

Rag-Based Chatbots for Iot Device Troubleshooting: A Review

Pankaj Rajesh Bhakde, Prof. Abhishek Popat Bangar, Prof. S. B. Bhosale, Prof. V. A. Karad

Department of Computer Engineering
Jaihind College of Engineering, Kuran, Maharashtra, India

Abstract — IoT deployments bring together a wide mix of sensors, gateways, embedded devices, and software services. Finding the cause of a fault, or setting up a device the technician has never seen before, calls for knowledge that is typically spread across datasheets, manuals, incident tickets, and past reports. This review looks at retrieval-augmented generation (RAG) and large language model (LLM) techniques for IoT troubleshooting, using five selected studies that cover incident diagnosis and reporting, technical-data extraction, agentic answer validation, an industrial troubleshooting assistant, and conversational machine monitoring. Across these works, retrieval anchors generated answers to outside evidence, structured extraction makes technical specifications easier to query, and validation steps cut down on mix-ups between similarly named fields and on answers the evidence does not support. The reported evaluations show measurable improvements in technical question answering and task performance, yet they also expose weaknesses with hard questions, varied document layouts, source reliability, and generalisation to new deployments. We compare the methods and contributions of the studies, point out research gaps, and outline an evidence-centred architecture that joins document ingestion, structured device knowledge, hybrid retrieval, sensor-context reasoning, answer verification, and technician feedback. Our main conclusion is that a dependable troubleshooting chatbot cannot rest on fluent text generation alone: its advice has to be traceable to device-specific evidence and bounded by technical rules. Since the review draws on five chosen papers rather than an exhaustive database search, it is offered as a focused narrative review, not a systematic one.

Keywords— Internet of Things, retrieval-augmented generation, chatbot, troubleshooting, large language model, agentic RAG, technical datasheets.

I. INTRODUCTION

IoT systems now play a role in industrial monitoring, smart buildings, healthcare, agriculture, and connected infrastructure. Their spread-out sensors and devices produce a steady stream of operational data, while the constraints that govern them sit in manuals, datasheets, maintenance tickets, and engineering records. When something fails, a technician has to match the symptoms they see against this scattered knowledge. Conventional support is often slow and leans heavily on a few experienced specialists.

Large language models (LLMs) make technical information reachable through conversation, but they may state things that no device manual supports, or confuse specifications that have similar names. Retrieval-augmented generation (RAG) tackles part of this by fetching relevant external material before the answer is written. Even so, the weaknesses of these systems go beyond language: they involve how evidence is gathered, represented, retrieved, checked, and finally turned into useful and safe diagnostic advice.

The chosen literature approaches these questions from several angles. Yuan et al. (2025) describe a RAG system that diagnoses and reports IoT incidents. Kong et al. (2026b) pull technical data out of datasheets, and Kong et al. (2026a) build on that with agentic answer validation. Löwhagen et al. (2025) test an industrial troubleshooting assistant with working technicians, while Jeon et al. (2026) deal with conversational machine monitoring that retrieves data in real time.

This review sets out to: (i) summarise the selected studies and what they found; (ii) compare their methods, contributions, and scope; (iii) identify shared challenges and possible gaps in this body of work; and (iv) suggest research directions along with an evidence-centred architecture for a safer and more transparent troubleshooting assistant. Throughout, the chatbot is seen as a tool that supports technicians, not one that replaces their judgement or accountability.

II. REVIEW METHODOLOGY AND SELECTION CRITERIA

This paper is a focused narrative review of five chosen studies: an incident diagnosis and reporting system for IoT (Yuan et al.,

2025); a generative-AI tool for extracting technical data (Kong et al., 2026b); an agentic RAG method for validating answers to datasheet questions (Kong et al., 2026a); an empirical study of an industrial troubleshooting assistant (Löwhagen et al., 2025); and a conversational machine-monitoring paper known only through its corrigendum (Jeon et al., 2026). They were picked because each bears directly on IoT incident analysis, technical-document retrieval, device-configuration question answering, troubleshooting support, or LLM-assisted monitoring.

Four criteria guided the selection: (1) a direct link to RAG- or LLM-based troubleshooting of IoT or industrial devices; (2) a clearly identifiable problem, technical approach, or evaluation; (3) publication in a conference proceeding or scholarly journal; and (4) a contribution to at least one of the review's main themes. For every work we noted the problem, the technical approach, the evidence from evaluation, and the limitations the authors state. No further papers were added. The set is small and varied, and no comprehensive database search, search strings, or formal screening were carried out. The work is therefore not a systematic review. We do not claim that all relevant literature has been gathered, and results from different datasets are not treated as directly comparable. The fifth source is a corrigendum and not the original article, so we rely on it only for what the notice itself says.

