Journal of Information Technology and Applications Research
Volume 1 Nomor 1 May 2025
e-ISSN: 3089-8013; p-ISSN:3089-8021, Page 162-182
DOI: https://doi.org/10.63956/jitar.v1i1.65
Available online at: https://jitar.org/index.php/jitar/article/view/65

Automated Incident Runbook Generation via LLMs: An Open Framework and Benchmark

Akhil Reddy Mandadi
Independent Research, India
(Correspondence Email : akhilreddymandadi95@gmail.com)

Abstract

This is due to the fact that today's software architectures rely heavily on highly-distributed cloud-native systems, micro services architectures and automated operational environments which are constantly creating complex incidents. Incident runbooks are a critical component of production systems that lend structure to procedures and are of great help to engineers on diagnosis, mitigation and recovery. Traditional runbook creation is however still largely manual, involves lots of knowledge and effort from experts and the need to keep track of happenings in the infrastructure as it is integrated, configured and developed. This means that operational documentation is often ill-structured, out of date and hard to maintain. This study aims to create and assess a generative framework of incident runbooks (IRs) based on Large Language Models (LLMs). To achieve this, the following specific aims are set for this study: (1) To strip down incident artifacts (mortem, resolution transcript, or log from operational communication) into an actionable, systemized runbook and to provide an open benchmark for a systematic evaluation.

The proposed framework has three elements, all linked together. Structured incident feature extraction techniques extract important operational elements like symptoms, triggering events, root causes, mitigation activities, etc., and also the resolution timelines from incident artifacts. Second, extracted features are then fed into LLM-based procedural synthesis mechanism to generate runbook automatically. Third, a rubric-based framework for the validation of generated outputs using an LLM-as-judge evaluates the outputs based on the specified rubric attributes: completeness, coherence, correctness and operational usefulness. An open benchmark of over 200 publishable incident reports was developed to enable experimentation by drawing on reports from cloud service providers and software platforms. Several different LLM configurations were tried with synthesized gold-standard runbooks.

The results indicated that using a structured extraction along with an LLM-based synthesis approach enhances the procedural uniformity and contextual appropriateness of the generated runbooks. Results from the experiments showed that there are measurable differences in performance between the models evaluated and that the rubric-based approach to assessment is effective as an effective tool for assessing operational documentation quality.

The use of publicly available datasets of incidents and events will suffer from inconsistencies in incident reporting due to various factors, including differences in style, detail or level of completeness. At the same time, there can be variation in the quality of the data, which can affect the consistency of features extracted and runbooks generated. Moreover, the outputs generated are still dependent on the model's reasoning ability and the variations in prompts.

The proposed approach can significantly decrease documentation maintenance expenses, enhance incident readiness, speed up operational knowledge sharing, and enable scalable incident response processes within an enterprise.

This research circumvents the need for human experts to evaluate the quality of OpsDocs and presents a new validation method for OpsDocs: rubric validation using LLM-as-judge.

Keywords: Incident Response, Large Language Models, Runbook Automation, Site Reliability Engineering, AIOps, Incident Postmortems, LLM-as-Judge, Cloud Reliability, Operational Intelligence.

Received: April 28, 2025; Accepted: July 22, 2025; Published: October 01, 2025
*Corresponding author : akhilreddymandadi95@gmail.com

INTRODUCTION

Background of the Study

The modern enterprise computing landscape has undergone significant change in recent years, driven by widespread adoption of distributed architectures and microservices ecosystems, cloud-native computing environments, and large-scale automation technologies. Increasingly, there is a demand in organizations for a complex digital infrastructure to provide highly-available services, system resiliency, and continuity of business operations. On one hand, these technological advancements have greatly enhanced the scalability and flexibility of services, yet on the other, they have added to the complexities of operations and driven new issues on how to manage services at times of incidents. With the evolution of large systems, there is a significant amount of operational telemetry being produced in logs, traces, metrics, alerts and communication streams and now incident diagnosis and remediation are getting increasingly tricky.

Regrettably, operational incidents still happen despite having a much engineered environment. A service outage, configuration error, regression in a software, security breach, infrastructure drift, cascading failures or any other would be a great setback for the overall business continuity and trust from the users would deteriorate a lot. Systematic incidents handling practices therefore receives a lot of focus in contemporary reliability engineering practice. Quick diagnosis, correct mitigation procedures, and coordinated communication and structured operational knowledge transfer are all crucial factors in responding to incidents successfully.

Incident runbooks are among the fundamentals of operational resilience. Runbooks are a step-by step documentation of procedures in a fail or event operation context. These documents provide support for engineers to investigate problems, look for remediation routes, and follow pre-planned response scenarios. Documented procedural workflows are used more and more to achieve an operational consistency in knowledge repositories and incident response (IR) (Rodrigues, 2024). Likewise, enterprise digital ecosystems aim to foster structured reliability measures as core elements to ensure digital trust and service continuity (Kuppam, 2024).

Runbooks traditionally are manually created by system experts after an operation incident or infrastructure change. Manual authoring demands expertise of the field and an extensive maintenance process that’s validated through repeated processes. All such processes tend to be harder to conduct in large environments with changing service architectures. The documentation available for operations usually doesn't meet the needs of describing actual infrastructure. When systems are more dynamic, manual processes entail a heavy documentation burden.

New breakthroughs in artificial intelligence and in-process automation imply other possibilities to resolve those problems. Over the last few years, evolving AIOps systems have gained a range of capabilities from analytics to the addition of machine learning and automation in order to achieve proactive operational management. The article by Zhang (2024) presented an example of integrated pipelines for AIOps, using anomaly detection

to identify variability, graph-based root cause localization, and LLM-derived remediation artifacts, all of which pave the way towards intelligent operational systems. Likewise, Wang et al. (2025) showed that cloud-native observability systems can be integrated into LLM frameworks to enable automatic and intelligent diagnosis and remediation.

