2026 Volume: 02 Issue: 06 www.ijrpa.com ISSN 2456-9995 Research Article

International Journal Research Publication Analysis

Page: 01-28

A SURVEY OF LLM AGENTS FOR PLATFORM OPERATIONS, INCIDENT RESPONSE, AND DIAGNOSTICS: A COMPREHENSIVE REVIEW AND RESEARCH AGENDA

***Akhil Reddy Mandadi**

Independent, India.

Article Received: 17 May 2026
Article Revised: 05 June 2026
Published on: 25 June 2026

***Corresponding Author: Akhil Reddy Mandadi**
Independent, India.

DOI: https://doi-doi.org/101555/ijrpa.6700

ABSTRACT

Platform operations, incident response, diagnostics, and operational automation are all revolutionized by Large Language Model (LLM) agents. Operational teams managing system reliability and performance are faced with enormous challenges due to the complexity of the cloud, distributed, microservices and hybrid computing infrastructures. The advent of recent developments in LLM results in the creation of intelligent agents that have the ability to comprehend the operational context, analyze logs, correlate events, retrieve knowledge, interact with tools, and support remediation activities.

Although research work is in progress on this subject, the current studies are still in isolation in different areas such as AIOps, DevOps, cyber security operations, infrastructure management, and autonomous system administration. Moreover, there are significant differences between prototypes in academia and real deployments in terms of reliability, safety, governance and evaluation approaches. The survey offers a thorough overview of over a hundred studies covering the past few years on LLM agents in platform operations. A new four-axis taxonomy is presented for literature classification based on reasoning architecture, grounding strategy, and action model and evaluation methodology.

The literature is classified on the basis of a novel four-axis taxonomy: reasoning architecture, grounding strategy, action model, and evaluation methodology. The survey also covers the key areas of application, such as incident response, log analysis, root cause diagnosis, infrastructure troubleshooting and automated remediation. Comparative analysis shows the common architectural characteristics, operational restrictions and design trends. Special emphasis is given to the transition from research systems to deployment in production highlighting issues related to trustworthiness, scalability, and governance and production

safety.

The results of these studies inform a broad research landscape for future studies to foster robust, explainable, secure, and production-ready operational LLM agents.

KEYWORDS: Large Language Models; LLM Agents; Platform Operations; Incident Response; Diagnostics; AIOps; DevOps; Root Cause Analysis; Automated Remediation; Autonomous Operations; Reliability Engineering.

I. INTRODUCTION

A. Background and Motivation

Cloud computing, container orchestration, microservices, edge computing and distributed infrastructure have added a significant amount of complexity to the operation of today's enterprise applications. More and more, organizations rely on highly interconnected digital systems that produce vast amounts of telemetry, logs, traces, metrics, alerts and operational events. However, such environments are complex, necessitating a significant amount of expertise, ongoing monitoring, and fast response times to ensure system reliability and service availability. Time-consuming manual investigations and rule-based automation are inadequate to match the pace of modern infrastructures.

At the same time, big language models exhibit great ability to reason, understand language, plan, synthesize knowledge, and use tools. In the last few years, LLMs have been shown to have a significant influence in scientific, engineering, and operational fields, and are expected to improve human decision-making and automate high cognitive workload tasks [1], [13]. Some new architectures, known as agentic architectures, have recently added more capabilities, allowing LLMs to interact dynamically with external systems, fetch contextual information, carry out operational tasks, and continuously adapt their reasoning processes.

Many operational workflows involve interpreting natural language, accessing knowledge, following troubleshooting steps, exploring incidents and making decisions collaboratively, making operational environments a particularly promising context for LLM adoption. The conventional automation systems are designed to perform automated tasks based on a predetermined workflows; however, they do not have the contextual sense in unexpected scenarios. LLM agents, on the other hand, can analyze a wide range of operational information sources and provide adaptive responses, which are tailored to the context of dynamic incidents

B. LLM Agents in Platform Operations

In the last few years, the use of LLM agents in platform operations has grown significantly. The adoption of language models in operating systems, infrastructure management systems, developer tools and platforms, and operational automation pipelines has been investigated [6] and [7]. More and more of these systems are used to help with a variety of operational activities such as alert triage, incident investigation, log summarization, anomaly explanation, root-cause analysis, configuration validation, and remediation planning.

In incident response settings, LLM agents can help operators by summarizing alerts, consolidating information from various monitoring systems, and providing recommended responses. In the realm of cybersecurity operations, studies have also shown the potential of conversational agents to simplify cyber investigations and shorten response times in the case of complex security incidents [5]. In the field of database management, dedicated systems are starting to use LLMs to analyze operational telemetry and provide intelligent diagnostics for performance issues and infrastructure failures [11].

LLM capabilities aligned with AIOps platforms is a significant development in the field of operational engineering. This modern approach to AIOps research doesn't just focus on techniques for managing failures, but also on enhancing observability and automated decision-support systems using language-based reasoning [14]. The developments show a general move from fixed automation to operational intelligence that is adaptive and can support increasingly complex infrastructures.

C. Research Challenges

While there have been developments to great effect, there are still many issues to overcome in deploying operational LLM agents. The reliability issue is a big concern since the hallucinated outputs may lead to wrong diagnosis or unsafe recommendations for remediation. Current generative systems are not capable of providing the level of precision, accountability and repeatability that is often required in operational environments.

There are several other barriers, chief among them trustworthiness and alignment. The outputs of the LLM must be in line with the organization's policies, compliance and operational goals. Trustworthy LLM deployment requires careful and extensive analyses, highlighting the significance of robustness, transparency, alignment assessment, and risk management strategies in critical decision-making scenarios [8]. The same issues come to mind when agents are directly manipulating production infrastructure where they might be causing problems with service or security.

Another challenge is to integrate external tools, monitoring systems, knowledge repositories and infrastructure management platforms. Operational agents need to possess reasoning and structured data access/retrieval and execution capabilities. Therefore, prompt engineering, agent orchestration, and tool coordination become essential elements of the operational system design that must be taken into consideration [4]. In addition, the methods of assessment also vary widely in the literature, making it difficult to make objective comparisons between methods.

D. Survey Scope and Contributions

This survey will address these challenges by offering a comprehensive review of LLM agents, intended for platform operations, incident response, diagnostics and remediation. The survey draws together knowledge on the latest trends in LLM architectures, operational intelligence, DevOps automation, AIOps, cybersecurity operations, and autonomous infrastructure management.

The main contribution is a four axis taxonomy of existing literature, based on reasoning architecture, grounding strategy, action model and evaluation methodology. This approach allows for the comparative analysis of a variety of approaches systematically and helps identify recurring architectural patterns and operating principles. Also, the survey covers practical deployment issues and points out the significant research to production gap.