III. LITERATURE REVIEW

1. Incident diagnosis and report generation

Yuan et al. (2025) present RAIDR, a retrieval-augmented LLM system for diagnosing and reporting IoT incidents. It relies on detected anomalies and a graph of sensor relationships to describe incident timelines and dependencies. Incident signatures are used to pull up similar past tickets and supporting material, such as sensor descriptions, operating manuals, and earlier reports, after which the LLM writes an incident report and a root-cause hypothesis. Its main strength lies in combining sensor context with historical operational knowledge, instead of handling an anomaly as a stand-alone text query. Because it is a demonstration paper, it depends on the quality of the anomaly detection, the sensor relationships, and the historical records, and it cannot by itself show how well the approach performs across different deployments.

2. Structured extraction from technical datasheets

Kong et al. (2026b) propose a generative-AI tool that converts device specifications found in PDF or HTML datasheets into a schema-normalised form. The extracted information is stored in a local vector database so that configuration questions can be

answered from the documents, including questions about devices unseen during model training. On a benchmark of 960 question-answer pairs, Recall improved from 0.636 to 0.926 and Accuracy from 0.595 to 0.807 relative to ChatPDF. The study shows that the quality of RAG hinges on more than retrieval and generation; the completeness and consistency of the underlying technical data matter too. Extraction errors and layout variation remain real boundaries.

3. Agentic RAG and answer validation

Kong et al. (2026a) extend datasheet question answering with an agentic workflow that picks its tools according to the type of question. The workflow combines structured fact lookup, top-k text retrieval, source checking, unit and specification validation, and rule-based compatibility comparison. Over 20 datasheets and 1,000 questions, the correct-answer rates were 0.958 for easy, 0.969 for medium, and 0.613 for hard questions, against 0.686, 0.549, and 0.273 for conventional top-k RAG. The gap that remains on hard questions indicates that multi-step compatibility reasoning is still difficult even when the evidence is structured and checked.

4. Industrial troubleshooting assistant

Löwhagen et al. (2025) assessed a multilingual, user-centred RAG assistant in the setting of train commissioning. Their controlled field experiment had 19 technicians complete 173 tasks. Participants who used the AI assistant performed significantly better than those who did not, with the largest gains among less experienced technicians; familiarity with AI and attitudes toward it also shaped the outcomes. This is empirical evidence that a troubleshooting assistant can improve practical task performance, and a reminder to designers that usability, training, and user trust affect real-world effectiveness. The small sample and single industrial setting limit how far the findings generalise.

5. Conversational monitoring and real-time retrieval

Jeon et al. (2026) is a corrigendum that identifies the original study, “ChatCNC: Conversational machine monitoring via large language model and real-time data retrieval augmented generation,” whose terminology was later changed to “CNC-Talks.” The correction states that the scientific content, methodology, and conclusions were unaffected. This line of work matters because troubleshooting often needs a conversational interface tied to current machine data, not just static documentation. However, the supplied file is a corrigendum and not the full article, so detailed performance claims cannot be judged from it alone.

IV. COMPARATIVE ANALYSIS OF EXISTING STUDIES

The five studies tackle related problems but differ in inputs, methods, and evaluation settings. Their findings complement one another and are not interchangeable. Table 1 summarises the scope, main contribution, and boundary of each.

Table 1. Comparative summary of the five selected studies

Study	Method / scope	Key finding and boundary
Yuan et al. (2025)	RAIDR: sensor anomalies, dependency graph, and incident tickets; similar cases are retrieved and an LLM drafts the report	Links sensor context to reports; relies on graph and historical-data quality; demonstration paper.
Kong et al. (2026b)	Schema-based extraction from PDF/HTML datasheets, local vector database, and RAG; 960 QA pairs	Accuracy rose from 0.595 to 0.807 over ChatPDF; extraction errors and layout variation still matter.
Kong et al. (2026a)	Agentic RAG with structured facts, retrieval, source checks, and compatibility rules; 20 datasheets, 1,000 questions	Better at every difficulty level; accuracy on hard questions stays at 0.613.
Löwhagen et al. (2025)	Multilingual RAG chatbot for train commissioning; 19 technicians, 173 tasks	Controlled field evidence of better task performance; small sample, single context.
Jeon et al. (2026)	Corrigendum on conversational LLM monitoring with real-time RAG (as the title indicates)	The supplied source is only a correction notice; full evaluation details are not available.

