, conditions, and resources improved ratings related to deterrence and implementability. Figure 2 should be read as the extent to which each KPI was lifted by revision. When KPI3–KPI5, which were lower in the initial version, exceed the threshold, it indicates that a minimum quality line suitable for practical deliberation was secured—not only for deterrence but also for operational and psychological considerations. Next, to characterize variability across the full set of KPIs in the final version, we summarized the distribution of KPI1–KPI11. Figure 3 presents the KPI distributions, showing relatively high scores for KPI1–KPI3, while KPIs related to external value—such as reputation, community impact, and sustained dialogue—remain lower. This suggests that requirements such as stakeholder-facing explanations, collaboration procedures, and visualization templates were not yet sufficiently embedded in the outputs. In Figure 3, the median and interquartile range of the boxplots indicate variability and stability across KPIs. KPI1–KPI3 have higher medians and smaller dispersion, suggesting that deterrence, implementability, and quality improvement were stabilized to a consistent level. In contrast, external-value KPIs show lower medians, implying that countermeasure text alone tends not to raise them. Rather than a limitation of the system, this gap highlights the boundary between domains where AI-driven text optimization is effective (frontline-implementable interventions) and domains that require managerial or political judgment by humans (external communication, cross-stakeholder collaboration, and investment decisions).

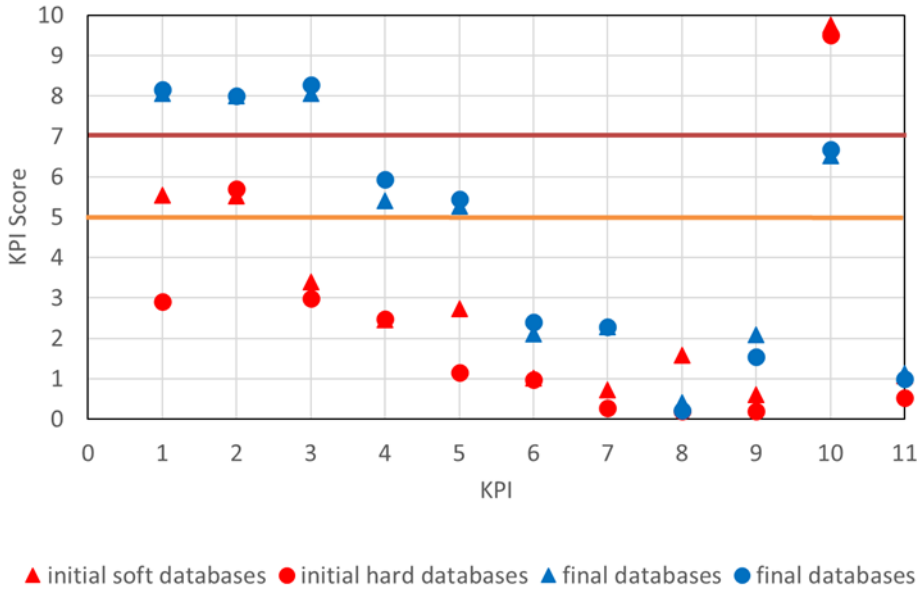

Figure 2: Comparison of mean KPI1–KPI5 between the initial and final versions (software and hardware countermeasures).

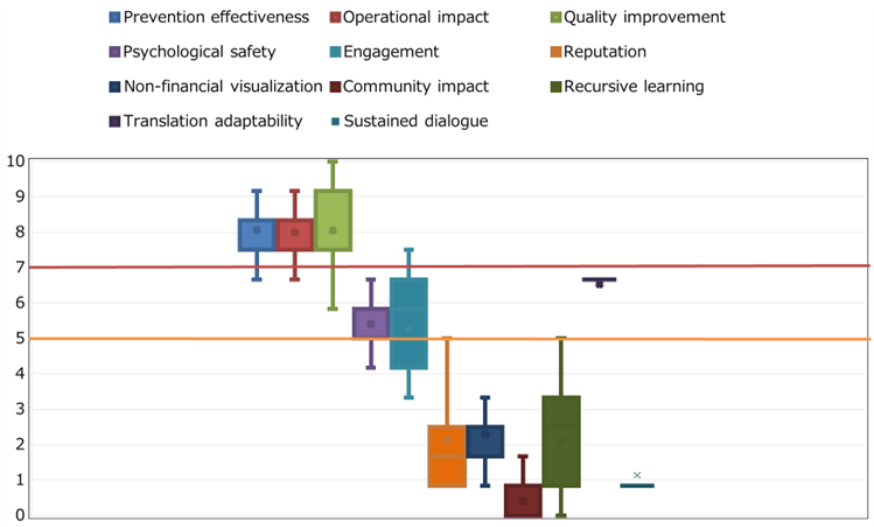

Figure 3: KPI distributions of the final dataset (software countermeasures).

This boundary implies that improving the wording quality of countermeasures is insufficient unless operational artifacts—such as stakeholder-facing explanation materials, visualization, and collaboration procedures—are also prepared. Therefore, WeaveBack may function as a filter that separates countermeasures that can be completed at the frontline from issues that should be escalated to senior management (e.g., reputation-related actions). Future work should explicitly incorporate templates for external communication and collaboration protocols (scope of sharing, consensus procedures, and feedback routes) as output requirements.