II. SURVEY METHODOLOGY

A. Literature Collection Strategy

The goal of the literature collection strategy was to build a representative and complete literature base of LLM agents for platform operations, incident response, diagnostics, infrastructure troubleshooting and automated remediation. Considering the interdisciplinary approach of the field of AI, the review included research from artificial intelligence, DevOps, AIOps, cybersecurity operations, cloud computing, infrastructure management, autonomous systems and operational intelligence. The literature gathering process focused on both the fundamental progress made in the field of large language models and the novel ways in which they are being applied. Theoretical foundations were developed by conducting in-depth surveys on LLM architectures, prompting strategies, trustworthiness and applications, which served to better understand the capabilities and limitations of agents [1, 2, 4, 8, and 13]. At the same time, domain-specific research on topics such as DevOps automation, database diagnostics, operating-system intelligence, edge intelligence, cybersecurity

.

operations and failure management provided hands-on knowledge about operation deployments [3, 5, 6, 11, 14, and 15].

For systematic coverage, publications from major academic databases, such as IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, arXiv, and Google Scholar were identified. The search strategy involved the use of various combinations of keywords to various extents, including "LLM agents," "AIOps," "incident response," "root cause analysis," "operational diagnostics," "autonomous remediation," "DevOps automation" and "platform operations. Backward and forward citation analysis was used to identify additional studies. Operational applications of LLMs have come a long way since 2022, so a particular focus was given to recent publications. This led to a collection of peer-reviewed journal articles, conference proceedings, industrial research reports and key preprints on the new operational agent architectures [7], [9], [10], and [12].

Table I summarizes the major categories used throughout the literature collection and classification to explain the selection process.

TABLE I: LITERATURE COLLECTION CATEGORIES AND SELECTION FOCUS.

Category	Research Focus	Representative References
LLM Foundations	Architectures, capabilities, limitations	[1], [13]
Prompting and Reasoning	Prompt engineering, agent reasoning	[4], [10]
Trustworthiness and Security	Reliability, safety, alignment	[8], [2]
Platform and DevOps Operations	Infrastructure automation and management	[3], [7], [15]
Diagnostics and Failure Management	RCA, troubleshooting, AIOps	[11], [14]
Cybersecurity Operations	Security investigations and response	[5], [9]
Emerging Operational Domains	Edge systems, UAVs, intelligent OS	[6], [12]

Source: Compiled by the authors based on references [1]–[15].

Table I shows literature from both fundamental LLM research and operational deployment fields is selected. This is important because operational agents need language reasoning skills as well as infrastructure specific knowledge, which means that they're dependent on several research communities. The overall workflow of the literature collection is presented in **Figure 1**.

.

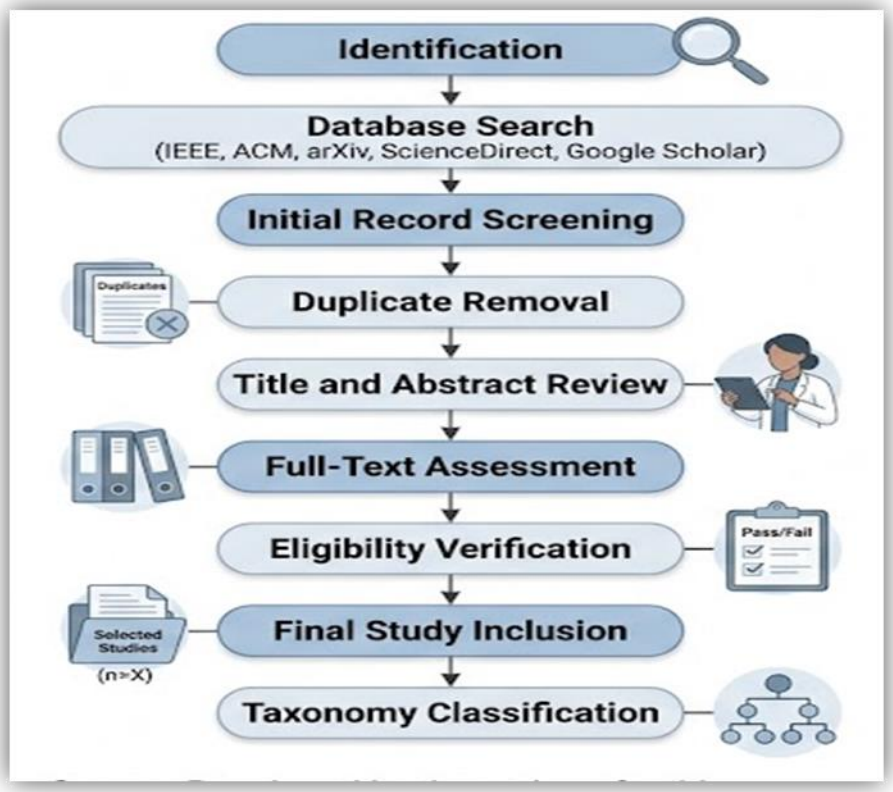

Figure 1. A workflow for literature collection and screening.

Source: Developed by the authors for this survey.

Figure 1 shows the sequential filtering process used to ensure the methodological rigor in the review process. Initial broad searches of databases were followed by increasingly relevant, methodologically sound and platform-appropriate studies being screened. In the initial screening phases, duplicate publications and those not LLM application related were removed. Each study was then reviewed using full-text assessments that assessed its contribution to the following categories: reasoning architectures, grounding strategies, action models, and evaluation methodologies.

III. Understand the basics of LLM agents for platform operations

A. Evolution from Traditional Automation to LLM Agents

Platform operations have had a history of deterministic automation mechanisms, which are based on rules, scripts and workflow engines. Operational systems were good at completing repetitive tasks, but were less successful when faced with new failures, uncertain alerts or multiple system dependencies. As infrastructure grew in size and complexity, organizations began to leverage AI for their operations to analyze telemetry data, correlate events and make decisions. An intelligent lifecycle management system has been shown to increase

operational efficiency and scalability in early developments of autonomous infrastructure management [3]. Later, the adoption of foundation models on developer platforms opened the door to go beyond procedural execution to contextual reasoning and knowledge synthesis [7]. The advent of large language models greatly accelerated this shift. Recent studies have shown that LLM models have capabilities such as reasoning, planning, summarization, and tool interaction, which can be used to enhance complex operational workflows [1] and [13]. Moreover, the literature in the field of investigating the integration of AI in operating system reveals a general trend of moving towards intelligent management systems that can comprehend the operational context and respond to the changing environments [6]. All these developments are steps towards the move from deterministic automation to agentic operational intelligence.

This evolution can be summarized as follows Table II highlights the key characteristics of major operational paradigms.

TABLE II: EVOLUTION OF OPERATIONAL AUTOMATION PARADIGMS.

Paradigm	Primary Capability	Limitations
Rule-Based Automation	Script execution and workflow enforcement	Limited adaptability
AIOps Systems	Pattern detection and event correlation	Restricted reasoning capability
LLM-Augmented Operations	Contextual analysis and recommendations	Reliability concerns
Autonomous LLM Agents	Planning, reasoning, and action execution	Governance and safety challenges

Source: Developed by the authors based on [1], [3], [6], [7], [13], [14], and [15].