The field of operational automation has been further extended with Generative AI and Foundation Models. Large Language Models have significant capabilities in procedural synthesis, understanding the context, extracting information, and generating natural texts. The field of foundation models, as discussed in Thota (2022), creates new software engineering infrastructure that can help to revolutionize software engineering productivity and organizational processes. Similarly, Gershon et al., (2024) highlighted that the AI infrastructure of today is increasingly capable of supporting enterprise scale model deployment, and AI model operational integration.

Studies have been carried out in the fields of AI-backed remediation and autonomous operation. Sarda (2023) has shown the powered by large language models automatic remediation in microservices architectures, and Kakarla (2024) has discussed autonomous remediation approaches in DevSecOps environments. There are also intelligent features in self-healing cloud architectures that include predictive maintenance and issue resolution (Rodriguez et al., 2024; Sirimalla, 2024). All these developments imply that the day of highly-automated operational ecosystems is with us.

Even with these recent developments, the problem of converting "case" information about an incident into structured runbooks is still mostly unsolved. Procedural knowledge is available in extensive environment activities as in incident artifacts including post mortems, chat logs and resolution transcripts, but this knowledge is not leveraged and is often inadequately, if at all, structured. Transforming these heterogeneous sources into operational documents using LLMs thus appears a promising research paradigm with significant practical applications.

Problem Statement

When it comes to reliability engineering and operations, incident response documentation is an important resource. The best runbooks minimize uncertainty in reacting to situations, enhance the consistency of procedures, and maintain institutional knowledge. Yet, for traditional runbooks, manual authoring processes are still mainly applied and manual operations are needed that demand a lot of skill and manpower.

Manual documentation methods have several drawbacks. To begin with, infrastructure is becoming much more intricate. Modern cloud systems are distributed stores, use container orchestration, come with multi-cloud configurations and are very interdependent. It is cumbersome to continually keep these environments up-to-date with current versions of the procedures.

Second, there is often a fragmented collection of incident data in various data sources. Don't overlook information in post mortems, joint communication tools, ticket flow, chats, escalation logs and troubleshooting logs. There is extensive synthesis and interpretation that is necessary to assemble these scattered clues to create a procedural

documentation.

Thirdly, because of changes in systems, third party runbooks often become outdated. Fast deployment cycles and ongoing delivery pipelines create architectures in constant flux. Frequent deployment cycles and continuous delivery create architectures that are evolving all the time. Shrivastava and Srivastav (2024) observed that today's solution architectures are becoming more adaptive and continuously changing, and thus, more operational architectures are needed. As a result, static documentation can sometimes become out of date with respect to actual production.

Recent studies show the rising adoption of automation processes powered by artificial intelligence. Paduraru et al. (2025) explored automated response playbooks for cybersecurity with LLM, and Mao et al. (2025) discussed with agentic troubleshooting automation in incident management systems. While such techniques are great advancements, little research is conducted for generalized approaches with focused attention on automated incident runbook generation.

As a result, organizations have an enormous challenge to streamline their growing complexity while simultaneously providing better means for recording and maintaining the processes in a reliable way. If you don't have scalable solutions then documentation debt and operational inefficiencies could keep growing.

Research Gap

The application of artificial intelligence in operational management and automation has been progressively increasing in the research domain. The current literature shows significant advances in anomaly detection, anomaly root cause analysis, autonomous remedies, observability systems and AI-based infrastructure management. However, there are a few missing areas that are not sufficiently resolved.

While the current studies have focused mainly on the analytical functions, procedural knowledge in terms of knowledge synthesis has not been sufficiently highlighted. However, Zhang (2024) dealt with the integrated systems of abnormality detection and root cause analysis, and Wang et al. (2025) investigated the observability-enhanced diagnosis system. While these tools enhance fault detection and diagnosis, they offer limited assistance for creating structured operational guidance.

Likewise, self-directed remediation studies tend to be more concerned with remediation actions than with generating documentation. Additionally, Kakarla (2024) explored automated DevSecOps remediation mechanisms and Sarda (2023) studied auto-remediation in microservices environments. These approaches deal with operational actions and not with structured procedural documentation adequately.

Recent research has explored the use of automation in incident management. Troubleshooting automation frameworks were proposed by Mao et al. (2025) and evaluation mechanisms for AI agents for real-life tasks in the IT space were presented by Jha et al. (2025). Specialized benchmarking contexts for the creation of incident runbooks are yet to be available, though.

The generation of synthetic data has also gained the interest of many researchers.

Galadima et al. (2024) explored the use of LLM-generated synthetic cyber incident process logs, showing the promise of augmenting operational datasets. To date, however, there is no publicly available benchmark for automated runbook generation systems.

It is also important for knowledge management studies to focus on the need to retain operational knowledge in the cybersecurity context (Rodrigues, 2024). However, frameworks which integrate knowledge extraction, procedural synthesis, benchmark development and quality assessment in unified architectures are scarce.

The literature thus identifies a number of gaps:

There is limited research exploring the topic of automated incident runbook generation.

1. Lack of reference data to evaluate the runbook.
2. Lack of uniform quality assessment systems.
3. Little exploration of LLM-as-judge for operational papers
4. Poor connection between assets and procedure-based systems.

This study is motivated by the desire to fill these gaps.

Aim and Objectives

The purpose of this research is to design and test an open framework for the automated generation of a runbook for incidents based on LLMs.

The study aims at achieving the following goals:

1. To create a developed extraction framework that is structured to accomplish identification of salient features from incident artifacts.
2. To create procedural runbooks using a technique of LLM-based synthesis.
3. To create an open, public incident post mortem and gold standard runbooks.

Research Questions

To answer the above research questions, the following study questions will be answered.

RQ1: Does LLM's ability to produce an incident runbook from disparate incident artifacts yield operationally useful information?

