

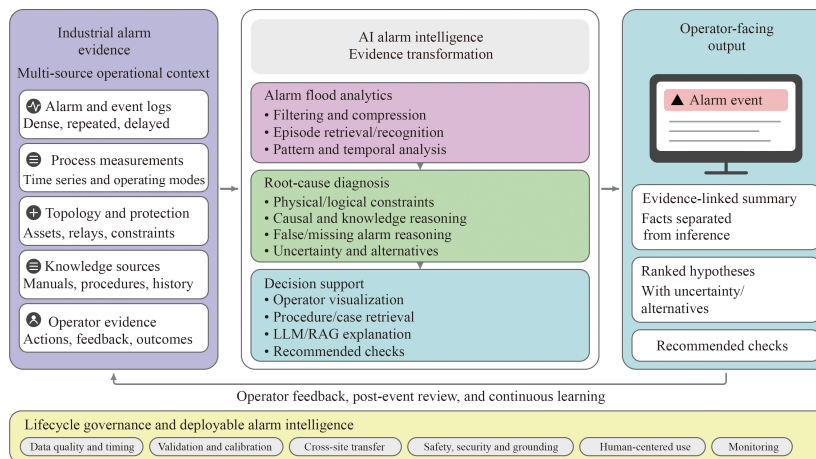
AI-Driven Alarm Management in Industrial Systems: Methods and Applications

Chen Liu (✉), Wei Peng, Xinghuo Yu

School of Engineering, RMIT University, Melbourne VIC 3001, Australia

© The Author(s) 2026. This article is published by Higher Education Press.

GRAPHICAL ABSTRACT



HIGHLIGHTS

- A six-stage framework unifies AI functions across the industrial alarm lifecycle.
- Alarm analytics, diagnosis, and decision support form a connected evidence workflow.
- Evidence-grounded LLM/RAG supports deployable alarm intelligence with safe boundaries.

ABSTRACT Industrial alarm systems are the main interface through which abnormal process conditions, equipment faults, protection events, cyber-physical disturbances, and diagnostic evidence are communicated to human operators. Practical alarm systems still suffer from nuisance alarms, chattering alarms, alarm floods, false and missing alarms, weak root-cause evidence, and limited integration with operator knowledge. This paper reviews artificial intelligence (AI)-driven industrial alarm management through three connected research streams: alarm flood analytics, root-cause diagnosis, and alarm-system decision support. Beyond a model-type taxonomy, this paper provides an operational-stage synthesis of AI-driven alarm management, in which techniques are interpreted according to their functional roles in the alarm lifecycle rather than as strict chronological phases. Unlike a taxonomy that only groups many studies under broad labels, the paper discusses individual studies according to the technical method used, the alarm-management problem addressed, the operational function supported, and the remaining limitation for deployment. This synthesis connects engineering and data foundations, alarm compression and reduction, alarm flood episode recognition, root-cause explanation, evidence-grounded decision support, and lifecycle governance. Within this framework, alarm flood analytics focuses on transforming dense and noisy alarm sequences into structured episode-level evidence, while root-cause diagnosis builds on such evidence to move from statistical associations toward physically, temporally, and causally plausible explanations. Building on these diagnostic outputs, decision-support methods translate alarm evidence into operator-facing information through visualization, knowledge retrieval, and AI-assisted interaction. The review further identifies future perspectives for deployable alarm intelligence, including episode-level benchmarking, uncertainty and false/missing-alarm reasoning, cross-site transfer, evidence-grounded large language model (LLM)/retrieval-augmented generation (RAG) assistance, prompt-injection protection, lifecycle validation, and human-centered deployment. The review shows that deployable alarm intelligence requires not only high model accuracy, but also evidence traceability, uncertainty calibration, cross-site generalization, lifecycle governance, and safe boundaries for LLM-assisted operation.

KEYWORDS Alarm management; Alarm flood; Root-cause diagnosis; Supervisory control and data acquisition (SCADA); Large language models (LLMs); Decision support

1 Introduction

Industrial alarm systems convert process deviations, protection operations, equipment failures, cyber events, and abnormal measurements into messages that operators can interpret and act on. A useful alarm should be timely, selective, prioritized, actionable, and explainable. In practice, however, alarm systems often violate these requirements because of chattering alarms, standing alarms, redundant annunciation, poorly configured thresholds, weak lifecycle management, and abnormality propagation across connected equipment. Wang et al. provided a foundational overview of industrial alarm overloading and clarified that the main problem is not merely the existence of many alarms, but the failure of alarm lifecycle management to maintain operator usefulness [1]. In power-system and supervisory control and data acquisition (SCADA) control rooms, alarm processing must further deal with dense relay, breaker, protection, fault-indicator, and measurement messages under uncertainty, communication delay, false alarms, and missing alarms [2–4].

Established industrial alarm-management guidance, including ISA-18.2 [5], IEC 62682 [6], and EEMUA 191 [7], provides the engineering context for activities such as alarm philosophy, rationalization, prioritization, operation, maintenance, monitoring, assessment, and management of change. The six-stage synthesis introduced in this review is not intended to replace these established lifecycle practices; rather, it provides an AI-oriented functional overlay that identifies where data-driven analytics, diagnostic reasoning, and decision-support methods can contribute to established alarm-management activities.

Recent AI methods address different but interdependent functions of this alarm-management workflow. Alarm flood analytics first transforms dense alarm sequences into structured episode-level evidence by retrieving similar historical episodes, mining frequent patterns, classifying partial floods, and predicting alarm evolution [8–11]. Building on such evidence, root-cause diagnosis incorporates temporal constraints, topology, protection logic, causal relationships, and knowledge representations to infer plausible fault sections and propagation paths [12–14]. The resulting diagnostic evidence can then be translated into operator-facing decision support, where large language models (LLMs), retrieval-augmented generation (RAG), and multi-agent methods assist with alarm summarization, procedure retrieval, report generation, and interactive operator

decision making [15–17]. Importantly, these three streams should not be treated as independent research topics, because the quality and usefulness of each downstream function depend on the evidence produced by the preceding stages. A noisy alarm configuration can corrupt the training data for alarm-flood analytics; an alarm-flood classifier is useful only if it helps operators locate the cause or decide the next check; and an LLM assistant is trustworthy only if its statements are grounded in alarms, measurements, topology, procedures, and diagnostic outputs.

Existing reviews often focus on one part of this landscape, such as industrial alarm overloading, process root-cause analysis (RCA), power-system alarm processing, smart-grid anomaly detection, power-plant fault diagnosis, or LLM applications in energy systems [1,18–21]. These reviews provide valuable domain-specific insights, but they generally examine individual components of alarm intelligence separately rather than their interactions within an end-to-end operational workflow. Reviewing AI-driven alarm management only by model type, data modality, or application domain can hide the functional relationships among techniques that support different stages of alarm operation. In a real control room, alarm flood analytics, diagnosis, and decision support are strongly coupled. The distinctive perspective of this review is therefore not simply to cover these three research streams together, but to examine how evidence is transformed and propagated across them: alarm flood analytics converts raw and noisy alarm information into structured episode-level evidence, root-cause diagnosis transforms this evidence into physically and causally plausible explanations, and decision support translates the resulting diagnostic evidence into operator-facing information and recommended checks. This coupling also extends through lifecycle feedback, as operator actions, alarm rationalization, configuration changes, and updated knowledge can subsequently affect the data and evidence used by upstream analytical and diagnostic methods. Therefore, AI-driven alarm management should be reviewed through three connected functions: alarm flood analytics, root-cause diagnosis, and alarm-system decision support. At a higher level, these functions can be interpreted through an operational-stage synthesis that organizes AI techniques according to their functional roles in the alarm lifecycle rather than as strict chronological phases.

1.1 Review scope and literature selection

The literature considered in this review was identified

through searches across IEEE Xplore, ScienceDirect, SpringerLink, MDPI, and arXiv using combinations of keywords related to industrial alarm management, alarm floods, root-cause diagnosis, SCADA alarm processing, anomaly detection, large language models, retrieval-augmented generation, and decision support. The search focused primarily on studies published from 2015 to 2026, supplemented by backward citation tracking to retain earlier foundational studies. Studies were included when they directly addressed industrial-process, power-system, or SCADA alarm management, or when methods from adjacent domains could be explicitly mapped to alarm-management functions. Studies without a clear connection to alarm analysis, diagnosis, or operator-facing decision support were excluded. Candidate studies were first screened based on their titles and abstracts, followed by full-text assessment of the retained studies against the predefined review scope and inclusion criteria.

Because the objective is a structured narrative synthesis rather than a quantitative meta-analysis, no numerical study-quality score was assigned. Instead, the retained studies were qualitatively assessed according to their relevance to industrial alarm-management functions, methodological contribution, application context, and type of validation evidence, including offline, simulation-based, shadow-deployment, and field-validation evidence.

Direct claims regarding industrial alarm-management capability are based primarily on studies involving industrial processes, power systems, or SCADA alarms, whereas studies from AIOPs, cloud services, cybersecurity, and visual anomaly detection are included only when they provide transferable methods or can be explicitly mapped to industrial assets and operational workflows. For interpretation, the reviewed literature is

therefore distinguished conceptually as core evidence directly involving industrial-process, power-system, or SCADA alarms; adjacent evidence explicitly connected to industrial assets or operating workflows; and transferable evidence from neighboring domains that provides methodological insight but is not treated as direct evidence of industrial alarm-management capability.

The proposed framework is intended as an operational-stage synthesis rather than a strict chronological workflow, taxonomy, or maturity model; the numbered stages represent major functional roles and their principal information dependencies. Fig. 1 presents the operational-stage synthesis map used to organize this review. The three research streams and the six operational stages represent two complementary levels of organization rather than two separate taxonomies. Alarm flood analytics primarily spans Stage 2 (compression and reduction) and Stage 3 (alarm flood episode recognition), root-cause diagnosis corresponds to Stage 4, and alarm-system decision support corresponds to Stage 5. Stage 1 provides the shared engineering and data foundation required by all three streams, whereas Stage 6 supports lifecycle governance, validation, monitoring, and continuous improvement across them. Thus, the three research streams define the main technical contents reviewed in this paper, while the six-stage framework explains how these contents are functionally connected within the broader alarm-management lifecycle.