As shown in Table II, from execution-centric systems to reasoning-centric architectures, operational intelligence is progressing. Autonomous agents provide more adaptability, but also pose validation, trust and control issues.

B. Core Components of Operational Agents

Modern operational agents integrate several architectural elements that all contribute to diagnosis, reasoning and remediation activities. The alert and log interpreter, as well as the incident and operational documentation interpreter, are cognitive functions powered by LLM reasoning engines. The LLM reasoning engines provide the cognitive power for interpreting alerts, logs, incident reports, and operational documentation. Prompt engineering frameworks are used in order to structure reasoning ways and to improve task-specific behavior with the

.

aim to augment the performance of the agent [4].

Other systems, such as retrieval systems, memory mechanisms, and external tool integrations, are also necessary for operational agents. In resource-constrained environments, Edge intelligence research reveals that combining retrieval and contextual grounding greatly enhances decision quality and accuracy [2]. The same principles can be seen in language models for structured operational information in database diagnosis systems [11]. Agents with a cybersecurity focus augment these capabilities by engaging in interactive reasoning and evidence correlation, which support investigative activities [5]. Figure 2 is a conceptual architecture for an operational LLM agent.

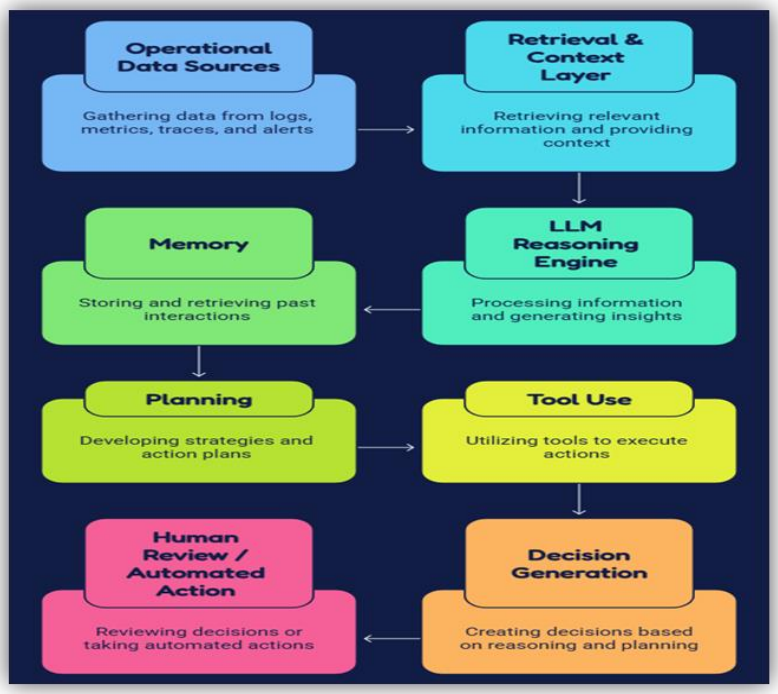

Figure 2. Core Architecture of Operational LLM Agents.

Source: Developed by the authors based on [2], [4], [5], [8], [11], and [14].

Figure 2 emphasizes the multi-layered structure of the operational agents. When developing language generation systems, it is not enough to rely on language generation alone; it is also necessary to include retrieval, memory, planning, and tool execution functions in the system for reliable operational decision-making.

C. Operational Workflow Lifecycle

Operational agents operate in the context of a larger incident-management lifecycle that includes detection, investigation, diagnosis, remediation and post-incident learning. With the

emergence of AI technology, modern research on AIOps is increasingly working with the ability of LLM in failure management, including assisting in alert triage, anomaly interpretation and root cause investigations, as illustrated in [14]. Transformer-based DevOps automation platforms which use intelligent orchestration to manage the operational workflow and remediation actions can be seen as similar. [15]

Multi-agent collaboration has also recently been shown to play a key part in complex decision settings. Distributed reasoning mechanisms are employed to make it possible for specialized agents to coordinate planning, information-gathering and execution activities across domains of operation [10]. Digital twins are also used in complementary research to model operational environments to assist with testing and validation as well as continuous learning prior to production deployment [9]. Applications to highly dynamic systems such as UAV systems further demonstrate the versatility of operational intelligence with LLM technology for emerging applications [12]. Taken together, they set the ground rules for modern operational agents and the conceptual framework for the taxonomy to follow in subsequent chapters.

IV. TAXONOMY OF LLM AGENTS FOR PLATFORM OPERATIONS

A. Axis I: Reasoning Architecture

The cognitive basis of operational LLM agents is called reasoning architecture. It frames the meanings of operational signals, generates hypotheses, investigates failures and makes remediation recommendations. Four basic paradigms have emerged in the literature regarding reasoning. In the literature, four dominant paradigms of reasoning have emerged.

Zero-Shot Reasoning

a) Characteristics

In zero-shot reasoning, the reasoning is performed directly using the prompt without any intermediate steps. Operational context (alerts, logs, or incident description) is provided to the model, and outputs diagnostics immediately. This was a very simple and also low computational design method that was often utilized in early operational applications, such as [1], [4] and [13].

b) Advantages

The main benefit of zero-shot reasoning is the efficiency. The level of prompt engineering and inference complexity is low to facilitate quick roll-out in operational settings. These

systems can be of great value for alert summarization, alert interpretation and initial alert triage when immediate responses are needed, e.g., [6] and [7].

c) Limitations

In most cases, however, multi-system failures, which require structured reasoning, are not dealt with very well by zero-shot approaches. There are often hidden dependencies and cascading failures that cause operational incidents that are beyond direct inference. Reliability and explainability are thus still important questions [8, 14].

Chain-of-Thought Reasoning

a) Multi-step Diagnosis

Chain of Thought (CoT) reasoning adds intermediate reasoning steps that are explicitly written to increase the analysis depth. Instead of drawing direct conclusions, agents check evidence, make hypotheses and make successive diagnoses. These methods have been shown to be successful in complex trouble-shooting surroundings [4] and [11].

b) Explainability Benefits

Evaluation of recommendations is frequently a need for operational teams to be transparent. CoT reasoning enhances interpretability as reasoning traces offer insights into how the conclusion is reached. This feature is in line with the increasing need for reliable and traceable AI systems [8, 13].

c) Operational Challenges

CoT reasoning is not only more expensive in terms of computation, but also more expensive in terms of latency, even when the diagnostic quality is improved. Although extended reasoning chains can also introduce errors, particularly when incorrect assumptions arise at early stages in the reasoning process, the reliability of such chains can be affected under high pressure conditions in the field of incident response.