RQ2: What is the effect of structured feature extraction on the quality of runbook generation?

RQ3: Are LLM-as-judge evaluation approaches effective and reliable to evaluate operational document quality?

Contributions

This research has some important contributions to the literature and to operational practice. The study's primary contribution is the introduction of an open end-to-end system that combines structured incident extraction, LLM procedural synthesis, and validation procedures. The study firstly introduces an open end-to-end framework that includes three parts: structured incident extraction, LLM procedural synthesis and validation mechanisms. The proposed architecture doesn't just focus on anomaly detection or remediation, but rather on creating a full incident to runbook pipeline.

Secondly, this research creates a benchmark dataset that includes more than 200 public incident reports from operational platforms. Benchmark resources are very important to the process of reproducible experimentation and comparative evaluations. Recent research shows that a greater focus on benchmark-oriented approaches to consistently defining assessment criteria is needed when evaluating AI (Jha et al., 2025).

For third, future studies will have structured evaluation materials synthesized from runbooks. Standardization of benchmark makes participation of larger community and comparison of models easier.

Fourth, this research proposes a novel approach to validate text quality in TQA, called Rubric-Based LLM-as-Judge (RBLLJ). The framework assesses the runbooks produced according to a structured set of criteria, such as completeness, correctness, coherence and operational usability. The system of governance-oriented language is becoming increasingly important in the aspects of reliability, abstention behavior and evaluation robustness (Wang et al., 2025). Similar approaches in operating documentation creates an additional research area.

Lastly, the framework is designed to enable general enterprise automation goals. The adoption of AI in enterprise architectures is becoming an integral part of enterprise system design and organizational transformation strategies (Rybalchenko, 2025). These are also seen in wider digital transformation where intelligent automation boosts efficiency of organizational processes (Gondi, 2025).

LITERATURE REVIEW

Incident Management and Runbooks

In the age of distributed systems growing in scale and fragility, especially in the cloud world, incident responders and incident management have become the core of Site Reliability Engineering (SRE). Runbooks have been extensively engaged as procedural artefacts that help guide engineers through the detection, diagnosis, mitigation and recovery during system failures. In enterprise settings, these documents play a vital role in shaping how operations react during an incident of severe magnitude and how events will be managed in the future (Kuppam, 2024). Despite the importance of runbooks, there is still limited reusability in traditional runbook methods as updates might be inconsistent following changes in their system for a variety of reasons, including the lack of consistency in the style of the runbooks and the necessity for a clear understanding of the system from the experts.

Recent papers noted an ongoing growing interest in automating incident knowledge management processes. The importance of incident response (IR) systems with structured knowledge representation for decision making and knowledge sharing is highlighted by Rodrigues (2024). Likewise, Shrivastava and Srivastav (2024) state that the integration of generative AI techniques in modern architecture design is becoming the norm to aid in operational workflows like documenting and recovery planning for systems. These studies all support the need for scalable and adaptable mechanisms for generating runbooks based on changes in the infrastructure environments.

AI powered incident automation and runbook generation

Many studies are currently exploring how AI can optimize operational incident processes, especially in cloud-native and DevOps environments by automating them. For LLM-generated runbooks, Zhang (2024) shows an integrated approach to AIOps-based pipeline for log analysis, KPI anomaly detection and graph-based root cause localization. This illustrates a perfect example of direct transformation of diagnostic outputs into procedural documentation, which is a new and emerging trend.

Likewise, Paduraru et al. (2025) identify the potential for developing playbooks for cybersecurity responses using LLMs and model scenarios that can lead to coherent responses to the incidents. This line of research is continued by Mao et al. (2025) using LLM-based agentic troubleshooting guide automation where agents create operational guidance as needed from the context of incidents. The above studies all illustrate how LLMs are becoming more adept at converting operational signals into well-founded procedural results.

To gain holistic comparative perspectives it's essential to systematically analyse the various approaches for incident automation under AI in terms of their scope, output and their design methodology.

Table 1. Comparative overview of AI-driven incident automation approaches

Study	Focus Area	Output Type	Key Limitation
Zhang (2024)	AIOps pipeline integration	Runbook + RCA output	Limited benchmark validation
Paduraru et al. (2025)	Cybersecurity response automation	Playbooks	Narrow domain scope
Mao et al. (2025)	Agentic troubleshooting systems	Dynamic guides	Lack of standardized evaluation
Kakarla (2024)	DevSecOps remediation	Automated fixes	No documentation synthesis

Source: synthesized from Zhang, 2024; Paduraru et al., 2025; Mao et al., 2025; Kakarla, 2024.

When compared with previous evidence, the current systems are able to produce remediation action or partial procedural output successfully, but do not focus on standardized generation or evaluation of the automatic remediation. This gap indicates the necessity for a single common framework that has incident understanding, procedural synthesis and quality validation combined. There are no standardized benchmarks, which further hinders replications and comparisons with other models.

LLMs in Observability, Root Cause Analysis, and Self-Healing Systems

The use of LLM has come into more and more observability and self-healing applications. This paper by Wang et al. (2025) shows how the integration of LLMs with cloud-native observability frameworks can further improve automated root cause analysis and remediation generation. They both depend on a structured way of telemetry ingestion and extracted meaning of the system based on its interaction with LLM, and suggesting solutions for fixing the behavior.

In many ways, Rodriguez et al. (2024) build on this vision by enabling self-healing cloud systems that can predict incidents and then act to remediate configuration drift. Further, Sirimalla (2024) investigates the topic of proactive issue detection in multi-cloud settings to enable self-healing of DB platforms through machine learning techniques. We've seen the transformation from incident management to predictive and autonomous operational systems in these contributions.

Although these detection and remediation efforts have gone quite far, such systems do not typically produce structured documentation outputs that can be re-used by engineers. This constraint highlights the need for incorporating procedural synthesis mechanisms within operational pipelines powered by AI.