The map contains six connected stages: engineering and data foundation, compression and reduction, alarm flood episode recognition, root-cause diagnosis, evidence-grounded decision support, and lifecycle governance. The first stage establishes trustworthy alarm and context data, including alarm philosophy, alarm tags, event logs, process time series, topology, manuals,

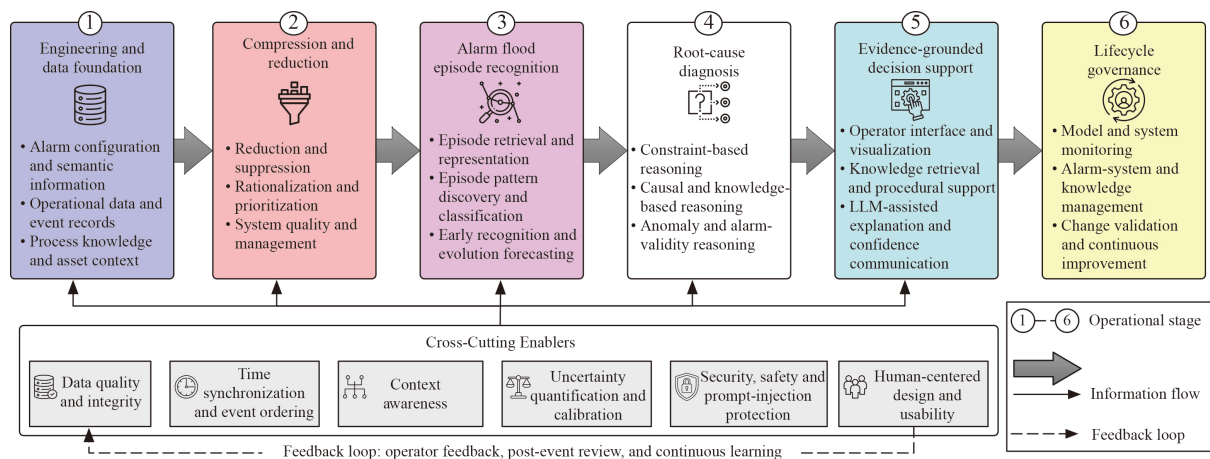


Fig. 1 Operational-stage map of AI-driven alarm management from engineering and data foundation to deployable decision support.

procedures, knowledge bases, and operator actions. The second stage aims to reduce overload and improve alarm quality through nuisance-alarm reduction, chattering and standing-alarm suppression, redundancy elimination, rationalization, prioritization, shelving, and alarm-system quality assessment. The third stage moves from alarm reduction to episode-level recognition by using similarity retrieval, sequence compression and alignment, pattern mining, template extraction, episode classification, early recognition, and episode prediction. The fourth stage shifts the objective from pattern recognition to root-cause diagnosis, where temporal and logical constraints, topology and physical consistency, information-theoretic causality, graph and knowledge reasoning, anomaly detection, and false/missing-alarm reasoning are used to explain why alarms occurred and which causes are most plausible. The fifth stage translates diagnostic evidence into operator-facing decision support through human-machine interface (HMI) visualization, RAG, LLM-assisted summaries and explanations, procedure retrieval, recommended checks, uncertainty communication, and what-if interaction. The final stage emphasizes lifecycle governance, including model and data monitoring, alarm-system management, knowledge and document versioning, benchmarking, change validation, operator feedback, and continuous learning.

The cross-cutting enablers in Fig. 1 apply across the operational stages rather than forming additional sequential stages. Data quality, time synchronization, and context awareness determine whether alarm and diagnostic evidence is reliable and correctly interpreted, whereas uncertainty quantification, security and safety controls, and human-centered design constrain how analytical outputs are converted into operator-facing recommendations. For example, incorrect event ordering can invalidate an inferred propagation path, while poorly calibrated confidence can cause operators to over-trust an unfamiliar episode classification or RCA result. Although lifecycle governance is shown as Stage 6 for organizational clarity, monitoring, version control, change validation, and operator feedback operate continuously throughout deployment rather than only after the preceding stages are completed. Alarm-configuration changes may therefore invalidate historical templates or trained models, while post-event operator feedback can update episode labels, diagnostic knowledge, procedures, and recommended checks.

The main contributions of this review are as follows.

- An operational-stage synthesis of AI-driven alarm management is introduced to connect engineering and data foundations, alarm compression and episode recognition, root-cause explanation, evidence-grounded decision support, and lifecycle

governance. This framework provides a functional view of how AI techniques contribute to different stages of the alarm-management lifecycle rather than organizing them only by model type or application domain.

- A coupling-oriented perspective is provided across alarm flood analytics, root-cause diagnosis, and alarm-system decision support. Rather than treating these streams as independent topics, the review examines how alarm evidence is transformed from episode-level patterns into causal explanations and subsequently into operator-facing information, while lifecycle feedback links downstream operation to upstream data, models, and knowledge.
- Key deployment bottlenecks and future research directions are identified for moving from offline alarm analytics toward deployable alarm intelligence, including episode-level evaluation, uncertainty and false/missing-alarm reasoning, cross-site generalization, evidence-grounded LLM/RAG assistance, lifecycle validation, and human-centered deployment.

The remainder of this paper is organized as follows. Section 2 reviews AI methods for alarm flood analytics. Section 3 reviews root-cause diagnosis methods for alarm systems. Section 4 discusses alarm-system decision support, including HMI, RAG, LLM, multi-agent, and cyber-physical alert applications. Section 5 summarizes future directions toward deployable alarm intelligence. Section 6 concludes the paper.

2 Alarm Flood Analytics

Alarm flood analytics studies dense bursts of alarm messages generated during abnormal operation. In the operational-stage map in Fig. 1, this stream covers two connected functions: compression and reduction, and alarm flood episode recognition. The goal is not only to reduce the number of alarms, but also to transform symbolic, sparse, delayed, repeated, missing, and context-dependent alarm sequences into structured episode evidence. Such evidence can support case retrieval, early episode recognition, prediction, RCA, and operator-facing decision support. Table 1 summarizes the major method families, while the following subsections further compare their assumptions, data requirements, evaluation characteristics, and deployment limitations.

2.1 Compression and reduction: alarm rationalization and nuisance-alarm control

Alarm rationalization is the upstream stage of alarm flood

Table 1 Alarm flood analytics methods and their operational roles.

Method group	Typical input	Output	Deployment concern
Compression and reduction	Alarm limits, alarm/event logs, operating modes, operator actions	Nuisance candidates, alarm-quality indicators, priority evidence [22–30]	Historical logs may encode poor practice; rare safety alarms require expert review
Episode retrieval, sequence compression, and alignment	Historical floods, alarm tokens, time gaps, coactivation series	Similar episodes, compressed and aligned sequences, case-based evidence [8,11,31–33]	Similar tags can hide different topology, operating mode, or root cause
Pattern mining and templates	Alarm groups, ordered subsequences, repeated floods	Frequent patterns, fault templates, propagation candidates [9,34–38]	Frequent patterns are not necessarily causal and may be site-specific
Early episode recognition, classification, and prediction	Partial floods, labels, word embeddings, templates	Event class, early warning, next alarms, remaining time [10,39–47]	Evaluation must report earliness, stability, calibration, and abstention

analytics because flood data quality depends on whether the alarm system contains redundant, chattering, standing, or poorly prioritized alarms. This subsection corresponds to the compression and reduction stage in Fig. 1. Its operational role is to improve the quality of alarm evidence before downstream episode recognition, RCA, or decision support is applied.

Existing studies approach alarm rationalization from several complementary perspectives. One line of research focuses on improving alarm-generation quality under dynamic and multivariate operating conditions. For example, Xiong et al. used Bayesian filters to account for uncertainty and time-varying process behavior in multivariate alarm design [22], while multivariate nuisance-alarm analysis considers correlated process variables jointly rather than treating each alarm in isolation [29]. A second line of research exploits historical alarm and event records to identify repetitive, redundant, or context-dependent alarm behavior. Representative approaches include data-driven alarm and event analysis for revealing alarm-system weaknesses [23], correlation analysis for identifying repeatedly co-occurring alarms [24], cyclic mining of alarm and operator-action events [25], association-rule discovery for mode-dependent alarms [26], and exploratory analysis of alarm distributions and repetition patterns [28]. From a human-centered perspective, smart-alarm evaluation further emphasizes that alarm usefulness should be assessed in relation to operator perception and response rather than solely through technical detection accuracy [27]. Beyond statistical association, transfer-entropy-based analysis has also been introduced to examine directional dependence among alarm variables and support alarm-overload mitigation [30].

Collectively, these studies show that alarm rationalization is not a single nuisance-detection problem, but a process that combines statistical evidence, operating context, causal information, and operator response to support engineering review. Accordingly, compression and reduction should be viewed as controlled evidence refinement rather than as an automatic alarm-suppression or deletion mechanism.

A further observation from this group is that rationalization studies often use different definitions of

“bad” alarms. Some works focus on alarm frequency and duration, some focus on co-occurrence and redundancy, some focus on operator response, and others focus on causal redundancy. This means that a rationalization tool should not return a single nuisance score without explanation. A high-frequency alarm with no operator response, a correlated alarm that always follows another alarm, and an alarm whose relevance depends on operating mode represent different engineering cases. They require different review actions, such as threshold tuning, deadband adjustment, shelving-rule revision, priority review, or procedure update. Therefore, a deployable AI tool should output a rationalization dossier rather than a binary keep/remove decision.

This also affects downstream alarm flood analytics. If nuisance reduction is performed too aggressively, sequence classifiers may lose early evidence that is useful for diagnosis. If rationalization is too conservative, alarm floods remain dense and operators still suffer from overload. The reviewed studies suggest that rationalization should be iterative: log mining proposes candidates, engineering review validates them, updated configurations are monitored, and the effect on flood analytics and operator workload is measured over time. This iterative loop also links the compression stage to lifecycle governance, because rationalization decisions should be revisited after configuration changes, process modifications, or changes in operator practice.

2.2 Episode retrieval, sequence compression, and alignment

Similarity retrieval supports operators by showing whether a current alarm flood resembles a previous incident. In the operational-stage map, this subsection marks the transition from alarm reduction to alarm flood episode recognition. The key objective is to compare an unfolding alarm sequence with historical episodes while preserving enough temporal and symbolic structure for later diagnosis and decision support. Ahmed et al. performed similarity analysis of industrial alarm flood data and demonstrated that historical flood episodes can be retrieved as case-based evidence for understanding current floods [8]. Zhou et al. proposed generalized

pattern matching through word processing and sequence alignment, treating alarm tags as symbolic tokens while allowing insertion, deletion, and shifts in alarm order [11]. Guo et al. developed an accelerated alignment method for analyzing alarm-flood time sequences, reducing the computational burden of comparing long or repeated alarm sequences [31]. Manca et al. clustered similar historical alarm subsequences using alarm series and characteristic coactivations, which provides a way to group recurring subsequences without requiring predefined fault labels [32]. The consequential-alarm similarity study in [33] focused on identifying alarms that are similar in their downstream consequence, shifting retrieval from simple token overlap to operational relevance.

These retrieval methods solve a different problem from classification. Instead of assigning a current flood to a fixed class, they expose previous cases that can be inspected. Their limitation is that similarity in alarm tokens does not necessarily imply similarity in cause. A current flood may share many tags with a historical event but occur under a different process mode, protection state, or topology. Future retrieval systems should therefore report not only the nearest episodes, but also the contextual differences that may invalidate the comparison. In other words, retrieval should provide comparable evidence, not a causal conclusion.

Another practical issue is the representation of time. Some similarity methods treat an alarm flood as an ordered list of tags, while others preserve time gaps or coactivation windows. The first representation is robust to irregular timestamps but may hide whether alarms happened simultaneously or through slow propagation. The second representation is more faithful to process dynamics but may be sensitive to logging delays and communication latency. For operator support, the retrieved case should therefore include both symbolic similarity and temporal similarity. An episode that has the same tag order but a much slower propagation speed may indicate a different severity or intervention window. Sequence compression adds another representation choice: it can improve tractability and remove repetitive alarm tokens, but it must avoid deleting early weak signals that are useful for RCA.