ReActBased Agents

a) Reasoning-Action Cycles

ReAct architectures are architectures that integrate reasoning with external actions. Agents switch between analytical thinking and invoking tools, facilitating dynamic evidence collection in investigations. This paradigm is similar to human troubleshooting process [5], [14].

b) Tool Interaction Mechanisms

Operational ReAct agents routinely communicate with knowledge repositories and

.

infrastructure APIs, monitoring dashboards, and observability platforms. Tools and reasoning are used to continuously update the situational awareness and refine the diagnosis of agents [2], [11]

c) Incident Investigation Workflows

ReAct architectures are especially well suited for incident response environments where information is constantly changing, as they are iterative. Investigative workflows are dynamic and not one-size-fits-all, and they allow agents to follow more than one diagnostic path before deciding on what to do.

Plan-and-Execute Agents

a) Hierarchical Planning

Plan and execute architectures break up the plan/execute process. High level planners break down complex goals into well-defined sub-goals, thus allowing systematic investigation and remediation strategies to be developed [10], [15].

b) Multi-Stage Remediation

Often, multiple systems must be coordinated to remediate operational failures. Plan-and-execute agents provide assistance for staged interventions and procedures that include diagnosis, validation, implementation, verification, and rollback.

c) Long-Horizon Operational Tasks

These architectures are particularly useful for long-term operations like capacity optimization, infrastructure migrations and disaster recovery planning. They have well organised planning systems, which means that they are more consistent and less uncertain to execute.

A comparative analysis is summarized to show the differences in reasoning architectures in **Table III.**

TABLE III: COMPARISON OF REASONING ARCHITECTURES FOR OPERATIONAL AGENTS

Architecture	Strength	Weakness	Typical Use Case
Zero-Shot	Fast inference	Limited depth	Alert triage
Chain-of-Thought	Explainability	Higher latency	Root-cause analysis
ReAct	Dynamic investigation	Tool dependency	Incident response
Plan-and-Execute	Structured planning	Greater complexity	Multi-stage remediation

Source: Developed by the authors based on [1], [4], [5], [10], [11], [14], and [15].

As illustrated in Table III, there is no one dominating reason architecture for all operational

.

scenarios. Rather, choosing an architecture depends on the goals you want to achieve, how much risk you are willing to take and how autonomous your operation needs to be.

B. Axis II: Grounding Strategy

Grounding strategies refer to the way the operational agent's link the outputs produced to credible information sources. In an operational context, accuracy, traceability and contextual relevance are demanded and depend on effective grounding.

Retrieval-Augmented Grounding

This will be achieved through the integration of information from operational repositories, documentation systems, and historical incident databases, which is known as Retrieval-Augmented Generation (RAG) and helps improve the performance of the agent. Retrieval techniques have been shown in numerous studies to enhance the accuracy of the context of the information retrieved and lower the risk of hallucinating information [2], [8].

Runbook-Structured Grounding

Runbook-structured grounding biases thinking within the framework of the operational steps specified in the runbook. Agents use organizational runbooks to ensure that they are aligned with the established operational practices. This way, the consistency and governance of the enterprise are enhanced [3] and [15].

Knowledge Base Grounding

Knowledge base grounding relies on the use of infrastructure configuration, dependency mapping, incident history, troubleshooting and more in a structured repository. These repositories can be used for agents to reason with organizational knowledge and not only with the parameters of the model [7], [11].

Tool Output Constrained Grounding

Here, agent outputs are limited based on the responses of external tools. Platforms, systems and tools for monitoring, configuration and observability deliver authoritative information that is used to limit reasoning paths. This greatly reduces the operational reliability [5], [14].

Hybrid Grounding Architectures

Various recent research efforts support hybrid grounding architectures that integrate retrieval systems, knowledge repositories, runbooks and tool outputs in one and the same grounding. Hybrid approaches are found to be more rich in context and robust across complex operational environment [6], [9]. The relationship among grounding mechanisms is illustrated in **Figure 3.**

.

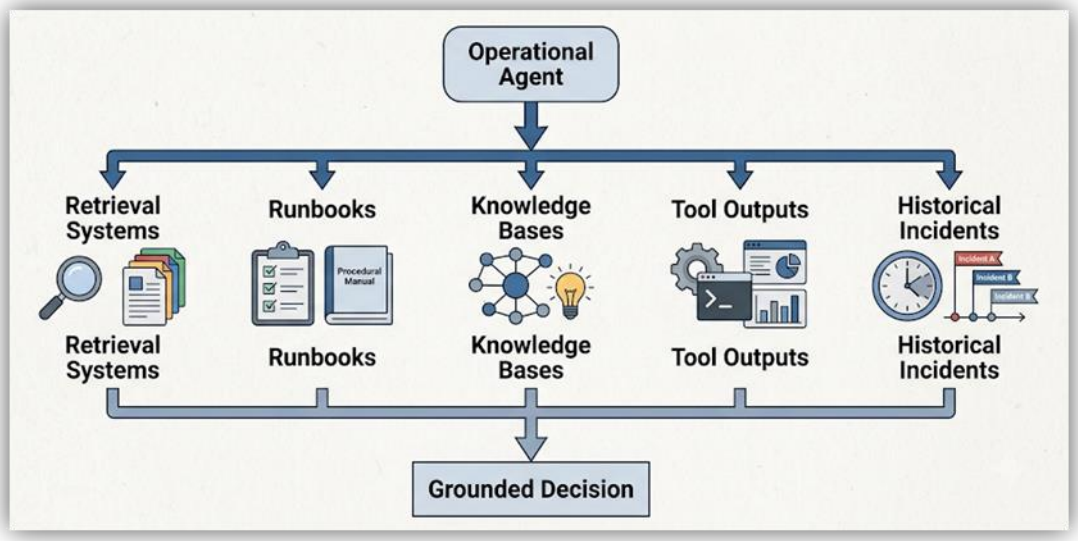

Figure 3. Grounding Strategies for Operational LLM Agents.

Source: Developed by the authors based on [2], [3], [5], [7], [8], [9], [11], and [14].

Figure 3 presents how multiple information sources can help enhance the quality of the decision as a whole. The hybrid grounding is a trend that is based on the recognition of the need for operational reliability in the presence of contextual diversity and not relying on a single information source.

C. Axis III: Action Model

Action models specify the extent of the authority of an agent

Read-Only Diagnostic Agents

These agents are analysis only and offer some insight into operational data without taking action. These are the most commonly used ones and they are able to reduce the operational risk while at the same time increasing the efficiency of the diagnosis [11], [14].

Advisory Recommendation Agent

Advisory systems provide information about remediation suggestions. The responsibility of the final decision is still in the hands of human operators, thus the implementation of governance and accountability [1], [5].

Human-in-the-Loop Execution Agents

Human-in-the-loop agents are only allowed to carry out approved actions when the operator confirms them. In the enterprise context, this model provides a good compromise between automation efficiency and operational safety and is gaining popularity. [8], [15].

.

Write-Enabled Remediation Agents

Write-enabled agents have direct permissions to change configurations, restart services and run operational workflows. These systems cause fast response times and pose major governance issues [3, 6].

Fully Autonomous Operational Agents

Fully autonomous agents self-explore, self-diagnose and self-repair failures. The concept is interesting, but the rollout is inhibited by trust, compliance and safety issues [10], [14].