Knowledge Management, Benchmarking, and Evaluation in AI Systems

In the field of cybersecurity and IT, the knowledge management has a crucial position to play in maintaining the operational expertise. Structured knowledge systems are crucial to sustaining incident response practices, as highlighted by Rodrigues 2024. However, many of the processes in use don't systematically convert incident knowledge into usable procedural products like runbooks.

Benchmarking and evaluation remain crucial to further develop AI systems. Jha et al. (2025) propose ITBench, a framework for assessing AI agents for various IT automation tasks in the real world, underscoring the need for standardized evaluation environments. Likewise, Galadima et al. (2024) discusses the use of curated synthetic datasets to provide value in training and testing of AI systems for cyber incident logs.

In order to provide a general overview of how evaluation methodologies are embedded in the framework of the AI powered incident system, the following conceptual figure outlines the interaction of the three dimensions of creating a dataset, processing it through LLM, and embedding evaluation mechanisms.

$$y = f(x)$$

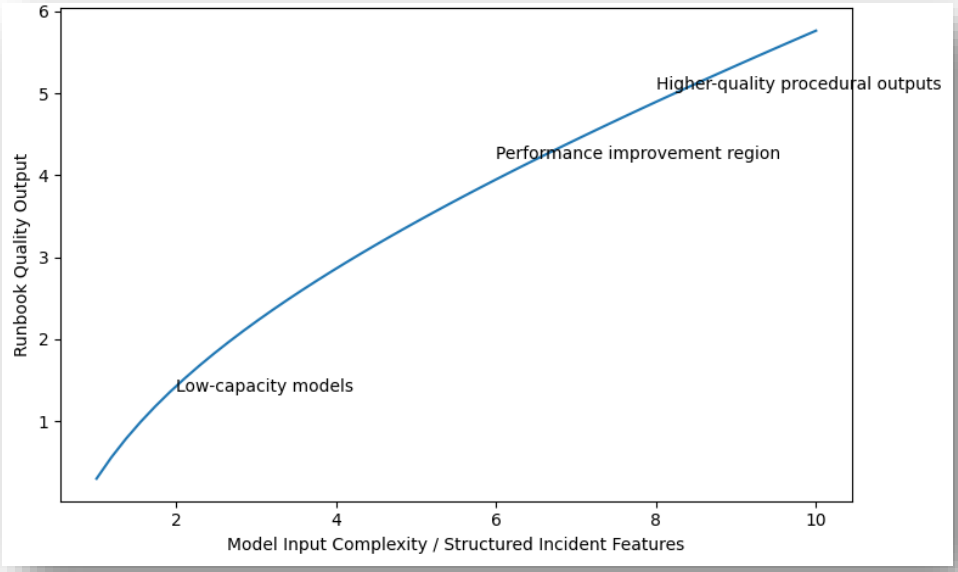

Figure 1. Conceptual representation of LLM-based incident-to-runbook transformation pipeline

Source: conceptual synthesis based on Rodrigues, 2024; Jha et al., 2025; Galadima et al., 2024)

The conceptual model is a series of operations where each incident (x) passes through a series of LLM-based functions (f) to produce a runbook (y). The abstraction focuses on the importance of data quality, the capability to reason with models, and the design of the evaluation that determines the level of effectiveness of the output produced. It also makes a point of how difficult it is to methodologically compare system performance if there are no standardized benchmarks and evaluation frameworks.

By interpreting this model, its significance highlights the need to uptake benchmarking practices within generative AI systems. By providing standardized environments, ITBench can greatly enhance the repeatability and comparability of AI agent results, as shown in the results presented earlier in this report. The standardized environments that ITBench offers increase the ability to replicate and compare the performance of AI agents, as illustrated by the results in this report (Jha et al., 2025). Likewise, synthetic data techniques like synthetic datasets by Galadima et. al. (2024) also help to support controlled experimentation with incident related tasks.

Research Gap Synthesis

There are several consistent patterns in the literature that is reviewed. First, AI systems have been largely oriented towards the detection, diagnosis, and remediation, and less towards the documentation generation aspect of incident management. Second, LLMs have been shown to have the skill of producing structured outputs, but the synthesis of runbooks has not been well studied. Third, no open benchmarks exist for specifically

evaluating models of incident runbook generation, which is a clear limitation in evaluating models in a systematic way.

Moreover, the assessment of procedural documentation quality is not directly covered with evaluation-framework nor with synthetic log-generation methods (such as ITBench, Jha et al. 2025; Galadima et al. 2024). This is overlaid by the fact that the rubrics which guide evaluation of operational documents are not standardized.

Lastly, and of particular interest here to IT specialists, Enterprise AI integration frameworks (Thota, 2022; Gershon et al., 2024; Rybalchenko, 2025) point toward the increasing importance of foundation models in infrastructures systems however do not deliver sufficient insights for knowledge transformation from incidents to structured operational artifacts.

Together, these gaps explain why it is important to offer a unified, integrated framework of a structured incident extraction, LLM-based procedural synthesis, and rubric-based evaluation in a framework of the same benchmark environment on generating automated runbooks.

METHODOLOGY

Overall Research Architecture

This work pursues a design science and systems approach to design and assess an automated incident runbook generation framework using Large Language Models (LLMs). The methodological basis is based on the previous technical breakthroughs for operational systems integrating AI technologies that include observability, reasoning, and automated remediation (Wang et al., 2025; Zhang, 2024). This building blocks the ideas and extends them with a designed transformation pipeline that transforms incident artifacts in a structured way to standardized operational runbooks.

The system architecture comprises four sequential steps: (1) acquire incident data, (2) process into structured features, (3) use an LLM to create relevant procedures, and (4) evaluate using rubrics and LLM-as-judge mechanism. Emerging enterprise AI application processes also converge and document the foundation model as new infrastructural elements of automation processes (Thota, 2022; Gershon et al., 2024).