Similarity retrieval also creates a natural bridge to RCA and decision support. If retrieved episodes include verified root causes, corrective actions, and outcomes, they become evidence for diagnosis and procedure recommendation. If they include only alarm sequences without context, they may confuse operators. Thus, future retrieval systems should store historical floods as structured cases containing sequence, plant mode, topology or equipment state, root-cause label, operator

action, and post-event assessment.

2.3 Pattern mining and template extraction

Pattern mining extracts repeated structures from alarm histories. This subsection further supports the episode-recognition stage by converting repeated alarm structures into reusable operational signatures. Lai and Chen proposed pattern mining in multiple alarm flood sequences, showing how recurring alarm groups can be discovered across more than one flood episode rather than in isolated event logs [34]. Hu et al. detected frequent alarm patterns using itemset mining, which identifies alarm groups that commonly appear together even when precise temporal order is not the main objective [9]. Zhou et al. modified the CloFAST algorithm to extract ordered patterns from alarm-flood sequences, preserving sequence information that itemset mining may ignore [35]. The modified PrefixSpan method in [37] also targeted sequential pattern mining, providing another way to recover frequent ordered alarm subsequences from flood data.

Template methods translate mined patterns into operational references. Charbonnier et al. extracted fault templates to assist operators during alarm floods, turning repeated alarm structures into templates that can be matched against future floods [36]. The interpretable template-matching study in [38] extended this idea to online alarm flood classification, emphasizing that template evidence should be understandable to operators. Pattern mining and template extraction are useful because they can reveal alarm-propagation signatures, but their main limitation is that frequency is not causality. A frequent sequence may reflect redundant annunciation or common operator actions rather than physical propagation. Therefore, mined patterns and templates should be treated as episode signatures rather than causal explanations. They identify what alarm structures tend to appear together, but they do not by themselves explain why the alarms occurred.

The difference between itemset mining and sequential mining is also important. Itemset mining is useful when the main question is which alarms form a recurring group, regardless of order. Sequential mining is useful when the order of alarms carries diagnostic meaning, such as a pressure deviation followed by a compressor alarm and then a shutdown alarm. Template extraction adds another layer by converting repeated subsequences into named operational patterns. This is attractive for operator support because templates can be documented, reviewed, and linked to procedures. However, templates can become stale after alarm configuration changes, process modifications, or control-logic updates. This

limitation again shows the need to connect episode-recognition methods with lifecycle governance.

A promising future direction is to combine pattern mining with hierarchy. Mining at the tag level captures detailed plant behavior, but it does not transfer well across sites because tag names are local. Mining at the unit or equipment level improves generalization but loses diagnostic precision. A hierarchical template could represent the same flood at multiple levels: tag-level sequence, equipment-level sequence, unit-level propagation, and process-area pattern. This would make mined patterns more reusable and easier for operators to interpret. Such hierarchical representations would also make the transition from episode recognition to RCA more explicit, because causal explanation often requires equipment-level or process-level reasoning rather than only tag-level matching.

2.4 Early episode recognition, classification, and prediction

Classification and prediction methods support early decision making during an unfolding alarm flood. In the operational-stage view, classification and prediction mark the transition from alarm compression to episode-level recognition. The operational question is not only which class a completed flood belongs to, but whether a developing episode can be recognized early enough to support diagnosis and operator action. Shang and Chen proposed exponentially attenuated component analysis, giving stronger weight to early alarms so that a partial flood can be classified before the full sequence is available [10]. Alinezhad et al. developed semisupervised early classification, reducing dependence on fully labeled flood episodes, which are scarce in industrial systems [40]. Alinezhad et al. also proposed a modified bag-of-words representation, converting alarm floods into compact textual-style features while adapting ordinary word-count ideas to industrial alarm semantics [48]. Lucke et al. reviewed advances in alarm data analysis and demonstrated an online alarm flood classification application, making the link between offline alarm analytics and online support explicit [39]. Sekiou et al. applied supervised machine learning to alarm prediction and classification in an industrial gas plant, showing how conventional labeled learning can be used when plant-specific training data are available [41].

Manca and Fay introduced convolutional kernel-based classification for alarm floods, using kernel methods to compare structured alarm sequences without relying only on hand-crafted similarity rules [42]. Tamascelli et al. used word2vec for online classification, learning distributed representations of alarm tokens so that

statistically similar alarm tags can support classification [43]. The deep word-embedding study in [47] similarly treated process alarms as language-like sequences, using embeddings to predict alarms and capture contextual relationships among tags. Hu et al. used word embedding and few-shot learning for root-cause identification of industrial alarm floods, addressing the common situation in which only a few labeled examples exist for each root cause [44]. The online pattern matching and prediction study in [45] focused on predicting incoming alarm floods as they develop, while Yang et al. estimated remaining time to alarm using variation rates and mixture entropies, extending prediction from event class to time-to-alarm information [46]. Classification and prediction should therefore be evaluated by earliness, stability, confidence calibration, and ability to abstain, not only by final accuracy after the complete flood is observed.

The reviewed classification studies also show a trade-off between interpretability and flexibility. Template-based methods give operators a recognizable pattern, but they may fail when the current flood deviates from historical templates. Embedding and kernel methods can tolerate more variation, but their decisions are harder to explain unless they expose influential alarms and similar training cases. Semisupervised and few-shot methods reduce labeling requirements, but they still require careful validation because industrial labels are often noisy or incomplete. A practical online classifier should therefore combine three outputs: a ranked list of candidate classes, evidence explaining why each class is plausible, and a confidence or abstention signal indicating whether the case is sufficiently familiar. This output structure is important because episode recognition should provide structured evidence for subsequent RCA and decision support, rather than merely producing a single predicted label.

Across these method families, different assumptions lead to different operational trade-offs. Sequence-alignment and template-based methods exploit repeated temporal or symbolic structures and provide relatively interpretable case evidence, whereas itemset-based methods relax sequence-order requirements but may lose temporal information. Embedding- and kernel-based methods can tolerate greater sequence variability, but their decisions are generally less transparent and depend more strongly on representative historical data. Few-shot and semisupervised approaches reduce labeling requirements, although their reliability remains sensitive to label quality and distribution shift.

Prediction also needs to be tied to operational consequence. Predicting the next alarm tag may be mathematically precise but operationally weak if the operator needs to know whether the plant will trip,

whether a safety function is involved, or which procedure should be opened. Future prediction studies should therefore move from tag-level likelihood to consequence-aware prediction, such as escalation risk, time to critical alarm, expected operator action, and the next diagnostic test that would distinguish competing hypotheses. This consequence-aware view provides the link from alarm flood analytics to the next operational stage, where RCA explains why the recognized episode occurred and whether the explanation is physically, temporally, and causally plausible.

3 Root-Cause Diagnosis

Root-cause diagnosis converts alarm evidence into explanations. In the operational-stage map in Fig. 1, RCA represents the transition from alarm flood episode recognition to causal explanation. Alarm flood analytics identifies episode structures, recurring patterns, similar cases, and likely episode classes. This distinction is important because a statistically frequent or similar alarm pattern may still be operationally misleading if it violates topology, protection logic, process dynamics, temporal order, or known causal constraints. Table 2 summarizes the main diagnostic method families.

3.1 Physically constrained RCA in power-system and SCADA fault diagnosis

Power-system alarm processing is a mature RCA setting because alarms are tied to topology, protection devices, breakers, and relay logic. This group of methods illustrates why RCA cannot be reduced to alarm correlation or historical similarity. A diagnostic candidate must be consistent with protection zones, breaker states, relay logic, topology, and event timing. Cho and Park used fuzzy relations for fault-section diagnosis, representing uncertain relationships between observed alarms and candidate fault sections [2]. De Souza et al. used a neuro-fuzzy approach, combining expert reasoning with learning-like adaptation for alarm processing in electrical power systems [49]. Guo et al. introduced

temporal constraint networks, testing whether alarm order and timing are consistent with a candidate power-system fault [3]. Wei et al. developed an online intelligent alarm-processing system for digital substations, showing how substation automation can support real-time alarm interpretation [50]. Wang et al. formulated MILP-based fault diagnosis in active distribution networks, explicitly optimizing over fault candidates and alarm evidence [12]. Zhang et al. proposed a quantitative representation of alarm information for power-grid fault diagnosis, converting qualitative alarm messages into structured diagnostic evidence [4].

These power-system methods show why RCA cannot be reduced to pattern matching. A statistically likely alarm pattern is not acceptable if it contradicts breaker states, protection zones, or temporal feasibility. Oh et al. reviewed power-system alarm processing and argued that future systems should move beyond overreliance on fault analysis toward anomaly detection, explainable AI, and operator support [19]. De La Cruz et al. reviewed fault location for distribution smart grids, showing that diagnosis increasingly depends on sparse measurements, distributed resources, and changing grid topology [64]. These reviews indicate that false alarms, missing alarms, delayed messages, and topology changes should be first-class objects in RCA evaluation.

The evolution of power-system alarm processing also shows why older expert and fuzzy systems remain relevant. Deep learning can learn statistical relationships from alarm data, but protection logic and topology often define hard constraints that should not be violated. For example, a relay alarm may be missing because of communication loss, and a breaker alarm may be false because of stale state reporting. A diagnostic model should reason about these evidence defects explicitly rather than forcing all observed alarms into a single learned pattern. Hybrid models that use learning for candidate generation and engineering constraints for plausibility checking are therefore more appropriate than fully unconstrained models. This hybrid design reflects the operational-stage view behind the operational-stage synthesis adopted in this paper: episode-recognition models may generate candidates, while RCA should test

Table 2 Root-cause diagnosis methods and the alarm evidence they use.

Method group	Evidence used	Diagnostic output	Main limitation
Physically constrained RCA models	Relay/breaker alarms, protection records, topology, temporal order	Fault section, false/missing alarm assessment, temporal feasibility [2–4,12,19,49,50]	Engineering models require maintenance when topology or settings change.
Information-theoretic and process-causal RCA	Alarm variables, process variables, time delays, sensor data	Causal direction, root variable, propagation path [18,30,51–55]	Causality estimates depend on sampling, stationarity, and operating regime.
Knowledge-augmented RCA	Knowledge graphs, manuals, causal graphs, domain constraints, LLM reasoning	Grounded root-cause explanation and causal chain [13,14,56–58]	Knowledge sources can become stale and must be versioned.
Anomaly and multimodal evidence for RCA	Time series, building data, grid measurements, images, logs	Anomaly location, fault candidate, action category [20,59–74]	Scores and descriptions must be converted into actionable alarm evidence.

whether those candidates remain admissible under physical, logical, and temporal constraints.

Another practical requirement is temporal granularity. Protection and SCADA events may arrive with different latencies and time stamps. Temporal-constraint networks and online substation systems are useful because they treat time order as diagnostic evidence, but they also need tolerance windows for delayed messages. Future power-system RCA benchmarks should therefore include both event values and timing uncertainty. Without timing uncertainty, an RCA method may overstate causal confidence when the observed alarm order is affected by communication delay or asynchronous logging.