CONCLUSION

This paper proposed WeaveBack as the countermeasure-generation module of WeaveChain and presented a design that structurally guarantees 20 candidates per case using the ten domains × two intents protocol, grounded in PSF–mode combinations. We also curated software and hardware reference datasets (100 factors each) and improved the implementability and comparability of countermeasure descriptions through an iterative refinement loop of KPI-based scoring and rewriting. For practical deployment, operation in closed environments (on-premises or private cloud) and information governance when integrating with external models are also critical. Because WeaveBack allows organizations to manage both the reference datasets and the output protocol, it can be deployed in implementations that match security requirements. Our evaluation confirmed that, in the final version, the mean of KPI1–KPI5 exceeded the threshold (≥ 5) for all factors in both datasets, indicating that a minimum quality line for frontline deliberation can be secured. At the same time, KPIs related to external value—such as reputation and community impact—remained comparatively harder to improve. Rather than indicating a system limit, this result clarifies the boundary between areas where AI-driven text optimization is feasible and areas where managerial or political decisions by humans are required. Accordingly, WeaveBack may serve as a filter that separates countermeasures that can be completed locally

from issues that should be escalated to management. Future work should expand output templates for external value (visualization, stakeholder-facing explanations, and collaboration procedures).

A key feature of this design is that it does not force the model to output a single conclusion; instead, it provides a candidate set linked to evidence (factors and interpretation), enabling teams to compare, select, implement, and reflect—an approach we refer to as Collaborative Orchestration Intelligence (COI). We also conducted preliminary walkthrough trials across multiple organizations and found that the factors–mode–actions backbone remains stable across differences in terminology and input granularity. Limitations include rubric-based scoring by the authors and the absence of multi-rater reliability testing and longitudinal field evaluation. Future work will examine inter-rater agreement, link outputs to operational metrics (implementation rate, sustained use, recurrence rate), and strengthen external-value KPIs by expanding output templates and operational artifacts.

REFERENCES

Bengio, Y., Hinton, G., Yao, A., et al. (2024). Managing extreme AI risks amid rapid progress. Science, 384(6698), 842–845. doi:10.1126/science.adn0117

Boit, S., and Patil, I. (2025). A prompt engineering framework for large language model–based mental health chatbots: Conceptual framework. JMIR Mental Health, 12, e75078. doi:10.2196/75078

Chandrashekar, N.D., Nizamani, S.B., Ellis, M., and Ramakrishnan, N. (2025). Demystify, use, reflect: An LLM-centric framework for LLM-users. arXiv. doi:10.48550/arXiv.2507.05480

Chen, Z., Wang, J., Xia, M., Shigyo, K., Liu, D., Zhang, R., and Qu, H. (2024). StuGPTViz: A visual analytics approach to understand student–ChatGPT interactions. arXiv. doi:10.48550/arXiv.2407.12423

Conchie, S.M., Taylor, P.J., and Donald, I.J. (2012). Promoting safety voice with safety-specific transformational leadership: The mediating role of two dimensions of trust. Journal of Occupational Health Psychology, 17(1), 105–115. doi:10.1037/a0025101

Cusin, J., and Goujon-Belghit, A. (2019). Error reframing: Studying the promotion of an error management culture. European Journal of Work and Organizational Psychology, 28(4), 510–524. doi:10.1080/1359432X.2019.1623786

Luo, Y., Jiang, J., Feng, J., Tao, L., Zhang, Q., Wen, X., Sun, Y., Zhang, S., and Pei, D. (2025). From observability data to diagnosis: An evolving multi-agent system for incident management in cloud systems. arXiv. doi:10.48550/arXiv.2510.24145

Meynhardt, C., Hölzing, C.R., Kranke, P., and Meybohm, P. (2025). Advanced prompt engineering for generative artificial intelligence in medical education: The PROMPT+ framework and practical examples from anesthesia and emergency medicine. Electronics, 14(5), 1028. doi:10.3390/electronics14051028

Roy, D., Zhang, X., Bhave, R., Bansal, C., Las-Casas, P., Fonseca, R., and Rajmohan, S. (2024). Exploring LLM-based agents for root cause analysis. arXiv. doi:10.48550/arXiv.2403.04123

Sultan, M.A., Ganhotra, J., and Fernandez Astudillo, R. (2024). Structured chain-of-thought prompting for few-shot generation of content-grounded QA conversations. In Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 16172–16187. doi: 10.18653/v1/2024.findings-emnlp.948

Umakanth, A.A.M. (2025). Human-in-the-loop architectures for validating GenAI outputs in clinical settings. European Journal of Computer Science and Information Technology, 13(47), 103–110. doi:10.37745/ejcsit.2013/vol13n47103110

Vassilev, A., Oprea, A., Fordyce, A., Anderson, H., Davies, X., and Hamin, M. (2025). Adversarial machine learning: A taxonomy and terminology of attacks and mitigations (NIST AI 100-2e2025). National Institute of Standards and Technology. doi:10.6028/NIST.AI.100-2e2025

Wang, Z., Liu, Z., Zhang, Y., Zhong, A., Wang, J., Yin, F., Fan, L., Wu, L., and Wen, Q. (2024). RCAgent: Cloud root cause analysis by autonomous agents with tool-augmented large language models. arXiv. doi:10.48550/arXiv.2310.16340

Zhang, X., Wang, Q., Li, M., Yuan, Y., Xiao, M., Zhuang, F., and Yu, D. (2025). TAMO: Fine-grained root cause analysis via tool-assisted LLM agent with multi-modality observation data in cloud-native systems. arXiv. doi:10.48550/arXiv.2504.20462