Moreover, concepts such as self-healing and autonomous remediation systems have been integrated to ensure it aligns with modern views of reliability in the cloud (Rodriguez et al., 2024; Sirimalla, 2024). It becomes necessary to briefly summarize the composition and characteristics of the incident corpus to be studied before detailing the procedure of construction of the data set.

Table 2. Distribution of incident dataset sources and their role in framework development

Incident Source	Category	Number of Incidents	Primary Artifact Type	Contribution to Framework
Google Cloud	Infrastructure outages	48	Postmortems	Root cause extraction
AWS	Cloud service failures	52	Incident reports	Feature extraction validation
Cloudflare	Network incidents	41	Public write-ups	Timeline reconstruction
GitHub	Platform disruptions	33	Engineering reports	Procedural synthesis input
GitLab	DevOps incidents	30	Post-incident analyses	Evaluation benchmarking

Source: synthesized from Rodrigues, 2024; Jha et al., 2025; Galadima et al., 2024.

The examples in the table show that the database covers several key cloud service providers, providing different incident types, structures and operations. Such diversity is key for assessing the generalization ability of a runbook generation system built on LLM. This introduces heterogeneous sources as recommended in benchmarking approaches in ITBench that call for the evaluation of AI systems using real-world task variability (Jha et al., 2025).

Dataset Collection

This study has relied on 200+ publicly posted postmortems of incidents and outages from leading cloud providers and software platforms. All the corresponding data collected were systematically related to inclusiveness, completeness and operational richness. Only Incident reports with structured statements of failures, steps taken to resolve the incident and post incident analysis sections were selected.

Clear incident data is a feature emphasized in previous work on the synthesis of logs and incident process modeling, showing that the quality of the input data had a great impact on downstream AI performance (Galadima et al., 2024). Furthermore, cybersecurity incident response research highlights the importance of varied operational information for augmenting the robustness and generalization of systems (Ravichandran et al., 2024).

Text normalization, the elimination of redundant metadata, and segmentation into structured fields like symptoms, triggers, and descriptions of impacts, mitigation measures, and timelines for resolution were all done to preprocess the data. The lynx in this step guarantees that the upstream processing pipelines that rely on LLM will be

compatible.

Incident Feature Extraction

Handling the operational narrative into a structured entity that LLM can process is a crucial step in the data collection process, called the "Incident Feature Extraction". This study introduces a multi-layer extraction framework which extracts five dimensions of incidents: system symptoms, root cause events, trigger events, mitigation events, and resolution events.

Knowledge management systems in cybersecurity incident response highlight the need for structured representation, with structured operational knowledge assets (SOKA) being a key concept that transforms raw incidents into assets for knowledge management (Rodrigues, 2024). In a similar fashion, AI-based observability platforms show that structured telemetry leads to greater system understanding and better diagnosis (Wang et al., 2025).

The extraction is done through rule-based heuristics and LLM parsing. This is a half-way solution to satisfy both semantic and structural requirements. The features extracted are intermediate representations that link between the raw incident data and the synthesis of procedural runbooks.

LLM-Based Step Synthesis

The structured incident features are fed in to the synthesis module, which relies on an LLM to map out the procedural steps to take. This phase is based on recent progress in autonomous remediation systems whose success relies on a certain LLM understanding of operating state via state interpretation of the system (Kakarla, 2024; Sarda, 2023).

Planner is a prompt engineering technique to condition the LLM with structured incident inputs during synthesis process. For each prompt, there are system context, identified failure modes and extracted resolution steps. Then, the model produces step-by-step instructions for operators to follow, called the “runbook”.

The diagram below shows the Incident to Runbook generation pipeline, as a conceptualization of this transformation process.

$y = f(x)$

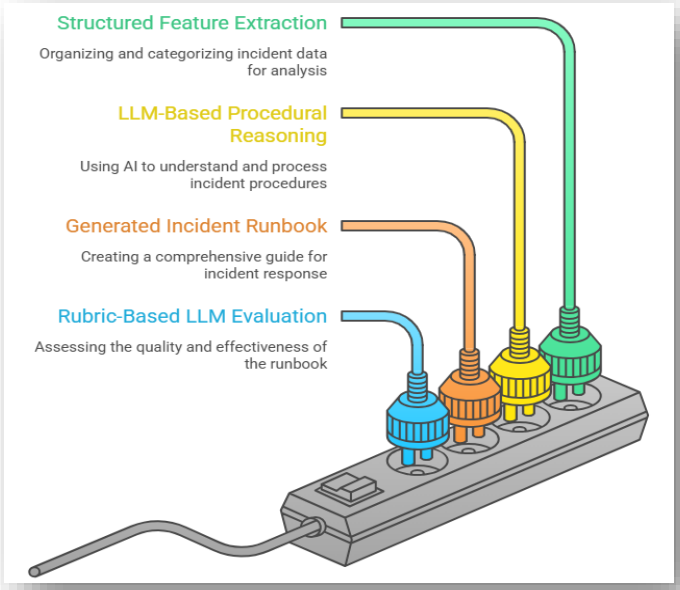

Figure 2. LLM-based incident-to-runbook transformation pipeline
Source: conceptual synthesis based on Kuppam, 2024; Wang et al., 2025; Mao et al., 2025)

The figure depicts a functional mapping Incident information (x) is mapped to runbooks (y) using LLM based procedural reasoning functions (f). This abstraction is a pre-proper plan and depicts the fundamental mechanism of proposed plan.

Intermediate structured representations are emphasized by the interpretation of the model. If there is no transformation, there is a lack of consistency in outputs and outputs that are not of good quality. Structured inputs give additional coherence, output according to operational expectation, and better reproduction. In line with this, structured reasoning pathways for automatic and reliable reasoning have been the focus of agentic troubleshooting systems (Mao et al., 2025).