3.2 Information-theoretic and process-industrial causality and RCA

Process-industrial RCA often uses alarm logs together with process variables to infer propagation. This subsection represents the move from co-occurrence-based alarm association to causal hypothesis generation. Alinezhad et al. reviewed alarm RCA in process industries and summarized common methods, recent research status, and challenges, thereby clarifying that RCA involves causal inference, temporal reasoning, and operator interpretability rather than only classification [18]. Yu and Yang used transfer entropy to detect causality between process variables based on industrial alarm data, providing a directional measure of information transfer [52]. The transfer-entropy cause-effect analysis in [51] similarly used information flow to infer relationships among alarm variables. Ji et al. used time-delayed mutual information for real-time fault diagnosis, addressing the delay between root causes and observed process responses [55]. Liu et al. combined Liang–Kleeman information flow with graphical lasso, using information-theoretic causality and sparse graphical structure to identify root causes [54]. The hybrid causal-diagnosis study in [53] combined alarm flood classification, transfer entropy, and multi-sensor fusion, illustrating how classification outputs can be refined by causal evidence.

The main strength of these methods is that they try to distinguish propagation from co-occurrence. Their limitation is that causality estimates are sensitive to sampling rate, time-delay selection, operating mode, stationarity, and unobserved variables. Therefore, their outputs should be interpreted as causal hypotheses rather than final causal explanations. A causal path inferred from historical data should therefore be checked against process knowledge and operator evidence before it is used as an operational recommendation.

A key advantage of information-theoretic RCA is that

it can use continuous process variables and alarm events together. This is important because the true cause of an alarm flood may begin as a gradual deviation in a process variable before discrete alarms appear. However, causal measures can be confounded by control loops and operator interventions. A controller may create negative feedback that masks the original propagation direction, while an operator action may introduce a new causal path after the abnormality has already started. RCA studies should therefore report whether data were collected before or after operator intervention and whether control actions were included in the model. This reporting is essential for judging whether the inferred propagation reflects the physical fault process, the control response, or human intervention after the event.

The reviewed causal methods also suggest that RCA should not be evaluated only by whether the final root variable is correct. A causal-chain estimate can be partially useful even if the first root cause is uncertain, because it may still identify the affected unit, propagation direction, or variables that require inspection. Future evaluation should therefore consider ranked root causes, path-level correctness, and the usefulness of the next recommended check. Such evaluation better matches the operational role of RCA, where the goal is to guide investigation under uncertainty rather than simply output a single root-cause label.

3.3 Knowledge-augmented RCA and interpretable causal chains

Knowledge-augmented RCA attempts to make diagnostic outputs semantically grounded. This stage aims to turn causal hypotheses into supported explanations by linking alarms, process variables, equipment, topology, procedures, and domain knowledge. Sun et al. integrated knowledge graphs and LLMs for industrial process anomaly RCA, using structured domain relationships to constrain and support language-model reasoning [14]. Chen et al. proposed sequence alignment and dual-graph fusion with manifold regularization, combining sequence evidence and graph structure for root-cause identification of industrial alarm floods [13]. Yang et al. developed KDAFL, a knowledge-data fusion method for interpretable anomaly causal-chain construction in process industries, explicitly targeting causal-chain explanations rather than only root-cause labels [57]. Zheng et al. studied fine-tuning pretrained LLMs for fault diagnosis of complex systems, showing that language models can support diagnostic tasks when adapted to domain-specific failure descriptions [56]. Chen et al. reviewed large models for machine monitoring and fault diagnostics, emphasizing that foundation models create

opportunities for diagnosis but also introduce challenges in data grounding, reliability, and deployment [58].

Knowledge-augmented RCA is promising because alarm tags refer to equipment, loops, procedures, and physical assets. The main risk is stale or incomplete knowledge. If a topology graph, procedure library, or knowledge graph is outdated, the system may generate explanations that sound authoritative but are no longer valid. Therefore, knowledge-augmented RCA should report which knowledge sources were used, their versions, and any conflicts between learned evidence and engineering constraints. This requirement connects knowledge-augmented RCA to lifecycle governance, because diagnostic validity depends on whether the underlying knowledge sources match the current plant configuration and procedure versions.

Knowledge-augmented RCA also changes the form of explanation. Instead of giving only a label such as “pump fault” or “line fault”, a system can give a chain: observed alarms, supporting variables, inferred propagation, equipment relationship, retrieved procedure, and rejected alternatives. This is closer to how engineers investigate incidents. However, it increases the responsibility of the model because each link in the chain must be supported. LLM-based RCA should therefore be used to organize and verbalize evidence, while graph and engineering constraints should control whether a proposed explanation is admissible. In this setting, LLMs should be positioned as explanation organizers and evidence interfaces rather than standalone causal reasoners.

A useful design principle is conflict visibility. If a sequence model suggests one root cause but a knowledge graph or topology model rejects it, the system should expose the disagreement. Such conflict may reveal a missing alarm, a stale topology model, a novel event, or an incorrect retrieved document. In safety-critical alarm management, uncertainty and disagreement are not failures to hide; they are signals for human review. Making such conflicts visible also prevents a fluent diagnostic narrative from masking weak, stale, or contradictory evidence.

3.4 Adjacent anomaly detection and multimodal evidence for RCA

Visual and multimodal anomaly-detection studies are treated here primarily as adjacent or transferable evidence rather than direct demonstrations of industrial alarm-management capability. Anomaly detection can provide upstream evidence for alarms or RCA, but it should not be treated as RCA by itself. Its outputs become operationally relevant only when the detected abnormality can be mapped to an affected asset, severity,

alarm definition or priority, diagnostic hypothesis, uncertainty, and subsequent operational response. Fan et al. analytically investigated autoencoder-based unsupervised anomaly detection for building energy data, using reconstruction error to identify abnormal consumption patterns [59]. Moghaddass and Sheng proposed a Bayesian hierarchical anomaly-detection framework for dynamic systems, representing uncertainty across time-varying operating conditions [60]. Sapountzoglou et al. developed a generalizable and sensor-independent deep learning method for fault detection and location in low-voltage distribution grids, targeting transfer across sensor layouts [61]. Himeur et al. reviewed AI-based anomaly detection for building energy consumption and emphasized weather, occupancy, and schedule dependence [20]. Gaggero et al. reviewed AI and physics-based anomaly detection in smart grids, highlighting the need to combine data-driven detection with physical consistency [62]. Maican et al. reviewed fault detection and diagnosis methods in power plants, showing that power-plant FDD requires full diagnostic pipelines rather than isolated anomaly scores [63].

Asset-specific studies further connect anomalies to alarm applications. Teixeira et al. applied intelligent multi-agents to reduce false alarms in wind turbine monitoring, showing how multiple agents can filter or validate abnormal events before they reach operators [75]. Zhang et al. proposed a data-driven intelligent alarm method for ultra-supercritical thermal power units, converting plant data into intelligent alarm evidence [76]. Castillo-Navarro et al. developed a methodological framework for managing wind-turbine SCADA alarms for failure analysis, linking alarms to maintenance and failure investigation [77]. BERT-based industrial automation anomaly detection used log modeling and edge sensor computing to detect abnormal automation events close to the data source [74]. Multimodal fault detection in Industry 4.0 used large language models to combine visual, textual, and sensor evidence for diagnostic interpretation [73].

Recent multimodal anomaly detection studies broaden the evidence available for RCA. These studies can be broadly understood from three complementary perspectives. One line of research focuses on combining visual and textual information to improve anomaly localization and explanation. For example, VMAD uses visual-enhanced multimodal LLMs to localize and describe previously unseen industrial anomalies [65], while EIAD links visual defects with textual interpretations to improve the explainability of industrial anomaly detection [66]. A second line introduces stronger domain knowledge or specialized model

architectures to improve robustness and operational relevance. Representative approaches include the expert-guided region-of-interest tokenizer and manufacturing-process knowledge used in Triad [67], the multi-expert architecture for enhancing multimodal LLM robustness [69], and the customization of general visual-language foundation models for industrial anomaly detection and reasoning [70]. More recent studies further extend multimodal anomaly detection toward explicit reasoning and action-oriented outputs. The detect-classify-act framework connects detected anomalies directly to action categories [68], whereas AnomalyR1 introduces GRPO-based end-to-end multimodal reasoning [71] and LR-IAD incorporates mask-free logical reasoning to reduce dependence on pixel-level annotations [72]. Collectively, these developments move multimodal anomaly detection from visual defect recognition toward more interpretable and reasoning-oriented diagnostic evidence. However, their value for RCA ultimately depends on whether such evidence can be translated into alarm-system terms, including the affected component, severity, uncertainty, operational consequence, and recommended check.

The reviewed anomaly-detection and multimodal studies are broader than traditional alarm floods because their inputs include time series, logs, images, text, and sensor streams. Their relevance to alarm management depends on whether the abnormality evidence is operationalized. A reconstruction error, heat map, or textual anomaly description is not yet an alarm. It becomes alarm evidence only after the system assigns a component, severity, confidence, and action implication. This conversion step is often underreported, but it is essential for deployment.

Another issue is false positive burden. An anomaly detector that produces many low-severity deviations may recreate the same alarm overload problem that traditional alarm systems already suffer from. Therefore, anomaly-based alarm generation should include suppression rules, operating-mode awareness, maintenance-window awareness, and operator feedback. The objective should not be maximum anomaly sensitivity in isolation, but useful alarm generation under workload constraints. This requirement links anomaly-based evidence generation back to the broader alarm-management lifecycle: detection outputs must be filtered, contextualized, and governed before they become operator-facing alarms or RCA evidence.

4 Alarm-System Decision Support

Alarm-system decision support converts alarm and diagnostic evidence into information that operators,

engineers, and maintenance teams can use. In the operational-stage map in Fig. 1, this section represents the transition from diagnosis to operator action. At this stage, the output of alarm AI is no longer only a class label, anomaly score, root-cause candidate, or causal path. It becomes an operator-facing communication and decision-support artifact. It includes HMI design, procedure retrieval, alarm summaries, report generation, cyber-physical alert prioritization, and LLM/agent workflows. A useful decision-support artifact should distinguish observed alarms, inferred causes, retrieved procedures, recommended checks, uncertainty, and authority boundaries.

To clarify the evidence inputs to this stage, Fig. 2 summarizes the evidence flow across the three main research streams. Alarm flood analytics transforms raw alarms into structured episode-level evidence, RCA refines this evidence into ranked causal hypotheses and supporting explanations, and decision support translates the resulting diagnostic evidence into operator-facing information and recommended checks. The feedback path further indicates how operator verification and actions can update upstream data, models, and knowledge. Table 3 summarizes typical decision-support roles and risks.