The first theme is the gap between fluent generation and dependable answers: structured extraction and source checking give better results than plain retrieval (Kong et al., 2026a; Kong et al., 2026b). The second is context: incident diagnosis gains from sensor relationships and history (Yuan et al., 2025), and monitoring work highlights the value of real-time data (Jeon et

al., 2026). The third concerns human outcomes, because the field experiment shows better task performance alongside the influence of usability, training, and trust (Löwhagen et al., 2025). Lastly, difficult multi-step questions remain the weakest area (Kong et al., 2026a).

RAIDR concentrates on event-level diagnosis and reporting; technical extraction strengthens the knowledge base; agentic RAG adds explicit answer checks; and the industrial assistant measures effects on human task performance. Because the metrics come from different tasks and baselines, the numbers should not be read as a common ranking. Taken together, though, the studies point toward a layered architecture in which careful document preparation and evidence checks matter as much as the conversational model.

V. RESEARCH CHALLENGES AND IDENTIFIED GAPS

The points below are possible gaps suggested by the five selected papers. They do not prove that no other research has addressed these topics.

1. Evidence quality and provenance

A retrieved passage may be relevant and still fall short of justifying a recommendation. Technical answers need traceable evidence, including the device model, specification field, units, operating conditions, and source location. A chatbot should separate “not found in the documentation” from “the specification is wrong,” and should hold back when evidence is inadequate.

2. Heterogeneous documents and devices

Datasheets differ in layout, terminology, tables, diagrams, and units. Mistakes made during extraction can carry over into the vector index and surface later in answers. New or legacy devices may have incomplete documentation, and device-specific assumptions can turn a fluent, general answer into an unsafe one.

3. Complex multi-sensor incidents

Faults can spread across connected components and change over time. Diagnosing them therefore needs temporal context, sensor dependencies, asset topology, and historical cases. Text retrieval by itself may miss causal relationships or fail to tell a root cause from a correlated symptom.

4. Evaluation and operational deployment

Current evaluations span very different settings: question-answer benchmarks, document extraction, and controlled

industrial tasks. Shared metrics are needed for factual correctness, citation support, diagnostic usefulness, abstention quality, response time, and technician outcomes. Larger field trials are also needed across device types, languages, organisations, and failure conditions.

5. Safety, privacy, and maintenance

Troubleshooting advice can affect equipment, production, and personal safety, so uncertainty has to be communicated clearly. Industrial records may hold operationally sensitive information, and knowledge sources keep changing as manufacturers publish new datasheets and firmware updates alter device behaviour. Older incident reports may no longer apply.

VI. PROPOSED RESEARCH DIRECTIONS

On the basis of the comparison, the directions below may help address the gaps identified. The reviewed studies suggest that dependable IoT troubleshooting should be designed as a complete evidence pipeline and not as a language model attached to a document search engine.

Hybrid and context-aware retrieval: Combine keyword matching, semantic vector search, and structured database queries, and filter by device model, firmware version, sensor identity, alarm time, and operating conditions so that a similar but incompatible device is never treated as authoritative. Graph relationships can help separate the initiating fault from the symptoms that follow.

Evidence-linked answer generation: Tie each important statement to a source location, such as a datasheet page, table row, manual section, or historical ticket, and keep verified facts apart from hypotheses. When evidence is missing or contradictory, the system should ask for clarification, display the conflict, or abstain.

Rule-based compatibility checks: Compare supply voltage, signal range, interface type, pin mapping, operating temperature, and protocol settings against explicit rules once units have been normalised. The language model chooses the tool, while deterministic checks decide whether a comparison is valid.

Live telemetry and incident history: Retrieve recent alarms, match them with known incident signatures, examine sensor dependencies, and cite the relevant operating procedure. Live data should be permission-controlled, time-stamped, and screened for faulty readings.

Robust extraction: Build extraction methods that handle tables, diagrams, and scanned documents, with human review for low-confidence fields.

Abstention, escalation, and human oversight: Sort tasks by risk, require human approval before any action that could damage equipment or endanger people, and pass high-risk, low-confidence, or contradictory cases to qualified personnel.