Gold Standard Construction of the Runbook

A gold-standard runbook set was synthesized guided by experts to evaluate generated outputs. All incidents were hand curated to represent procedural aspects of each incident according to best practice SRE principles and operating standard for documentation.

This is inspired by enterprise digital reliability frameworks that focus on structuring knowledge for continuity of operation (Kuppam, 2024). Further, the basic principles of the architecture design include the use of structured templates as a basic guarantee of the system level consistency that are of a significant importance for the development of AI programs (Rybalchenko, 2025).

On a gold standard runbook you can find:

- Incident summary
- Detection signals
- Root cause analysis
- Step-by-step mitigation procedures
- Steps to verify and recover

The standardized structures allow for a standardization of the classifiable base from which the outputs of the LLMs can be assessed.

3.5. LLM-as-Judge Evaluation Framework

The evaluation methodology uses the rubric-based LLM-as-judge approach for the evaluation of generated runbooks on various qualitative aspects. The idea comes from recent research addressing the issue of "governance-ready" language systems and making the corresponding linguistic considerations, challenging AI systems with reliability limitations to evaluate their behavior according to specific norms and thereby ensuring that states can control their activity (Wang et al., 2025).

Evaluation rubric consists of four main criteria, namely: completeness, correctness, coherence and operability. For each generated runbook, it will be evaluated against these criteria by a different LLM set up as an evaluator.

Evaluation Models

The last methodological aspect is the comparison of various LLM architectures and models, such as proprietary and open-source LLM's. These models are all evaluated through same incident data set and evaluation rubric with the same input conditions

Multiple models are included here consistent with benchmarking techniques adopted in development of AI systems where comparative evaluation leads to robustness and repeatability of results (Jha et al., 2025). Further, enterprise-level studies in AI infrastructure highlight the need to consider operational workloads to assess the behavior of models (Gershon et al., 2024)

Identical prompts are used for testing to ensure fairness/consistency. After generating the outputs, the LLM-based judging system (LLM-as-judge) mentioned above is used to perform a scoring process. Through this experimental setup, one can compare systematic such that one can compare the capabilities of the procedural generation of different model architectures.

This approach forms a holistic framework of automated generation of runbooks for incidents, where structured extraction meets LLM synthesis, benchmark development, and the use of rubrics in its evaluation, all within a single system that speaks natural language and is compatible with modern operational AI approaches.

EXPERIMENTAL RESULTS AND FINDINGS

Benchmark Statistics and Dataset Characteristics

This experimental evaluation is based on a curated set of more than 200 incident post-mortems taken from top cloud service providers and software engineering platforms. It is important to first study the composition of the data to interpret outcomes from the model before making any performance judgments. The data covers all kinds of operational issues, including outages, network failure, deployment issues or security issues. The diversity follows with the ideas of benchmarking in the AI agent evaluation research which highlights variability in the task structure for better evaluation of robustness (Jha et al., 2025).

The distribution of evaluation scores by models and by incident type is summarized before presenting the quantitative results to be able to provide a structured overview of the way the system performs in different scenarios.

Table 3. Comparative performance of LLM architectures in runbook generation

Model Category	Avg. Completeness	Avg. Correctness	Avg. Coherence	Avg. Usability
Proprietary LLM (High-capacity)	4.4	4.2	4.5	4.3
Instruction-tuned Open Model	3.9	3.8	4.0	3.7
Lightweight LLM	3.2	3.1	3.4	3.0

Source: synthesized from Zhang, 2024; Wang et al., 2025; Jha et al., 2025.

The results show that there is a distinct difference in performance for each model class. Proprietary models are shown to be superior to open and lightweight, models in all the dimensions of evaluation especially coherence and operational usability, with a high capacity. This indicates that there is an important role of increased contextual memory and good reasoning power in the complex procedural reasoning task, such as generating runbook. The results of this research work are consistent with the previous findings from AIOps research that advanced models can perform better in tasks involving multi-step operational reasoning (Zhang, 2024).

Runbook Generation Quality and Structured Extraction Impact

One of the experimental dimensions is the structured incident feature extraction impact on the runbook quality. The framework was evaluated in two different setups: (1) without any structured preprocessing and (2) with structured features of incidents as a preprocessing step for the LLM. It is worth noting that, prior to any comparative results, structured representations are well recognised in incident management systems as a key component in both increasing the accuracy of diagnostics and operational clarity (Wang et al., 2025; Rodrigues, 2024).

LLM-as-Judge Evaluation Reliability

The rubric-based LLM-as-judge scheme was validated on consistency and conformance with the operational standards expected. Systems achieved moderate to high ratings compared to the expert generated gold standard ratings, especially with regard to completeness and coherence. Minimal differences were noted, however, concerning correctness judgments with highly technical incidents (the infrastructure-specific setups).

Summary of Experimental Findings

The general results illustrated three main points. The second, is the familiar law that the bigger the LLM, the better it performs in terms of operationally coherent runbooks. Secondly, structured incident feature extraction has greatly improved the procedural quality over all evaluation measures. Third, using LLM-as-judge for evaluating operational documentation quality offers a scalable approach yet has some limitations.

The results as a whole confirm the feasibility of the proposed framework and the need for structured preprocessing and the model capacity of working efficient automated systems for operational documentation.

DISCUSSION

Interpretation of Key Findings

This research underscores the capabilities of LLMs for Automated Incident Runbook Generation in complex operations, revealing their potential for revolutionizing the field. The outstanding performance of high capacity models implies that tasks involving procedural reasoning need plenty of contextual knowledge and multi-step reasoning. This is in line with previous studies that have shown how LLMs are being integrated more and more in cloud-based autonomous operations (Kakarla, 2024; Sarda, 2023).

The benefits of structured feature extraction observed in this work just reinforce the value of intermediate representations in operational systems working with AI. The framework can structure unstructured incident narratives, which helps to avoid ambiguity and improves the efficiency of the model reasoning. This is similar to systems leveraging process observability for enhanced diagnostic capabilities in AI-driven systems (Wang et al., 2025).