4.1 Human-machine interfaces, smart alarms, and operator-facing reports

Decision support begins before LLMs. This subsection focuses on the communication layer of alarm intelligence, where diagnostic outputs must be transformed into information that operators can verify and act on. Bantay and Abonyi studied machine-learning-supported HMI design, showing that AI can help structure the information displayed to operators rather than merely generating alarm labels [78]. Smart alarm evaluation for Industry 4.0 also treated the alarm as a human-facing object whose value depends on whether operators can understand and act on it [27]. Dalzell et al. applied LLMs to SCADA alarm management for power grids, using language models to summarize and interact with dense SCADA alarm information [15]. Santoso et al. generated structured alarm reports from industrial logs using LLMs and RAG, showing how retrieved evidence can be converted into report-like outputs for the pulp and paper industry [16]. Ongwae described an industrial LLM application for asset integrity and process safety management, illustrating how language models can support safety documentation and operational knowledge work [80].

These studies indicate that the output of an alarm AI system is not just a prediction. It is a communication artifact that may influence operator attention and action.

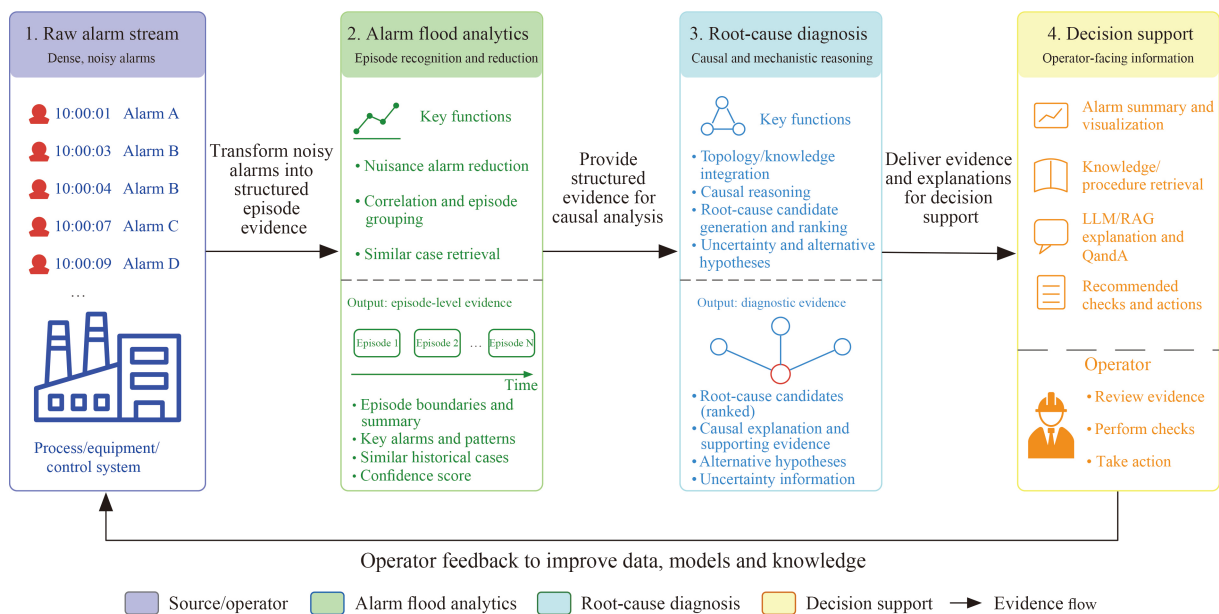


Fig. 2 Illustrative evidence flow across alarm flood analytics, root-cause diagnosis, and operator-facing decision support.

Table 3 Decision-support functions for alarm systems and their risk boundaries.

Function	Typical method	Useful output	Risk boundary
Operator display and HMI	Smart alarm evaluation, HMI design, alarm-quality analytics	Prioritized views, clearer alarm grouping, reduced cognitive load [27,28,78]	Interface changes should be evaluated with operators, not only by offline metrics.
Alarm reports and summaries	LLM summarization, RAG, structured log-to-report generation	Evidence-linked narratives and incident summaries [15,16,79–81]	Generated text must distinguish facts, inferences, and recommendations.
Evidence-grounded LLM/RAG and agentic alarm workflows	LLM agents, multi-agent systems, retrieval and checking modules	Coordinated retrieval, analysis, diagnosis, and report drafting [17,21,82–88]	Agents need constrained tools, logging, approval, and rollback.
Cyber–physical alerts	IDS, protocol anomaly detection, asset graphs, contextual LLM reasoning	Prioritized alerts based on operational consequence [81,89–94]	Cyber scores must be linked to physical assets and control authority.

A report or summary should therefore separate observed alarms, inferred causes, retrieved procedures, and recommended checks. It should also cite evidence and state uncertainty when the diagnosis is incomplete. A generated alarm report should not merge facts, inferences, and recommendations into a single fluent narrative. It should explicitly separate observed alarm facts, diagnostic hypotheses, supporting evidence, missing evidence, retrieved procedures, recommended checks, and uncertainty.

A useful way to evaluate operator-facing reports is to ask whether they reduce verification effort. Operators should be able to trace a summary sentence back to an alarm tag, timestamp, measurement, procedure, or diagnostic output. If a generated report says that a fault likely originated in one unit, it should identify the evidence supporting that claim and the evidence that remains missing. This requirement is stronger than ordinary text quality and is central to alarm decision support. Therefore, report evaluation should include

evidence traceability, unsupported-claim rate, citation precision, and whether operators can verify the recommendation efficiently.

HMI and report generation should also support layered users. Operators need concise priorities and immediate checks; control engineers need causal or logical traces; maintenance teams need recurrence and asset-history evidence; managers may need a post-event narrative. The same alarm evidence should therefore be rendered at different levels of detail rather than forced into a single summary. This layered design is important because the same RCA output may require different forms of explanation for real-time operation, engineering review, maintenance planning, and post-event governance.

4.2 Evidence-grounded LLM, RAG, and multi-agent decision workflows

LLM and RAG methods are increasingly used to aggregate alerts, retrieve procedures, and coordinate

decision workflows. In alarm management, these methods should be framed as evidence-grounded interfaces for retrieval, summarization, explanation, and report drafting rather than autonomous decision makers. Recent studies show a progression from task-specific LLM applications toward more coordinated alarm-management workflows. At the task level, LLMs have been used to consolidate large volumes of operational alerts into higher-level incident information [79] and to support alarm-setting analysis in equipment health models [82]. Building on these individual functions, RAG and multi-agent architectures introduce explicit separation between retrieval, reasoning, and decision-support tasks. For example, the blended RAG-enhanced framework in [17] separates retrieval and reasoning across agents, while Mingyue emphasizes human-AI collaborative decision making under explicit safety constraints in complex power grids [83]. Similarly, LEMAD uses agent decomposition to support anomaly detection and complex service monitoring in power-grid environments [84].

A related line of research extends LLM-based decision support toward adaptive anomaly reasoning and evidence grounding. AAD-LLM demonstrates how LLMs can support adaptive anomaly detection rather than serving only as text-generation tools [85]. RAAD-LLM further incorporates RAG so that anomaly interpretation can be grounded in retrieved information [86]. Collectively, these developments indicate a shift from isolated LLM functions toward workflows that combine alert aggregation, retrieval, anomaly reasoning, and coordinated decision support. Recent reviews of LLM applications in smart grids and electrical power systems further highlight both the operational opportunities of these approaches and the associated requirements for safety, reliability, and domain-specific constraints [21,87,88].

LLM/RAG decision workflows should therefore be positioned as evidence-grounded interfaces. They can summarize, retrieve, draft, and explain, but they should not independently suppress alarms, modify thresholds, or issue operational commands without approval. Tool calls, retrieved documents, prompt templates, model versions, and final recommendations should be logged for audit. This boundary is critical because a fluent LLM output can appear authoritative even when the retrieved evidence is incomplete, stale, or contradictory. Therefore, LLM/RAG systems should expose the evidence used for each claim and should clearly mark unsupported hypotheses, uncertainty, and recommended human checks.

The distinction between RAG and fine-tuning is important in this setting. Fine-tuning can adapt a model

to alarm language, protocol descriptions, and diagnostic vocabulary, but it does not guarantee that the model knows the current plant configuration or procedure version. RAG can retrieve current documents, but retrieval errors can introduce stale or irrelevant evidence. Multi-agent designs can separate retrieval, reasoning, checking, and reporting, but they also create more tool calls and more points of failure. A practical alarm assistant may therefore combine all three: domain adaptation for vocabulary, RAG for current evidence, and agentic decomposition for task control. However, each retrieved source, tool call, reasoning step, and final output should remain auditable, because decision support in alarm systems requires traceability rather than only conversational usefulness.

LLM evaluation should include negative cases. A useful assistant should refuse to answer when the retrieved evidence is insufficient, when documents conflict, when an instruction in a log looks like prompt injection, or when a requested action exceeds authority. Without such refusal behavior, a language model can increase risk even when most summaries are correct. Thus, evaluation should include refusal correctness, evidence sufficiency, contradiction handling, prompt-injection resistance, and authority-boundary compliance, in addition to ordinary text-quality metrics. Evaluation should also report the knowledge base used, retrieval quality, source and citation correctness, and whether adversarial or prompt-injection cases were explicitly tested.

4.3 Cyber-physical alert interpretation and industrial security

Cyber-physical alerts are decision-support problems because cybersecurity events matter operationally only when they affect equipment, measurements, controls, or safety functions. This subsection extends decision support from process and equipment alarms to cyber-physical consequences, where alert priority should depend on operational impact rather than only cyber anomaly scores. Ma et al. proposed programmable intrusion detection for distributed energy resources in cyber-physical networked microgrids, connecting intrusion detection to power-system assets [89]. Zhao et al. fine-tuned Llama3 for anomaly detection in oilfield industrial control systems, using protocol communication fingerprints to identify abnormal devices or behaviors [90]. Manowska and Syta studied LLMs for protecting industrial IoT systems in critical infrastructure, showing how language models may assist monitoring and security reasoning [91]. Ning et al. discussed technical evolution and practice of industrial Internet security protection in

the era of LLMs, highlighting the interaction between industrial security mechanisms and language-model-based analysis [92]. Jaffal et al. reviewed LLMs in cybersecurity, covering applications, vulnerabilities, and defense techniques that are relevant to prompt injection and alert reasoning in alarm systems [93]. Sharma and Mehta proposed adaptive and explainable AI agents for anomaly detection in critical IoT infrastructure, using LLM-enhanced contextual reasoning for interpretable alerts [94]. Shah et al. introduced aLLarMa, an LLM-based application for operational data, alarms, and networks, directly linking alarm management and network evidence [81].

The main decision-support requirement is consequence reasoning. A network anomaly should be prioritized according to device criticality, control authority, measurement trustworthiness, and physical consequence, not only according to a cybersecurity score. This requirement creates a role for asset graphs, cyber-physical knowledge bases, and RAG systems that can retrieve relevant procedures while respecting authority boundaries. For example, the same protocol anomaly may require different responses depending on whether the affected device is a monitoring-only sensor, a protection relay, a controller, or an actuator with direct process authority.