D. Axis IV: Evaluation Methodology

One of the most diverse areas of ongoing research is the evaluation methodology

Synthetic Benchmark Evaluation

Assessment of reasoning abilities are generally based on synthetic datasets and benchmark environments. These benchmarks are very useful for controlled experimentation, but lack the complexity of operation [4], [13].

Historical Incident Replay

Historical incident replay is a form of evaluating agents by past failures that have been captured. This methodology is more realistic and yet controlled experimentally [11, 14]. To compare evaluation approaches, Table IV summarizes their characteristics.

TABLE IV: COMPARISON OF OPERATIONAL EVALUATION METHODOLOGIES.

Methodology	Realism	Operational Risk	Scalability
Synthetic Benchmarks	Low	None	High
Historical Replay	Medium	None	Medium
Shadow Mode	High	Low	Medium
Human Assessment	High	None	Low
Production A/B Testing	Very High	Moderate	Low

Source: Developed by the authors based on [5], [8], [10], [11], [14], and [15].

It is obvious that there is a trade-off between realism and deployment risk as shown in Table IV. More realistic evaluations are better for showing effectiveness of operations, but place more demanding governance requirements.

Shadow-Mode Deployment

Shadow-mode deployment is a deployment that tests agents with human operators but does not enable agents to run independently. This way, agent behavior can be safely studied in real operation environment, as in [14, 15].

.

Human Expert Assessment

Recommendation quality, the validity of reasoning, and the usefulness of the operation are assessed by human expert evaluations. In this regard, evaluations of explainability and trustworthiness are still relevant to the assessment of such evaluations as in [8] and [13].

Production A/B Testing

Production A/B testing is the most stringent evaluation method as agents are tested in real production environments. This is the most expensive and complicated solution, but offers most convincing evidence on operational value, scalability, and real-world performance [3] [9]. Together, these four axes form a comprehensive taxonomy of the rapidly growing body of literature on operational LLM agents, and offer a structured framework for future comparative analysis and the development of new research.

V. APPLICATION DOMAINS AND USE CASES

A. Incident Response Automation

Operational LLM agents are one of the most developed application spaces, with incident response being one of them. Today's infrastructures produce a tremendous amount of alerts, which often become too much for the operational team to handle. LLM agents help with alert triage, prioritizing based on the severity of the alert, business impact, and context. These agents lower the cognitive load and increase the efficiency of the responses with natural language understanding and contextual reasoning [5], [14].

B. Log Analysis and Monitoring

Another interesting application area is log analysis. Modern infrastructures generate a wealth of log streams that are too big and outpace manual analysis techniques. LLM agents can help with summarizing logs automatically, converting them from unstructured to structured data and providing brief explanations that facilitate quick situational awareness. Database diagnostics has been studied, and shows how language models can be used to understand operational telemetry and detect meaningful diagnostic patterns from the complex output of a system [11]. Table V summarizes the variety of operational applications.

TABLE V: MAJOR APPLICATION DOMAINS OF OPERATIONAL LLM AGENTS.

Application Domain	Primary Functions	Operational Benefits
Incident Response	Triage, escalation, coordination	Faster response times
Log Analysis	Summarization, interpretation	Improved observability
Root-Cause Diagnosis	Investigation and reasoning	Reduced diagnostic effort

Infrastructure Troubleshooting	Configuration validation	Increased reliability
Automated Remediation	Recovery execution	Reduced downtime

Source: Developed by the authors based on [3], [5], [10], [11], [14], and [15].

As shown in Table V, the operational agents are increasingly tasked to cover the whole incident life cycle, rather than to deal with isolated operational tasks. The movement towards integrated operational intelligence platforms is continuing in this trend.

C. Root-Cause Diagnosis

Root cause diagnosis is still one of the most valuable operational applications; tracing down the origin of failures often takes a lot of engineering effort. LLM agents can help by means of dependency tracking, failure localization, and causal reasoning. Dependency tracing is the process of looking at the relationships between services, infrastructure elements, databases and network resources to determine if there are any potential failure propagation paths. New operational intelligence systems are helping to enhance this process by using contextual reasoning [6], [11].

Failure localization is an extension of diagnosis, providing more scope for investigation and a pointer to likely sources of failure. Instead of having to check all of the components of a system, some of the attributes used by agents to decide where to look for failure are the "contextual" attributes and "historical" knowledge. Causal reasoning improves the success of diagnosis by determining the relationships between observed symptoms and system behaviors. The study on intelligent orchestration and autonomous infrastructure management shows that these features have a great impact in the efficiency of operational decision making processes [3], [15].

D. Configuration and Infrastructure Troubleshooting

In today's cloud-native world, configuration-related issues continue to be one of the top reasons for service disruptions. LLM agent’s help the operational team look for misconfigurations, validate policy compliance, and troubleshoot infrastructure anomalies. Agents can identify inconsistencies between their organization and documented best practices or organizational standards before they create failures in service delivery by using comparison [7].

The other important ability is the policy validation. Security, Governance and Operational policies are often in place within Infrastructure environments. Organizational knowledge bases can be used to evaluate the configuration against the requirements and to provide

remediation suggestions to agents. Other research that has explored the integration of AI into operating systems indicates that the intelligent diagnosis of infrastructure problems could become even more automated in future operation contexts [6]. Therefore, LLM-based troubleshooting is a scalable solution to mitigate operational risk and enhance reliability of infrastructure.

E. Automated Remediation

The highest level of usage of operational agents is automated remediation. Here, agents create recovery plans, schedule recovery process workflows and confirm the remediation results. Recovery recommendation systems help the operator by suggesting corrective actions based on past incidents, documented procedures and contextual evidence [15].

More sophisticated systems can be automated by way of direct integration with infrastructure management tools. These capabilities can facilitate a fast response to common failures and minimize the need for manual efforts. Validation and rollback mechanisms are equally important to ensure intended actions are carried out, as their actions must be validated. Digital twin environments and simulation-based approaches are becoming more and more used for safe validation before entering production [9]. All these developments suggest a steady step towards ever more independent operational ecosystems, with adequately safeguarding governance measures present.

VI. The existing approaches are analyzed comparatively

A. Comparison across Reasoning Architectures

The choice of the reasoning architecture for an operational agent significantly affects the performance, scalability, and reliability of the system. While zero-shot approaches are simple to implement and can have low latency, they may not be sophisticated enough for complex investigations [1], [4]. However, chain-of-thought architectures explicitly go through reasoning steps to enhance their diagnostic accuracy, which ends up consuming more computations. Adaptive investigation can be achieved by adapting the reasoning and tool interactions in ReAct agents, whereas plan-and-execute systems can be used for structured management of long-horizon operational tasks [10, 14]. Based on a comparative analysis, it can be found that the more complex the reasoning architecture, the higher the level of problem-solving capacity, but the more complex the operation and governance.