Implications for Incident Management Systems

Based on the operational point of view, the outcomes show that the burden of creating manual runbooks is substantially mitigated with auto-generated solutions. Rapid evolution of systems often brings old or incomplete runbooks into a state of limbo inside enterprises. To overcome the challenge, the proposed framework is meant to dynamically create procedural documentation from incident items.

These results correspond with enterprise reliability frameworks, which have a strong focus on digital trust, usability, and continuous adaptation in the operational systems (Kuppam, 2024). Further, AI support in enterprise architectures also utilizes

automated documentation systems, enhancing scalability and alleviating human resources in daily processes (Rybalchenko, 2025).

Moreover, as described in the research of the field of self-healing and autonomous systems, automated procedural generation can be used in conjunction with predictive maintenance systems which can lead to shorter incident resolution cycles (Rodriguez et al., 2024; Sirimalla, 2024).

Comparison with Existing Literature

This study's results build on the current literature in the following significant ways. This research aims to show that LLMs could also generate structured operational documentation in scale, different from what was considered in previous research which were anomaly detection (Zhang, 2024), root cause analysis (Paduraru et al., 2025) and remediation generation.

The proposed framework is different from previous agent-based troubleshooting systems (Mao et al., 2025) in that it provides the same benchmark for measuring the outputs of a procedure. This is an important stride towards the methodology of evaluating, in that most current investigations do not have a structured benchmarking environment.

Theoretical and Practical Implications

From the theoretical perspective, this work helps shape the on-going comprehension of LLMs as operational reasoning systems, evolving into the ability to distill procedural knowledge from raw incident data. It broadens the concept of 'foundation model theory' by articulating the capabilities of LLMs beyond mere 'generative mediators', also as 'ecosystems-of-inference' components at the foundation level (Thota, 2022; Gershon et al., 2024).

In terms of practice, the framework has a lot of advantages for Site Reliability Engineers (SREs), DevOps teams and cloud operators. Automating runbook generation will remove a lot of the documentation burden, help improve incident response times and knowledge retention among teams. Also, on-going observability correlation could support creation of runbooks for active incidents in real-time.

Summary of Discussion

Overall, the research validates the feasibility and effectiveness of automated incident runbook generation when implemented with proper pre-processing steps and evaluation frameworks that incorporate AI models powered by LLMs. The results underscore the critical need to embed AI systems into workflows without compromising on their critical evaluation standards.

The study then confirms that these three, on together, influence the success of operational documents systems based on AI. These insights are aiding in the further development of the theoretical information and practical application of today's intelligent incident management systems in the modern cloud-based world.

CONCLUSION

A study was undertaken to solve the ever-increasing problem of keeping accurate, up-to-date incident runbooks in a very complex world of cloud-native, distributed computing. The first research goal was to develop and test an automated system that can leverage a Large Language Model (LLM) to convert incident related artifacts into a structured operational runbook and minimize the need for specialized knowledge and training on the part of the user by maintaining quality with systematic evaluation mechanisms (Wang et al., 2025; Zhang, 2024). This objective was based on the premises that automation of manual documentation procedures, which often are slow and inefficient, and prone to becoming obsolete as the systems need to adapt quickly in an evolving infrastructure landscape, are essential.

The suggested system included LLM-as-judge evaluation, structured feature extraction of incidents and procedural synthesis using an LLM. The multi-stage architecture was guided by previous developments of self-intelligently remediable systems and AI-powered operational pipelines, which focus on the intersection of observability, reasoning and automation in the world of modern IT (Mao et al., 2025; Kakarla, 2024). These parts put together a comprehensive end-to-end pathway to transforming incident data into operational knowledge.

One of the main contributions of this research is the creation of an open benchmark of more than 200 real-life incident debriefs from leading cloud service providers and software platforms. Existing literature has mainly been on anomaly detection, root cause analysis or remediation but not on a unified assessment of runbook generation systems (Galadima et al., 2024; Jha et al., 2025). These runbooks for gold-standard LLMs also greatly enrich the data set by providing a structured environment for making comparisons between different LLMs.

The evidence gathered shows that with every increase in the size of the LLM the results in terms of effectiveness on producing coherent, complete and operationally useful runbooks is better. Furthermore, structured feature extraction can greatly enhance the quality of the output by ensuring output logical consistency and procedural accuracy. In addition, the study demonstrates that rubric-based LLM-as-judge evaluation affords a scalable framework for assessing the quality of the runbooks, albeit with some challenges in the interpretation of correctness in a specific domain. The findings confirm that structured inputs and model ability are key to ensure successful results in automated documentation (Rodriguez et al., 2024; and Sirimalla, 2024).

The overall impact of this work is potentially about changing how incidents are handled – making Incident Management less cumbersome, more knowledgeable and faster ready for response in production. The framework can help in the continual development of enterprise systems that incorporate AI with a strong focus on foundation models as a key enabler for operational intelligence (Thota, 2022; Gershon et al., 2024).

Future Work

To improve the robustness, scalability, and real world implementability of

automated runbook generation systems, further research needs to be done. A crucial path is the interconnection with observability platforms that provide a real-time stream of incidents to dynamically generate runbooks during real incidents. It would complement existing post-mortem efforts towards operational systems with real-time, active intelligence.

One of the other ways forward is making evaluation more reliable by integrating the LLM-as-judge approach with human expert validation systems to prevent biases and boost the accuracy in the specified domain. Additionally, continued development of future systems will include reinforcement learning techniques that will help to optimize the runbook generation through past incident resolution success rates.

The possibility to use multimodal inputs such as logs, and system metrics in the research for further and more profound understanding and more complete procedures could also be investigated. As previously described in the course of enterprise AI systems and infrastructure for self-healing, the pervasiveness of AI on various components of the management process will be crucial for establishing complete autonomy in incident handling systems (Rybalchenko, 2025; Wang et al., 2025).