Cyber-physical decision support also introduces adversarial risk. Alarm logs, network messages, ticket descriptions, and retrieved documents may contain untrusted text. If an LLM-based alarm assistant follows malicious instructions embedded in those sources, the assistant may ignore alarms, reveal sensitive information, or recommend unsafe actions. This makes prompt-injection resistance part of alarm-system safety, not only a cybersecurity add-on. The safer architecture is to treat logs and retrieved documents as data, never as instructions, and to restrict action tools through whitelists and human approval. This safety boundary should be enforced across the whole decision-support pipeline, including retrieval, summarization, agent tool use, recommendation generation, and report drafting.

5 Future Directions for Deployable Alarm Intelligence

This section summarizes future research directions for moving AI-driven alarm management from offline analytics to deployable alarm intelligence. Building on Fig. 1, the discussion highlights future work needed to address the cross-cutting enablers of deployable alarm intelligence across compression, episode recognition, RCA, and decision support. Although lifecycle governance is shown as Stage 6 for organizational clarity, it should be understood as a continuing function that supports monitoring, version control, change validation, and operator feedback throughout deployment and subsequent system change. Table 4 summarizes the main deployment bottlenecks, corresponding technical directions, and evaluation evidence discussed in this section.

5.1 Episode-level benchmarks and replay protocols

A major methodological issue in this field is that many papers evaluate the part of the workflow that is easiest to measure rather than the part that matters most to operators. Alarm rationalization papers often report alarm-rate reduction, but they may not report whether critical alarms remain visible. Classification papers often report final accuracy, but they may not report whether the correct class was available early enough for action. RCA papers may report the correct root cause, but they may not report whether the inferred path was physically plausible or whether false and missing alarms were handled. LLM papers may report examples of fluent summaries, but they may not quantify unsupported claims. Therefore, future evaluation should move from isolated model metrics to episode-level evidence that reflects how alarm intelligence is used during operation. A stronger benchmark culture should align metrics with the operational use of each method.

Future benchmarks should evaluate alarm episodes

Table 4 Deployment bottlenecks and future directions for deployable alarm intelligence.

Bottleneck	Current gap	Needed technical direction	Evaluation evidence
Alarm flood analytics	Many studies classify complete floods or mine site-specific patterns	Partial-sequence replay, hierarchical representations, calibrated abstention [8–11,35,38,40,43–45]	Earliness, stability, retrieved-case quality, operator workload
Root-cause diagnosis	Correlation, classification, and causality are often disconnected	Hybrid RCA combining sequence, topology, temporal, causal, and knowledge constraints [2–4,12–14,51,55,57]	False/missing-alarm tolerance, path correctness, expert validation
Decision support	Fluent summaries can hide weak or stale evidence	Evidence-grounded RAG, versioned retrieval, authority boundaries, prompt-injection defenses [15–17,81,83,93]	Citation precision, refusal correctness, audit logs, human studies
Cross-site transfer	Alarm tags, procedures, thresholds, and equipment names are site-specific	Semantic representations based on equipment hierarchy, operating mode, protection zone, process connectivity, and causal direction [18,19,58]	Cross-site replay, mapping quality, degradation analysis, expert validation
Lifecycle governance	Models degrade after alarm, topology, sensor, or document changes	Drift monitoring, model/version control, rollback, operator feedback loops [1,19,21,58,88,95]	Shadow deployment, post-change validation, field KPIs

rather than isolated samples. For alarm-flood analytics, datasets should include raw alarm sequences, operating modes, event labels, operator actions, maintenance periods, and verified outcomes. For RCA, datasets should include topology snapshots, protection settings, process variables, causal annotations, false alarms, and missing alarms. For decision support, datasets should include manuals, procedures, incident reports, retrieved evidence, and operator evaluations. Such benchmarks should also record version information because alarm procedures, topology, and equipment status can change. Without versioned context, an apparently correct explanation may be invalid for the actual time of the event. Ordinary accuracy is not enough. Alarm-flood methods should report earliness, prediction stability, similarity quality, and abstention. RCA methods should report causal-path correctness, missing-alarm tolerance, and topology consistency. LLM/RAG methods should report citation precision, unsupported-claim rate, refusal correctness, prompt-injection resistance, and auditability. These metrics should be reported at the episode level so that alarm reduction, episode recognition, RCA, and decision support can be evaluated as connected operational functions rather than isolated algorithmic tasks.

For synthesis purposes, the validation evidence reported in the reviewed literature is broadly distinguished into four levels: historical/offline validation, simulator-based evaluation, shadow deployment, and live or controlled field validation. Offline replay is useful for comparing algorithms on historical data, but it cannot measure how operators react to advice. Simulator evaluation can measure workload, response time, and trust calibration, but it may not reproduce the full stress of a live control room. Shadow deployment can run a model on real streams without affecting operation, which is useful for assessing drift and false positives. Live deployment provides the strongest evidence, but it requires rollback, approval, and post-deployment monitoring. Review papers should encourage authors to state clearly which stage their evidence belongs to. This distinction is essential for moving from offline analytics to deployable alarm intelligence, because a method that performs well in replay may still fail under operator workload, changing topology, or incomplete evidence.

5.2 Uncertainty, false alarms, and missing-alarm reasoning

Alarm uncertainty has several forms: uncertainty about whether an alarm is true, whether an expected alarm is missing, which fault caused a sequence, whether a retrieved document is valid, and whether the model is outside its training distribution. These uncertainties

should not be collapsed into a single confidence value. Future alarm intelligence should represent uncertainty at multiple levels, including alarm truth, missing evidence, episode familiarity, root-cause plausibility, causal-path validity, retrieval reliability, and action safety.

False and missing alarms should be treated as first-class reasoning objects rather than as noise to be removed before modeling. In practical control rooms, a missing relay alarm, delayed breaker status, stale measurement, or false cyber alert can change the interpretation of the whole event. RCA methods should therefore reason about which expected alarms are absent, which observed alarms may be false, and how these evidence defects affect candidate causes. Decision-support systems should expose this uncertainty to operators instead of hiding it behind a single diagnosis.

This requirement also affects evaluation. Alarm-flood classifiers should report calibrated confidence and abstention when the current episode is unfamiliar. RCA methods should test robustness under injected false alarms, removed alarms, delayed messages, and conflicting evidence. LLM/RAG systems should state when retrieved documents are insufficient, stale, or contradictory. A deployable system should downgrade confidence, ask for human review, or recommend additional checks when the evidence base is incomplete.

5.3 Cross-site transfer and semantic generalization

Cross-site transfer is unresolved because alarm tags and procedures are site-specific. Future systems should separate local identifiers from transferable structures such as equipment hierarchy, operating mode, protection zone, process connectivity, and causal direction. This is important because two sites may use different tag names, alarm priorities, thresholds, and procedures for similar equipment functions, while similar tag names may correspond to different process roles.

A model trained at one plant should not be assumed valid at another plant simply because the equipment names look similar. Transfer should be justified by explicit mappings between equipment hierarchy, process functions, operating modes, and alarm semantics. For alarm flood analytics, this suggests hierarchical representations that move from tag-level sequences to equipment-level, unit-level, and process-area-level episode signatures. For RCA, this suggests causal and topology-aware representations that can separate local nomenclature from physical propagation. For decision support, this suggests retrieval systems that can map site-specific procedures to shared operational concepts while preserving local authority boundaries.

Future studies should evaluate cross-site transfer

explicitly rather than only reporting performance on one plant or one dataset. Useful evidence includes performance degradation across sites, quality of tag-to-equipment mappings, robustness to changed alarm settings, and expert validation of transferred episode templates or causal explanations.

5.4 Evidence-grounded LLM/RAG and prompt-injection protection

LLM-assisted alarm management must distinguish communication from control. Low-risk uses include summarizing alarms, retrieving procedures, and drafting reports for review. Medium-risk uses include explaining diagnostic candidates and suggesting next checks. High-risk uses include alarm suppression, threshold tuning, and operational recommendation. Direct control or automatic alarm reconfiguration should require formal approval, rollback, and safety constraints. Therefore, LLM/RAG should be framed as an evidence-grounded interface for retrieval, explanation, summarization, and report drafting, not as an autonomous decision maker.

Another important direction is the design of evidence objects shared between diagnostic modules and LLM interfaces. Instead of passing free-text alarm logs directly to a language model, the system should pass structured evidence: alarm identifiers, timestamps, confidence scores, missing-alarm hypotheses, retrieved documents, diagnostic candidates, and rejected alternatives. This structure reduces hallucination and makes the final explanation auditable. It also makes it easier to test the assistant because each generated statement can be checked against a specific evidence object. A decision-support output should explicitly separate observed alarms, inferred causes, retrieved procedures, recommended checks, uncertainty, and authority boundaries.

Prompt injection is a specific risk because alarm logs and retrieved documents may contain untrusted text. Future LLM/RAG systems should use constrained tools, source validation, action whitelists, retrieval versioning, and human approval for high-risk outputs. Logs, tickets, network messages, and retrieved documents should be treated as data rather than instructions. The assistant should refuse or escalate when the retrieved evidence is insufficient, when sources conflict, when a document is stale, when an embedded instruction attempts to override the system role, or when a requested action exceeds operational authority.

Finally, LLM-assisted alarm management should be evaluated with adversarial and failure-oriented tests. These tests should include stale documents, conflicting procedures, missing alarms, injected instructions in logs,

ambiguous user requests, and out-of-scope action requests. A safe assistant should not only answer correctly in normal cases; it should refuse, ask for review, or downgrade confidence when the evidence is insufficient or unsafe. Evaluation should therefore include citation precision, unsupported-claim rate, refusal correctness, prompt-injection resistance, audit-log completeness, and human verification effort.

5.5 Lifecycle governance and human-centered deployment

Lifecycle governance is equally important. Alarm limits, topology, sensors, manuals, and procedures change over time, so deployed models should monitor drift, missing tags, retrieval quality, model confidence, operator feedback, and version consistency. Without lifecycle governance, a model that was valid during offline evaluation may become unsafe after alarm rationalization, topology reconfiguration, sensor replacement, procedure revision, or operating-mode change.

A practical governance mechanism should define what happens after drift is detected. Low-risk drift may trigger recalibration or additional review; high-risk drift should disable automatic recommendations and move the system into advisory mode. Governance should also include model and data versioning, change logs, rollback procedures, post-change validation, operator feedback loops, and periodic review of retrieved knowledge sources. These controls are especially important for LLM/RAG systems because document versions, retrieval indexes, prompts, and model checkpoints can all affect the generated recommendation.

Human-centered deployment should be treated as part of technical validation. Offline replay can show whether a model recognizes past episodes, but it cannot show whether operators trust, verify, ignore, or over-rely on the output. Simulator studies, shadow deployment, and controlled field trials are needed to evaluate workload, response time, trust calibration, explanation usefulness, and post-event learning. The final objective is not maximum model autonomy, but reliable alarm intelligence that supports human operators under uncertainty and remains valid across the alarm-management lifecycle.

6 Conclusions

This paper reviewed AI-driven industrial alarm management from a coupled operational perspective rather than as a collection of independent AI techniques. The synthesis shows that alarm flood analytics, root-

cause diagnosis, and decision support form a connected evidence-processing workflow: alarm analytics structures noisy alarm information into episode-level evidence, diagnosis converts this evidence into physically and causally plausible explanations, and decision support translates the resulting evidence into operator-facing information. The six-stage framework further places these functions within shared engineering foundations and lifecycle governance, highlighting that the reliability of downstream decisions depends on the quality, context, and traceability of upstream evidence.