Evaluation beyond answer accuracy: Measure evidence precision, factual correctness, citation validity, unsupported-claim rate, compatibility decision accuracy, time to diagnosis, first-time fix rate, and unnecessary escalations. Report them separately for easy, medium, and difficult cases, and include unseen devices, varied layouts, and conflicting sources.

Privacy-preserving and maintainable deployment: Consider role-based access, encryption, audit logs, version-aware indexing, scheduled re-ingestion, removal of obsolete documents, regression testing after updates, and validated multilingual translation of units and warnings.

To draw these directions together, Table 2 sketches a staged, evidence-centred pipeline for an IoT troubleshooting chatbot. It is a conceptual proposal derived from the literature and has not been tested experimentally.

Table 2. Proposed evidence-centred pipeline (not experimentally validated)

Stage	Function	Related studies
1. Ingestion	Manuals, datasheets, incident tickets, and sensor metadata are collected	Yuan et al. (2025)
2. Extraction	Schema-based facts stored with source page or section, units, and conditions	Kong et al. (2026b)
3. Hybrid indexing	Structured facts and text indexed together for hybrid retrieval	Kong et al. (2026a, 2026b)
4. Context enrichment	Device identity, recent telemetry, alarm history, and component relationships attached to each question	Yuan et al. (2025); Jeon et al. (2026)
5. Workflow selection	A question-specific route for lookup, diagnosis, or compatibility checking	Kong et al. (2026a)

Stage	Function	Related studies
6. Verification	Claims tested against retrieved evidence and deterministic technical rules	Kong et al. (2026a)
7. Presentation	Short answer with citations, confidence, and safe next steps; technician feedback collected	Löwhagen et al. (2025)

Future studies should keep measured findings separate from conceptual claims, and should not assume that benchmark gains in one setting carry over to every factory, device family, or maintenance workflow.

VII. CONCLUSION AND FUTURE SCOPE

This review examined five studies on RAG-based troubleshooting for IoT and industrial devices. Together they show a clear progression: from retrieving relevant documents, to preparing structured technical knowledge, to validating answers, and finally to embedding AI assistance in real troubleshooting work. RAG lessens reliance on what a model has memorised, while structured specifications and source checks help hold back technically misleading answers. Incident graphs and real-time machine context can further link a user's question to what the system is actually doing. Still, hard compatibility questions, heterogeneous documentation, data quality, and generalisation remain open problems.

The reviewed work supplies separate building blocks and not a single complete solution, and a fluent reply is not proof of technical correctness. The most promising path is not an unconstrained chatbot but a human-supervised assistant that coordinates tools, grounds its recommendations in traceable evidence, checks them against technical constraints, and escalates when it cannot resolve a case safely.

The review is limited by its small, hand-picked set of five papers, one of which is only a corrigendum, and it is not a full account of the field. The studies use different tasks and baselines, so their findings should not be read as one uniform result. A broader future review should search several scholarly databases, record search terms and screening decisions, and cover more real-world deployment work. Future research should give priority to difficult multi-step questions, unseen devices, larger and more varied IoT deployments, transparent handling of uncertainty, and measurable gains in both correctness and maintenance outcomes.

REFERENCES

1. Jeon, J., et al. (2026). Corrigendum to “ChatCNC: Conversational machine monitoring via large language model and real-time data retrieval augmented generation” [J Manuf Syst 79 (2025) 504–514]. *Journal of Manufacturing Systems*, 85, 248. <https://doi.org/10.1016/j.jmsy.2026.01.008>
2. Kong, D., Funabiki, N., Kyaw, H. H. S., Kotama, I. N. D., & Zhu, Z. (2026a). An AI answer-validation method using agentic RAG for datasheet inquiry for IoT application system deployment. *Future Internet*, 18, Article 442. <https://doi.org/10.3390/fi18080442>
3. Kong, D., Funabiki, N., Kyaw, H. H. S., Kotama, I. N. D., Zhu, Z., & Rahmadani, A. A. (2026b). A generative AI-based technical data extraction tool for IoT application systems. *Sensors*, 26(4), Article 1081. <https://doi.org/10.3390/s26041081>
4. Löwhagen, N., Schwendener, P., & Netland, T. (2025). Can a troubleshooting AI assistant improve task performance in industrial contexts? *International Journal of Production Research*. Advance online publication. <https://doi.org/10.1080/00207543.2025.2527368>
5. Yuan, P., Tang, L.-A., Liu, Y., Yuji, K., Sato, M., & Chen, H. (2025). Incident diagnosing and reporting system based on retrieval augmented large language model. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)* (pp. 29721–29722).