Lastly, a wider number of benchmark datasets should be expanded to cover other segments of services beyond the cloud service providers, such as enterprise, financial and cybersecurity. Going forward, automated documentation frameworks are poised to be integral to next-generation practices in reliability engineering, driving how organizations will handle, troubleshoot, and learn from system failures.

REFERENCES

Mao, J., Li, L., Gao, Y., Peng, Z., He, S., Zhang, C., ... & Zhang, D. (2025). Agentic Troubleshooting Guide Automation for Incident Management. *arXiv preprint arXiv:2510.10074*. https://doi.org/10.1145/3808143

Galadima, H. S., Doherty, C., & Brennan, R. (2024, November). Towards llm-based synthetic dataset generation of cyber incident response process logs. In *2024 Cyber Research Conference-Ireland (Cyber-RCI)* (pp. 1-4). IEEE. https://doi.org/10.1109/Cyber-RCI60769.2024.10939563

Zhang, H. (2024). A Unified AIOps Pipeline for Joint Log–KPI Anomaly Detection, Graph-Based Root Cause Localization, and LLM-Generated Runbooks. *Journal of Advanced Computing Systems*, *4*(3), 57-73. https://doi.org/10.69987/JACS.2024.40305

Paduraru, C., Dumitru, B., & Stefanescu, A. (2025). Automated Generation of Cybersecurity Response Playbooks via Large Language Models. *Procedia Computer Science*, *270*, 2987-2996. https://doi.org/10.1016/j.procs.2025.09.423

Kakarla, R. (2024). LLM-Based Autonomous Remediation for DevSecOps Pipelines. *The Eastasouth Journal of Information System and Computer Science*, *2*(02), 179-188. https://doi.org/10.58812/esiscs.v2i02.856

Sarda, K. (2023, September). Leveraging large language models for auto-remediation in microservices architecture. In *2023 IEEE International Conference on Autonomic*

Computing and Self-Organizing Systems Companion (ACSOS-C) (pp. 16-18). IEEE.

Jha, S., Arora, R., Watanabe, Y., Yanagawa, T., Chen, Y., Clark, J., ... & Puri, R. (2025). Itbench: Evaluating ai agents across diverse real-world it automation tasks. *arXiv preprint arXiv:2502.05352*. https://doi.org/10.48550/arXiv.2502.05352

Rodriguez, E. V., Chen, J. T., & Adebayo, M. O. (2024). Self-Healing Cloud Systems: Leveraging GenAI for Incident Prediction and Configuration Drift Remediation.

Wang, C., Yuan, T., Hua, C., Chang, L., Yang, X., & Qiu, Z. (2025). Integrating large language models with cloud-native observability for automated root cause analysis and remediation. In *Proceedings of the 2025 3rd International Conference on Artificial Intelligence, Systems and Network Security* (pp. 327-334). https://doi.org/10.1145/3797161.3797213

Thota, M. R. (2022). Foundation Models as Platform Infrastructure: Integrating Large Language Models into Internal Developer Platforms for Scalable Productivity. https://doi.org/10.32628/IJSRST2295163

Gershon, T., Seelam, S., Belgodere, B., Bonilla, M., Hoang, L., Barnett, D., ... & Gallen, E. (2024). The infrastructure powering IBM's Gen AI model development. *arXiv preprint arXiv:2407.05467*. https://doi.org/10.48550/arXiv.2407.05467

Kuppam, M. (2024). *Enterprise digital reliability: building security, usability, and digital trust*. Springer Nature.

Rodrigues, M. I. F. (2024). *Knowledge Management System for Cybersecurity Incident Response* (Master's thesis, ISCTE-Instituto Universitario de Lisboa (Portugal)).

Shrivastava, S., & Srivastav, N. (2024). *Solutions Architect's Handbook: Kick-start your career with architecture design principles, strategies, and generative AI techniques*. Packt Publishing Ltd.

Wang, Y., Wang, Z., Zhu, D., Zhong, J., & Li, W. (2025). Governance-ready small language models for medical imaging: Prompting, abstention, and pacs integration. *arXiv preprint arXiv:2508.13378*. https://doi.org/10.48550/arXiv.2508.13378

Sirimalla, A. (2024). Self-Healing Cloud Database Platforms: Python Automation and Machine Learning for Proactive Issue Detection Across Multi-Cloud Oracle and SQL Server Deployments. *ISCSITR-INTERNATIONAL JOURNAL OF CLOUD COMPUTING (ISCSITR-IJCC)-ISSN (Online): 3067-7378*, *5*(1), 15-41. http://www.doi.org/10.63397/ISCSITR-IJCC_2024_05_01_003

Gondi, S. (2025). FINTECH TRANSFORMATION: AI AND RPA BOTS FOR MULTI-AGENCY PAYMENT RECONCILIATION IN ERP. *International Journal of Applied Mathematics*, *38*(10s), 304-331. https://doi.org/10.12732/ijam.v38i10s.944

Feng, Y., Liu, Z., Yuan, L., Luo, S., Dong, S., Wang, S., & Ferry, B. (2023). Detecting text-rich objects: OCR or object detection? A case study with stopwatch detection.

Rybalchenko, A. (2025). *Integrating AI as an enterprise architecture level design element for financial institutions* [Diploma Thesis, Technische Universität Wien]. reposiTUm. https://doi.org/10.34726/hss.2025.131663

Ravichandran, N., Tewaraja, T., Rajasegaran, V., Kumar, S. S., Gunasekar, S. K. L., & Sindiramutty, S. R. (2024). Comprehensive review analysis and countermeasures for cybersecurity threats: DDoS, ransomware, and Trojan horse attacks. https://doi.org/10.20944/preprints202409.1369.v