From this synthesis, a central conclusion is that deployable alarm intelligence should be modular but not fragmented. An RCA method should provide a testable causal or physical explanation, not only a root-cause name. An LLM assistant should cite evidence and state uncertainty, not only produce fluent text. LLM/RAG systems should therefore be positioned as evidence-grounded interfaces for retrieval, explanation, summarization, and report drafting, rather than autonomous decision makers. Future progress will depend on episode-level benchmarks, uncertainty-aware models, false and missing alarm reasoning, cross-site semantic representations, evidence-grounded LLM/RAG workflows, prompt-injection protection, and lifecycle validation with operators in the loop. Overall, AI-driven alarm management should be understood through an operational-stage view of deployable alarm intelligence, where accuracy, evidence traceability, uncertainty communication, operational safety, and lifecycle governance must be considered together.

Data Availability No datasets were generated or analyzed for this work.

Acknowledgments This work was supported in part by The Australian Research Council (grant DP240100830 and IC240100010). The authors used ChatGPT 5.5 (online version) as a language-editing tool to improve grammar, clarity, and readability. The AI tool was not used for data analysis, data generation, image processing, or scientific content creation. All outputs were carefully reviewed and verified by the authors, who take full responsibility for the final content.

Competing interests The authors declare no conflicts of interest.

References

1. Wang J D, Yang F, Chen T W, et al. An overview of industrial alarm systems: Main causes for alarm overloading, research status, and open problems. *IEEE Transactions on Automation Science and Engineering*, 2016, 13(2): 1045–1061
2. Cho H J, Park J K. An expert system for fault section diagnosis of power systems using fuzzy relations. *IEEE Transactions on Power Systems*, 1997, 12(1): 342–348
3. Guo W X, Wen F S, Liao Z W, et al. An analytic model-based approach for power system alarm processing employing temporal constraint network. *IEEE Transactions on Power Delivery*, 2010, 25(4): 2435–2447
4. Zhang X, Du M X, Wang Y X, et al. Research on power grid fault diagnosis based on a quantitative representation of alarm information. *IEEE Transactions on Industrial Electronics*, 2023, 70(9): 9582–9592
5. Stauffer T, Clarke P. Using alarms as a layer of protection. *Process Safety Progress*, 2016, 35(1): 76–83
6. Hollender M, Skovholt T C, Evans J. Holistic alarm management throughout the plant lifecycle. In: 2016 Petroleum and Chemical Industry Conference Europe. Berlin, Germany, 2016, 1–6
7. Brown D C C. Alarm system performance/one size fits all? *Measurement and Control*, 2003, 36(4): 120–123
8. Ahmed K, Izadi I, Chen T W, et al. Similarity analysis of industrial alarm flood data. *IEEE Transactions on Automation Science and Engineering*, 2013, 10(2): 452–457
9. Hu W K, Chen T W, Shah S L. Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods. *IEEE Transactions on Industrial Electronics*, 2018, 65(9): 7290–7300
10. Shang J, Chen T W. Early classification of alarm floods via exponentially attenuated component analysis. *IEEE Transactions on Industrial Electronics*, 2020, 67(10): 8702–8712
11. Zhou B Y, Hu W K, Brown K, et al. Generalized pattern matching of industrial alarm flood sequences via word processing and sequence alignment. *IEEE Transactions on Industrial Electronics*, 2021, 68(10): 10171–10179
12. Wang C Y, Pang K Y, Shahidepour M, et al. MILP-based fault diagnosis model in active power distribution networks. *IEEE Transactions on Smart Grid*, 2021, 12(5): 3847–3857
13. Chen N, Hu W K, Shang J, et al. Sequence alignment and dual-graph fusion-based manifold regularization for root-cause identification of industrial alarm floods. *IEEE Transactions on Industrial Informatics*, 2026, 22(4): 3066–3077
14. Sun Q, Li Y H, Zhou C J, et al. Root cause analysis for industrial process anomalies through the integration of knowledge graph and large language model. In: 2024 43rd Chinese Control Conference. Kunming, China, 2024, 6855–6860
15. Dalzell G, Liu C, Shah S, et al. Enhancing SCADA alarm management for power grids using large language models. In: IECON 2024-50th Annual Conference of the IEEE Industrial Electronics Society. Chicago, USA, 2024, 1–6
16. Santoso H, Wijaya O H, Andriani F, et al. From logs to insights in the pulp & paper industry: generating structured alarm reports using LLMs and RAG. *Journal Teknik Informatika (JUTIF)*, 2025, 6(5): 4022–5236
17. Wang H, Wang Z Y, Wang L, et al. Blended RAG-enhanced LLM multi-agent framework for anomaly detection and alert management. In: 2025 9th International Conference on Smart Grid and Smart Cities. Chengdu, China, 2025, 217–222
18. Alinezhad H S, Roohi M H, Chen T W. A review of alarm root cause analysis in process industries: common methods, recent research status and challenges. *Chemical Engineering Research and Design*, 2022, 188: 846–860

19. Oh J Y, Yoon Y T, Sohn J M. A comprehensive review of alarm processing in power systems: addressing overreliance on fault analysis and projecting future directions. *Energies*, 2024, 17(13): 3344
20. Himeur Y, Ghanem K, Alsalemi A, et al. Artificial intelligence based anomaly detection of energy consumption in buildings: a review, current trends and new perspectives. *Applied Energy*, 2021, 287: 116601
21. Amjad F, Korötko T, Rosin A. Review of LLMs applications in electrical power and energy systems. *IEEE Access*, 2025, 13: 150951–150969
22. Xiong W Q, Wang J D, Chen K. Multivariate alarm systems for time-varying processes using Bayesian filters with applications to electrical pumps. *IEEE Transactions on Industrial Informatics*, 2018, 14(2): 504–513
23. Goel P, Pistikopoulos E N, Mannan M S, et al. A data-driven alarm and event management framework. *Journal of Loss Prevention in the Process Industries*, 2019, 62: 103959
24. Rao H R M, Zhou B Y, Brown K, et al. Alarm correlation analysis with applications to industrial alarm management. *Control Engineering Practice*, 2024, 143: 105812
25. Rahaman H, Alinezhad H S, Al-Dabbagh A W, et al. Cyclic mining of alarm and operator action events for enhanced process operation. *Control Engineering Practice*, 2024, 153: 106069
26. Hu W K, Chen T W, Shah S L. Discovering association rules of mode-dependent alarms from alarm and event logs. *IEEE Transactions on Control Systems Technology*, 2018, 26(3): 971–983
27. Chang C W. Evaluation of smart alarm systems for industry 4.0 technologies. *Applied Sciences*, 2020, 10(6): 2022
28. Bezerra A, Silva I, Guedes L A, et al. Extracting value from industrial alarms and events: a data-driven approach based on exploratory data analysis. *Sensors*, 2019, 19(12): 2772
29. Kaced R, Kouadri A, Baiche K, et al. Multivariate nuisance alarm management in chemical processes. *Journal of Loss Prevention in the Process Industries*, 2021, 72: 104548
30. Zhang Y F, Qu H K, Liu Y, et al. Transfer entropy-based causal inference for industrial alarm overload mitigation. *Electronics*, 2025, 14(20): 4066
31. Guo C, Hu W K, Lai S Q, et al. An accelerated alignment method for analyzing time sequences of industrial alarm floods. *Journal of Process Control*, 2017, 57: 102–115
32. Manca G, Dix M, Fay A. Clustering of similar historical alarm subsequences in industrial control systems using alarm series and characteristic coactivations. *IEEE Access*, 2021, 9: 154965–154974
33. Wang J, Li H G, Huang J W, et al. A data similarity based analysis to consequential alarms of industrial processes. *Journal of Loss Prevention in the Process Industries*, 2015, 35: 29–34
34. Lai S Q, Chen T W. A method for pattern mining in multiple alarm flood sequences. *Chemical Engineering Research and Design*, 2017, 117: 831–839
35. Zhou B Y, Hu W K, Chen T W. Pattern extraction from industrial alarm flood sequences by a modified CloFAST algorithm. *IEEE Transactions on Industrial Informatics*, 2022, 18(1): 288–296
36. Charbonnier S, Bouchair N, Gayet P. Fault template extraction to assist operators during industrial alarm floods. *Engineering Applications of Artificial Intelligence*, 2016, 50: 32–44
37. Niyazmand T, Izadi I. Pattern mining in alarm flood sequences using a modified PrefixSpan algorithm. *ISA Transactions*, 2019, 90: 287–293
38. Rahimi Y, Rao H R M, Zhou J, et al. Online alarm flood classification via interpretable template extraction and structured convolutional matching. *Computers & Chemical Engineering*, 2026, 208: 109570
39. Lucke M, Chioua M, Grimholt C, et al. Advances in alarm data analysis with a practical application to online alarm flood classification. *Journal of Process Control*, 2019, 79: 56–71
40. Alinezhad H S, Shang J, Chen T W. Early classification of industrial alarm floods based on semisupervised learning. *IEEE Transactions on Industrial Informatics*, 2022, 18(3): 1845–1853
41. Sekiou S, Behloul A, Nait-Said R, et al. Alarms prediction and classification in industrial processes using supervised machine learning techniques: a case study in an Algerian gas plant. *Computers & Chemical Engineering*, 2026, 204: 109378
42. Manca G, Fay A. Convolutional kernel-based classification of industrial alarm floods. *Data-Centric Engineering*, 2024, 5: e30
43. Tamascelli N, Rao H R M, Cozzani V, et al. Online classification of alarm floods using a word2vec algorithm. In: *IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society*. Singapore, Singapore, 2023, 1–6
44. Hu W K, Yang G, Li Y P, et al. Root cause identification of industrial alarm floods using word embedding and few-shot learning. *IEEE Transactions on Industrial Informatics*, 2024, 20(2): 1465–1475
45. Lai S Q, Yang F, Chen T W. Online pattern matching and prediction of incoming alarm floods. *Journal of Process Control*, 2017, 56: 69–78
46. Yang Z, Wang J, Li H, Gao S. Pre-warning for the remaining time to alarm based on variation rates and mixture entropies. *Entropy*, 2025, 27(7): 736
47. Cai S, Palazoglu A, Zhang L B, et al. Process alarm prediction using deep learning and word embedding methods. *ISA Transactions*, 2019, 85: 274–283
48. Alinezhad H S, Shang J, Chen T W. A modified bag-of-words representation for industrial alarm floods. In: *2022 IEEE International Symposium on Advanced Control of Industrial Processes*. Vancouver, Canada, 2022, 331–336
49. de Souza J C S, Meza E M, Schilling M T, et al. Alarm processing in electrical power systems through a neuro-fuzzy approach. *IEEE Transactions on Power Delivery*, 2004, 19(2): 537–544
50. Wei L H, Guo W X, Wen F S, et al. An online intelligent alarm-processing system for digital substations. *IEEE Transactions on Power Delivery*, 2011, 26(3): 1615–1624
51. Hu W K, Wang J D, Chen T W, et al. Cause-effect analysis of industrial alarm variables using transfer entropies. *Control Engineering Practice*, 2017, 64: 205–214