B. comparison across grounding strategies

The operational trustworthiness is strongly influenced by the grounding strategies used. To achieve a higher level of contextual relevance, retrieval augmented methods include external

information sources which limit hallucination risks [2]. Runbook-grounded grounding increases the procedure consistency and knowledge-based grounded increases the organization alignment by giving access to historical and operational knowledge repositories [7]. In tool-output-constrained grounding, reasoning by agents is constrained by authoritative outputs, which offers further protections. Hybrid approaches are always shown to be more robust (as they combine more than one information source at once) [8], [14].

For ease of comparison between various dimensions of taxonomy, some salient features observed in the literature are summarized in Table VI.

TABLE VI: COMPARATIVE ANALYSIS OF TAXONOMY DIMENSIONS.

Taxonomy Dimension	Strengths	Challenges
Reasoning Architecture	Diagnostic flexibility	Computational overhead
Grounding Strategy	Improved reliability	Integration complexity
Action Model	Operational efficiency	Governance concerns
Evaluation Methodology	Performance assessment	Lack of standardization

Source: Developed by the authors based on [1]–[15].

As shown in table VI, betterment in one aspect often comes with a cost in another. Therefore, the design of operational agents needs to be a compromise between effective, reliable, scalable and governance goals.

C. Comparison across Action Models

Action models describe different degrees of autonomy of operations. Read only diagnostic agents offer great value without posing significant risk. Advisory recommendation systems are devised to offer remediation actions to extend functionality but maintain human control over implementation decisions [5]. Models that involve human-in-the-loop provide a compromise between efficiency and safety, making them appealing for enterprise use. However, the advantages of write-enabled remediation agents, and fully autonomous operational systems, offer higher levels of automation, but come with significant issues of trust, compliance, accountability and operational risk management [3, 15].

D. Comparison across Evaluation Methodologies

There is great variability in evaluation methods across the literature. Synthetic benchmark evaluations are scalable and repeatable, but are often lacking in the ability to reflect operational complexity in the real world [13]. Historical incident replay is more realistic but at the same time offers controlled experimentation. Shadow-mode deployments allow for observing under real operational conditions, without adding production risks. While

production A/B testing provides the most evidence of operational value, at a higher implementation cost [14], human expert assessments are still vital in assessing interpretability and practical usefulness.

E. Cross-Taxonomy Observations

Common design patterns

There are some recurring design patterns that are observed in the literature surveyed. Most successful systems are based on both advanced reasoning architectures and retrieval-based grounding and controlled execution mechanisms. This combination not only enhances the diagnostic quality but also the operational safety [2] and [8].

Emerging architectural trends

New developments such as the use of multi-agent systems, hybrid grounding systems, and intelligent orchestration architectures. There is an increasing trend toward research on collaborative agents that are able to share specialized tasks among operations within workflows [10], [15]. At the same time, the addition of edge intelligence, digital twins and intelligent infrastructure platforms opens up more deployment opportunities [2], [6], [9], and [12].

Frequently observed limitations

Thanks to great progress, there are still some common constraints. The issues of reliability, hallucination risk, evaluation issues, governance challenges, and scalability remain to be an issue for production adoption. Previous research often shows high levels of benchmark performance, but lacks evidence for the effectiveness of operation in the real world [8], [13], [14]. These observations inspire the next step of investigation of the production academic divide and the transition of operational LLM agents from the experimental research LLM system to enterprise-scale deployment.

VII. THE PRODUCTION ACADEMIC GAP

A. Differences between Research Prototypes and Production Systems

There are significant differences between research prototypes of LLM agents and operational systems in production. Academic work often focuses on proof-of-concept and performance demonstrations, benchmarking, and small-scale, controlled experiments; while production environments have a greater focus on reliability, scalability, governance, security, and operational continuity. Synthetic data sets, small operational conditions, or single case studies are often used in the evaluation of many research systems, which lack sufficient complexity

.

in their enterprise infrastructure to represent real enterprise systems [13], [14]. As a result, the benefits that are taken from experimental conditions do not always translate to real-world applications.

In the case of research prototypes, they usually make simplifying assumptions about infrastructure heterogeneity, workload variability, and/or organizational constraints. By contrast, production systems need to operate across distributed cloud-native environments with thousands of interdependent services, databases, applications and network elements. Research on autonomous infrastructure management and intelligent DevOps automation points to the challenge of operating in environments with a large deployment scale [3], [15]. Likewise, studies of foundation models as platform infrastructure show that improvements to model performance are an important but not exclusive part of operational integration as the question of successful deployment is one of interoperability, governance and organizational adoption [7].

Another difference lies with the accountability for operations. While diagnostic accuracy is usually important in academic systems, traceability, auditability, rollback and compliance verification are important in production systems. Research that highlights operating-system intelligence and platform-level automation also acknowledges that readiness for deployment of a reasoned isn't only about the reason’s performance, but also about the operational protection and life cycle management requirements of the platform on which the reasoned is deployed [6]. This means that even well-developed research systems are not able to be implemented in the real world because they are not technically capable.

B. Requirements for reliability and safety

The most important factors that hinder production adoption are reliability and safety. The availability of services is directly affected by platform operations, as is the continuity of business and the customer's experience. This means that the action choices made by the LLM agent have to adhere to high accuracy standards. The resulting hallucinated diagnosis, wrong remediation suggestions or incorrect automated actions can cause service disruptions that go beyond the effect of the incident.

Robustness, alignment, consistency, and validation are consistently found to be required attributes of trustworthy deployment of LLM in studies on the topic [8]. In large-scale evaluations of LLM capabilities and limitations, uncertainty about the model's behavior is one of the key challenges to high-risk deployments, as highlighted in [1] and [13]. Unlike many consumer applications, operational environments are complex and can involve disparate components of infrastructure that cascade effects, cascading failures and significant

degradation of service.

The issue of safety becomes much more relevant with the presence of execution agents. Read-only diagnostic agents impact chiefly the decision support process, while write-enabled agents have an impact on production systems. The study of cybersecurity operations demonstrates the potential damage to the integrity of the system, security issues, and interference with incident investigations that can be caused by incorrect automated operations [5]. As a result, multiple layers of validation, approval, and rollback measures are often used in production deployments before allowing them to be executed independently.

C. Operational Governance and Compliance

Another key difference between research and production deployments is the governance and compliance needs. The regulatory, organizational and security factors affect all decisions made in enterprise infrastructures. In addition to technical performance, production systems need to meet the requirements of accountability, transparency, policy compliance and risk management in research studies.

Operational governance includes oversight of agent activity to ensure it is in line with organizational goals and agreed upon operational processes. The use of runbook guided execution, policy validation and controlled authorization are gaining importance in production-ready systems [3] and [15]. If not, an autonomous agent can take technically correct actions, but not actions in line with organizational policies or compliance requirements.

In highly regulated industries, particularly where changes to infrastructure are required, compliance issues are especially relevant, as changes have to be recorded and traceable. Explainability, transparency, and traceability are key factors for responsible use of trustworthy AI frameworks, as highlighted in studies [8]. The same rules are true for operational agents all recommendations and actions should be attributable, reviewable, and reproducible.