52. Yu W J, Yang F. Detection of causality between process variables based on industrial alarm data using transfer entropy. *Entropy*, 2015, 17(8): 5868–5887
53. Shirshahi A, Aliyari-Shoorehdeli M. Diagnosing root causes of faults based on alarm flood classification using transfer entropy and multi-sensor fusion approaches. *Process Safety and Environmental Protection*, 2024, 181: 469–479
54. Liu X D, Liu J, Yang X H, et al. Fault root cause analysis based on Liang–Kleeman information flow and graphical lasso. *Entropy*, 2025, 27(2): 213
55. Ji C, Ma F Y, Wang J H, et al. Real-time industrial process fault diagnosis based on time delayed mutual information analysis. *Processes*, 2021, 9(6): 1027
56. Zheng S W, Pan K, Liu J, et al. Empirical study on fine-tuning pre-trained large language models for fault diagnosis of complex systems. *Reliability Engineering & System Safety*, 2024, 252: 110382
57. Yang T N, Zhu M R, Ji Y J, et al. KDACL: a knowledge-data fusion-driven method for interpretable anomaly causal chain construction in process industries using large language models. *Journal of Intelligent Manufacturing*, 2026: 1–35
58. Chen X F, Lei Y G, Li Y F, et al. Large models for machine monitoring and fault diagnostics: opportunities, challenges, and future direction. *Journal of Dynamics, Monitoring and Diagnostics*, 2025, 4(2): 76–90
59. Fan C, Xiao F, Zhao Y, et al. Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. *Applied Energy*, 2018, 211: 1123–1135
60. Moghaddass R, Sheng S W. An anomaly detection framework for dynamic systems using a Bayesian hierarchical framework. *Applied Energy*, 2019, 240: 561–582
61. Sapountzoglou N, Lago J, De Schutter B, et al. A generalizable and sensor-independent deep learning method for fault detection and location in low-voltage distribution grids. *Applied Energy*, 2020, 276: 115299
62. Gaggero G B, Girdinio P, Marchese M. Artificial intelligence and physics-based anomaly detection in the smart grid: a survey. *IEEE Access*, 2025, 13: 23597–23606
63. Maican C A, Pană C F, Pătrașcu-Pană D M, et al. Review of fault detection and diagnosis methods in power plants: algorithms, architectures, and trends. *Applied Sciences*, 2025, 15(11): 6334
64. De La Cruz J, Gómez-Luna E, Ali M, et al. Fault location for distribution smart grids: literature overview, challenges, solutions, and future trends. *Energies*, 2023, 16(5): 2280
65. Deng H L, Luo H C, Zhai W, et al. VMAD: visual-enhanced multimodal large language model for zero-shot anomaly detection. *IEEE Transactions on Automation Science and Engineering*, 2026, 23: 3607–3618
66. Zhang Z Y, Ruan J C, Gao X, et al. EIAD: explainable industrial anomaly detection via multi-modal large language models. In: *2025 IEEE International Conference on Multimedia and Expo*. Nantes, France, 2025, 1–6
67. Li Y Z, Yuan S H, Wang H L, et al. Triad: empowering LMM-based anomaly detection with expert-guided region-of-interest tokenizer and manufacturing process. In: *Proceedings of 2025 IEEE/CVF International Conference on Computer Vision*. Honolulu, USA, 2025, 21917–21926
68. Mokhtar S, Mousakhan A, Galesso S, et al. Detect, classify, act: categorizing industrial anomalies with multi-modal large language models. In: *Proceedings of 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. Nashville, USA, 2025, 4058–4067
69. Chen Z L, Imani F. A multi-expert framework for enhancing multimodal large language models in industrial anomaly detection. *Pattern Recognition*, 2026, 172: 112752
70. Xu X H, Cao Y K, Zhang H X, et al. Customizing visual-language foundation models for multi-modal anomaly detection and reasoning. In: *2025 28th International Conference on Computer Supported Cooperative Work in Design*. Compiegne, France, 2025
71. Chao Y H, Liu J, Tang J, et al. AnomalyR1: a GRPO-based end-to-end MLLM for industrial anomaly detection. *arXiv preprint: arXiv: 2504.11914*, 2025
72. Zeng P J, Pang F Y, Wang Z B, et al. LR-IAD: mask-free industrial anomaly detection with logical reasoning. *arXiv preprint: arXiv: 2504.19524*, 2025
73. Alsaif K M, Albeshri A A, Khemakhem M A, et al. Multimodal large language model-based fault detection and diagnosis in context of industry 4. 0. *Electronics*, 2024, 13(24): 4912
74. Gao T N. Industrial automation anomaly detection based on BERT and edge sensor computing. *Neural Computing and Applications*, 2025, 37(34): 28671–28689
75. Teixeira W C E, Sanz-Bobi M Á, de Oliveira R C L. Applying intelligent multi-agents to reduce false alarms in wind turbine monitoring systems. *Energies*, 2022, 15(19): 7317
76. Zhang X F, Ye L H, Zhang C, et al. Data-driven-based intelligent alarm method of ultra-supercritical thermal power units. *Processes*, 2024, 12(5): 889
77. Castillo-Navarro J, Kristjanpoller F, Mena R, et al. A methodological framework for managing the alarms in wind turbine control and data acquisition systems for failure analysis. *Machines*, 2024, 12(9): 597
78. Bántay L, Abonyi J. Machine learning-supported designing of human-machine interfaces. *Applied Sciences*, 2024, 14(4): 1564
79. Zha J J, Shan X W, Lu J X, et al. Leveraging large language models for efficient alert aggregation in AIOPs. *Electronics*, 2024, 13(22): 4425
80. Ongwae B. An industrial application of a large language model to enhancing asset integrity and process safety management. *Journal of Artificial Intelligence, Virtual Reality, and Human-Centered Computing*, 2025, 1(1): 1–13
81. Shah S, Kelly A, Tripathy S, et al. aLLarMa: LLM-based application for operational data, alarms and networks. *Journal of LaTeX Class Files*, 2020, 18(9): 1–10
82. Yang W C, Yang X H, Ren H B, et al. Development of an intelligent agent for optimizing alarm deviation in equipment health models based on large language models. In: *2025 5th International Conference on Intelligent Power and Systems*. Xi'an,

- China, 2025, 1–10
83. Cai X L, Meng Z J, Guo Q, et al. Mingyue: a multi-agent system for human-AI collaborative decision-making under safety constraints in complex power grids. In: 2025 5th Power System and Green Energy Conference. Hong Kong, China, 2025, 434–439
 84. Ji X, Zhang L, Zhang W Y, et al. LEMAD: LLM-empowered multi-agent system for anomaly detection in power grid services. *Electronics*, 2025, 14(15): 3008
 85. Russell-Gilbert A, Sommers A, Thompson A, et al. AAD-LLM: adaptive anomaly detection using large language models. In: 2024 IEEE International Conference on Big Data. Washington, USA, 2024, 4194–4203
 86. Russell-Gilbert A. RAAD-LLM: adaptive anomaly detection using LLMs and RAG Integration. Starkville: Mississippi State University, 2025
 87. Madani S, Tavasoli A, Astaneh Z K, et al. Large language models integration in smart grids. *Energy Reports*, 2025, 14: 1562–1577
 88. Zhang L D, Ji W L, Zhao Y L, et al. Review on application and challenges of large language models in power grids. *IEEE Access*, 2025, 13: 106932–106941
 89. Ma S Y, Li Y, Du L, et al. Programmable intrusion detection for distributed energy resources in cyber-physical networked microgrids. *Applied Energy*, 2022, 306: 118056
 90. Zhao J M, Jin Z W, Zeng P, et al. An anomaly detection method for oilfield industrial control systems fine-tuned using the Llama3 model. *Applied Sciences*, 2024, 14(20): 9169
 91. Manowska A, Syta J. Application of large language models in the protection of industrial IoT systems for critical infrastructure. *Applied Sciences*, 2026, 16(2): 730
 92. Ning B W, He K, Zong X J, et al. Technical evolution and practice of industrial internet security protection in the era of large language models. In: 2025 International Conference Automatics and Informatics. Varna, Bulgaria, 2025, 94–99
 93. Jaffal N O, Alkhanafseh M, Mohaisen D. Large language models in cybersecurity: a survey of applications, vulnerabilities, and defense techniques. *AI*, 2025, 6(9): 216
 94. Sharma R, Mehta M. Adaptive and explainable AI agents for anomaly detection in critical IoT infrastructure using LLM-enhanced contextual reasoning. In: Nguyen G N, Swaroop A, Shukla P, eds. *Proceedings of Fifth International Conference on Computing and Communication Networks*. Cham: Springer, 2026, 247–266
 95. Goel P, Datta A, Mannan M S. Industrial alarm systems: challenges and opportunities. *Journal of Loss Prevention in the Process Industries*, 2017, 50: 23–36



Chen Liu received the M.S. degree in information technology, the B.S. degree (Hons.) in Computer Science, and the Ph.D. degree in Electrical and Electronic Engineering from RMIT University, Melbourne, VIC, Australia, in 2015, 2016, and 2021, respectively. He is currently a Research Fellow with the School of Engineering, RMIT University. His research interests include smart grid optimization, machine learning, large language models (LLMs), industrial artificial intelligence, and urban computing. He has authored or co-authored more than 35 research publications in leading AI conferences and *IEEE Transactions* journals.



Wei Peng received the BEng degree in Electrical Engineering from Wuhan University of Technology, Wuhan, China, in 1992, the Master's degree in information science from the University of New South Wales, Sydney, Australia, in 2001, and the P.h.D degree from the University of Sydney, Sydney, Australia, in 2007, with a focus on cognitive agents. He is currently a Principal Research Fellow in Industrial Artificial Intelligence with the School of Engineering, RMIT University, Melbourne, Australia. He has around 20 years of experience in academic research and industrial AI development. His research interests include generative and agentic AI, large language models, cognitive and multi-agent systems, and industrial decision support. Dr. Peng has published more than 80 peer-reviewed papers and holds multiple AI patents.



Xinghuo Yu is a Distinguished Professor and an Associate Deputy Vice-Chancellor at RMIT University (Royal Melbourne Institute of Technology), Melbourne, Australia. His research interests are in control systems, intelligent and complex systems, and power and energy systems. He is a Fellow of Australian Academy of Science, an Honorary Fellow of Engineers Australia, and a Fellow of IEEE, IFAC, Australian Computer Society and Australian Institute of Company Directors. He served as the President of *IEEE Industrial Electronics Society* in 2018 and 2019. He received a number of awards and honours for his contributions, including 2018 M A Sargent Medal from Engineers Australia, 2018 Australasian AI Distinguished Research Contribution Award from Australian Computer Society, and 2013 Dr.-Ing. Eugene Mittelman Achievement Award from *IEEE Industrial Electronics Society*.