Issues in governance also apply to data management. Operational agents routinely log into logs, telemetry, incident logs and internal documentation. Studies exploring AI's use in operating systems and platform infrastructures reveal issues with information access control, data protection and secure knowledge usage [6], [7]. Governance frameworks need to adapt to the growing complexity of interactions between AI-influenced decision-making and enterprise operational processes as operational agents become more autonomous.

D. Human Oversight Requirements

Most of the production deployments are still human supervised. Although significant

progress has been made in the development of reasoning architectures, organizations are not generally inclined to delegate absolute control over the autonomous operational agents. Many deployments, however, use human-in-the-loop approaches with a mix of automation efficiency and human judgment and accountability.

Human supervision is crucial in several application areas. When making a recommendation in cybersecurity operations, analysts are likely to consider the potential impact of the recommended change before taking it into production, for fear of negative side effects [5]. Also, database diagnosis systems sometimes provide information and evidence to administrators, but cannot make changes without further administrator input. This team collaboration is a sign of the recognition that expertise goes beyond pattern recognition to context awareness and risk assessment.

Another area of research that is of particular interest for future operational scenarios is multi-agent reasoning, which includes the possibility of collaboration with human operators as well as AI agents [10]. Such partnerships enable distribution of responsibilities according to relative strengths. Many operational facts and figures can be processed by the agents, and patterns can be spotted, but humans can provide strategic reasoning, ethical judgment, and organizational understanding.

E. Scalability and Cost considerations

Real-world issues of scalability and costs are not always adequately addressed in academic research. Experimental studies are usually performed in controlled settings and under specific workloads whereas production deployments are deployed in large-scale infrastructures and cannot afford to interrupt their operations.

When scaled to enterprise size, the computational complexity of such high-level reasoning architectures with LLMs can grow significantly. Common approaches like chain-of-thought reasoning, multi-agent collaboration and retrieval-augmented workflows have often been seen to increase inference costs and resource usage. Research on edge intelligence and operational AI applications highlights the need for performance and efficiency trade-offs [2], [12]. Organizations consequently need to balance the sophistication of the diagnostics with sustainability operations.

Scale of infrastructure adds to the deployment complexity. Large enterprises can produce millions of logs, alerts, metrics and traces per day. To process these data streams, efficient information retrieval systems, scalable orchestration frameworks and optimized reasoning pipelines are needed. The architectures that support continuous large-scale analysis without significant latency are gaining attention in the research on AIOps and intelligent platform

operations [14].

F. Lessons learned from real world deployments

From these early experiences of deployment, there are several key takeaways about using LLM agents in operations. Based on these early deployments, there are a number of common lessons learned about operating LLM agents. First, retrieval and grounding mechanisms always improve the reliability and decrease the reliance on the parametric knowledge of the model. Operational documentation, runbooks, monitoring outputs, and historical incident data are often found to be effective in supporting systems, as opposed to standalone language models, in operational settings [2], [11], and [14].

Secondly, autonomy should be implemented gradually and not all at once. A frequent approach to successful deployments is to start by providing a read-only diagnostic support and then move to recommendation generation and controlled execution. Autonomous infrastructure management and intelligent orchestration frameworks have been investigated to help implement phased infrastructure adoption processes with gradual operational responsibility and ensure governance controls are maintained [3], [15].

Third, the user's acceptance will still require explainability. If reasoning pathways, sources of the evidence, and the context for the evidence, are made visible, operators are significantly more likely to trust the recommendations. It has been repeatedly stressed in the studies on trustworthy deployment and operational diagnostics of LLMs that the processes of decision making should be transparent [8], [11].

Fourthly, domain specialization is important for performance. It is typical that generic language models need grounding, retrieval and context adaptation to be effective in platform operations. As in investigations with database diagnostics, cybersecurity operations [5] and infrastructure management [11, 14], domain specific knowledge integration benefits are always shown.

Lastly, production success is based on integration of the ecosystem rather than model performance. Good deployments include reasoning capabilities and observability platforms, configuration systems, workflow engines and governance frameworks. This observation further strengthens the idea of operational agents as part of an operational ecology, instead of standalone AI applications, as discussed in [6] and [7] and [15].

VIII. OPEN CHALLENGES AND RESEARCH GAPS

A. Evaluation Standardization

One of the significant research gaps is the lack of standardized evaluation methods.

Comparisons of the existing studies are not easy to make because different metrics, datasets, and testing procedures are used in different studies, and some studies are not reproducible [13], [14].

B. Robustness and Hallucination Mitigation

A major risk is hallucination as incorrect diagnoses or remediation suggestions can lead to operational hazards. Future work should focus on finding ways to create more factually consistent, uncertain estimation and validation processes [1], [8].

C. Long-Term Memory and Context Management

Operational incidents often occur over a long time and have a lot of historical dependencies. Increasingly, current agents are having trouble keeping track of the long-term context in an investigation, requiring better memory architectures [2, 11].

D. Multi-Agent Coordination

Coordination between specialized agents is an important direction for research, since the operational environment is becoming more and more complex. Small scale operational deployment has been shown to be a possibility, but remains somewhat uncertain based on previous studies [10].

F. Technology and Innovation

There is still a critical need for improved transparency for operational adoption. Future systems need to deliver interpretable reasoning paths, mechanisms to attribute evidence to support levels of trust, and assessments of confidence to aid in the development of trust [8], [13].

The student will identify and explain how to build an app that is secure and resilient to adversarial activity.

Operational agents are still susceptible to adverse inputs, quick manipulation and malicious operational data. Enhancing resilience to these threats is important research goal [5], [9].

G. Autonomous Remediation Safety

Autonomous execution safe is not fully addressed. Future research should focus on creating validation frameworks, policy enforcement mechanisms and rollback safeguards which enable trustworthy remediation procedures [3], [15].

H. Benchmark and Dataset Limitations

Many of the current benchmarks cannot achieve a realistic operational complexity. More representative data sets of true incidents, true infrastructure diversity, and real constraints of organization are needed to further this work [14].

.

IX. RESEARCH AGENDA

A. Towards Standardized Operational Benchmarks

Future studies should define norms for community-based evaluation of the accuracy of diagnosis, reasoning quality, effectiveness of remediation, and safety during operation under realistic conditions.

B. Safe Autonomous Remediation Frameworks

Architectures that enable trustworthy execution, by combining autonomy with validation, policy enforcement and risk-aware decision-making mechanisms should be the focus of research.

C. Human-Agent Collaborative Operations

Collaboration of human experts with intelligent agents to take advantage of complementary strengths and enhance reliability and decision quality should be a focus in future operational environments [8, 10].

D. AgenticObservability Platforms

Future generations of observables should also have direct integration of reasoning, retrieval, and operational intelligence in monitoring systems, enhancing situational awareness [14].

E. Multi-Agent Incident Management Systems

A more promising approach for dealing with increasingly complex infrastructures is to coordinate multi-agent ecosystems that can investigate, diagnose and respond in a distributed manner across the infrastructure [10].

F. Self-Improving Operational Agents

Operational feedback, incident outcome, and organization knowledge based continuous learning frameworks may enhance long-term effectiveness and adaptation [6, 7].

G. Production-Ready Evaluation Frameworks

Evaluation methodologies need to be further developed to address the impact of academic experiments on enterprise deployment with shadow-mode testing, governance assessment and operational performance metrics. Together, these research directions serve as a stepping stone toward creating robust, trustworthy, scalable, and ready-to-use platform intelligence systems with operational LLM agents.

X. CONCLUSION

The world of LLM agents for platform operations, incident response, diagnostics and automated remediation has rapidly evolved, and this survey has explored all these aspects. The study combined a wide range of literature from recent advancements in large language

models, operational intelligence, AIOps, DevOps automation, cybersecurity operations, and infrastructure management, and synthesized them to a common framework for analysis [1] - [15]. The results show that operational LLM agents are starting to evolve from experimental decision-support systems to intelligent systems that can reason, ground, plan, and coordinate in complex operational settings. The taxonomy used in this survey was based on four axes: reasoning architecture, grounding strategy, action model and evaluation methodology, and was used to frame a structured view of the design space of operational agents and to detect common patterns in this space in the literature.

The analysis showed that advanced reasoning paradigms like Chain-of-Thought, ReAct, and plan-and execute architectures have proven to significantly outperform traditional automation approaches, particularly in the context of diagnosis, investigation, and remediation planning, and are capable of contextual diagnosis, adaptive investigation, and structured remediation planning.Advanced reasoning paradigms, including Chain-of-Thought, ReAct, and plan-and-execute architectures, were found to greatly improve over traditional automation approaches, especially for diagnosis, investigation, and remediation planning, and to be able to perform contextual diagnosis, adaptive investigation, and remediation planning in a structured manner.

Likewise, retrieval systems, operational knowledge repositories, runbooks, and outputs constrained in tools contribute significantly to reliability and minimize hallucination risks [2, 8, and 11]. The survey, however, showed that there is continued disconnect between research and deployment. Although many studies show good results in ideal conditions, in reality, there are difficulties with trustworthiness, governance, scalability, safety, explainability, compliance and standardization of the evaluation of the models [3], [5], [6], [7], [13], [15].

Further research on setting realistic operational limits, making the systems more robust to hallucinations, building scalable multi-agent coordination structures, and enhancing more autonomous remediation methods that are safe to put into practice. It is also important to focus the development of human-agent collaboration [8, 9], LTM architectures [12, 14], adversarial resilience, and production-ready evaluation methodologies. It is crucial to overcome these obstacles to enable reliable, trustworthy, and enterprise scale deployment of operational LLM agents that will transform next generation platform operations.

REFERENCE

1. Rane, N. L., Tawde, A., Choudhary, S. P., &Rane, J. (2023). Contribution and performance of ChatGPT and other Large Language Models (LLM) for scientific and

research advancements: a double-edged sword. *International Research Journal of Modernization in Engineering Technology and Science*, *5*(10), 875-899.

2. Friha, O., Ferrag, M. A., Kantarci, B., Cakmak, B., Ozgun, A., &Ghoualmi-Zine, N. (2024). Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness. *IEEE Open Journal of the Communications Society*, *5*, 5799-5856. https://doi.org/10.1109/OJCOMS.2024.3456549
3. Vangoor, V. K. R. (2022). Autonomous DevOps Infrastructure: AI-Driven Lifecycle Management of Large Scale Linux Server Ecosystems. *Journal of Management and Science*, *12*(4), 156-163.https://doi.org/10.26524/jms.12.83
4. Liu, X., Wang, J., Sun, J., Yuan, X., Dong, G., Di, P., ...& Wang, D. (2023). Prompting frameworks for large language models: A survey. *arXiv preprint arXiv:2311.12785*. Liu, X., Wang, J., Sun, J., Yuan, X., Dong, G., Di, P., ...& Wang, D. (2023). Prompting frameworks for large language models: A survey. *arXiv preprint arXiv:2311.12785*. https://doi.org/10.48550/arXiv.2311.12785
5. Kaheh, M., Kholgh, D. K., &Kostakos, P. (2023). Cyber sentinel: Exploring conversational agents in streamlining security tasks with gpt-4. *arXiv preprint arXiv:2309.16422*. https://doi.org/10.48550/arXiv.2309.16422
6. Zhang, Y., Zhao, X., Li, Z., Cheng, G., Yin, J., Zhang, L., & Chen, Z. (2024). Integrating Artificial Intelligence into Operating Systems: A Survey on Techniques, Applications, and Future Directions. *arXiv preprint arXiv:2407.14567*. https://doi.org/10.48550/arXiv.2407.14567
7. Thota, M. R. (2022). Foundation models as platform infrastructure: Integrating large language models into internal developer platforms for scalable productivity. *International Journal of Scientific Research in Science and Technology*, *9*(5), 853-864. https://doi.org/10.32628/IJSRST2295163
8. Liu, Y., Yao, Y., Ton, J. F., Zhang, X., Guo, R., Cheng, H., ...& Li, H. (2023). Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*. https://doi.org/10.48550/arXiv.2308.05374
9. Mohsin, A., Janicke, H., Nepal, S., & Holmes, D. (2023, November). Digital twins and the future of their use enabling shift left and shift right cybersecurity operations. In *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)* (pp. 277-286). IEEE. https://doi.org/10.1109/TPS-ISA58951.2023.00042

.

10. Tsao, W. K. (2023, December). Multi-agent reasoning with large language models for effective corporate planning. In *2023 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 365-370). IEEE. https://doi.org/10.1109/CSCI62032.2023.00065
11. Zhou, X., Li, G., Sun, Z., Liu, Z., Chen, W., Wu, J., ...& Zeng, G. (2023). D-bot: Database diagnosis system using large language models. *arXiv preprint arXiv:2312.01454*. https://doi.org/10.48550/arXiv.2312.01454
12. Javaid, S., Fahim, H., He, B., & Saeed, N. (2024). Large language models for UAVs: Current state and pathways to the future. *IEEE Open Journal of Vehicular Technology*, *5*, 1166-1192. https://doi.org/10.1109/OJVT.2024.3446799
13. Hadi, M. U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., ...&Mirjalili, S. (2023). Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea preprints*, *1*(3), 1-26.
14. Zhang, L., Jia, T., Jia, M., Wu, Y., Liu, A., Yang, Y., ...& Li, Y. (2024). A survey of aiops for failure management in the era of large language models. *arXiv preprint arXiv:2406.11213*.https://doi.org/10.48550/arXiv.2406.11213
15. Vollem, S. (2024). From deterministic pipelines to intelligent orchestration: A transformer-driven framework for LLM-augmented DevOps automation. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)*, *7*(1), 9964-9975. https://doi.org/10.15662/IJRPETM.2024.0701009

.