SSRG International Journal of Computer Science and Engineering
ISSN: 2348–8387 / https://doi.org/10.14445/23488387/IJCSE-V13I8P102

Volume 13 Issue 8, 11-28, August 2026

Original Article

A Multi-Agent Generative AI and Machine Learning Framework for Autonomous Incident Intelligence, Predictive Service Assurance, and Customer Experience Optimization in Cloud-Native Digital Service Platforms

Darshan Bhansali

[1]*Principal System Architect, AT&T – Consumer Technology Experience, Dallas, TX, USA.*

[1]*Corresponding Author : bhansalidarshan13@gmail.com*

Received: 19 June 2026 Revised: 27 July 2026 Accepted: 15 August 2026 Published: 29 August 2026

Abstract *- The growing complexity of the online service platforms based on the cloud has posed new challenges to the traditional IT operations unprecedentedly, and reactive incident management is no longer adequate in ensuring the reliability of service and customer satisfaction. The authors present a Multi-Agent Generative AI and Machine Learning (MAGML) system for autonomous incident intelligence, as well as predictive service assurance and customer experience optimization in this article. The domain-specific ecosystem of intelligent agents to identify root causes, investigate incidents, assess the impact, draw up remediation plans and learn about incidents is all incorporated under the framework incorporation. With a Generative AI-based engine, heterogeneous operational data, such as telemetry, logs, distributed traces, dashboards, and past incidents, are converted into contextual operational intelligence, automated incident summaries, diagnostics, and remediation suggestions. Infrastructure performance is combined with application observability as well as customer journey analytics and business key performance indicators to develop machine learning models that are capable of predicting the occurrence of a service failure prior to the disruption of the service affecting customers. Instead, reinforcement learning enables learning to remediate itself based on outcomes of the learning, and an Operational Knowledge Graph can model the connection between services to increase correlations between events, decrease unnecessary alerts about events, and accelerate the process of root cause detection. Another Customer Experience Intelligence overlay that measures the business impact, the severity of the incident, and SLA/SLO compliance or an AI-based operational copilot that supports the engineer with runbooks, intelligent predictions, and diagnostics. With heterogeneous observability data packets that comprised the HDFS Log Dataset, the BlueGene/L (BGL) Log Dataset, the OpenTelemetry distributed traces, Kubernetes operational metrics and the Application Performance Monitoring (APM) data, the given framework has been experimentally tested and compared with six current approaches, including the threshold-based monitoring, the Random Forest, the LSTM, the Knowledge Graph-based RCA, the RAG-enabled LLM incident intelligence, and the traditional enterprise AIOps platform. Laboratory results demonstrate that the proposed MAGML model achieves an accuracy, precision, recall and F1-score of 98.4 per cent, 98.1 per cent, 98.3 per cent, and 98.2 per cent, respectively, and reduces MTTD to 3.1 minutes and MTTR to 14.6 minutes. It also boasts of 98.6% accuracy RCA, 98.3% accuracy alert-correlation, 99.1% SLA compliance and 99.0% recoverability success, which confirms that it is solid in its predictability incident management model, autonomous remediation model, service assurance model, and customer-focused operation insight of cloud-native digital service.*

Keywords *- Multi-Agent Systems; Generative AI; AIOps; Predictive Incident Management; Cloud-Native Microservices; Customer Experience Intelligence; Reinforcement Learning.*

1. Introduction

The foundation of a high level of digitalisation of business operations has radically altered the way modern-day applications are developed, delivered, as well as their management. Cloud-native infrastructure, microservices, containers, Kubernetes orchestration, serverless computing, and distributed application ecosystems have taken root in becoming the backbone of digital businesses in all industry sectors like banking, healthcare, telecommunications, retail, manufacturing, and e-commerce. These technologies are

also allowing organizations to enable new digital services fast due to their unprecedented scalability, elasticity, and flexibility in deploying new services. However, there has also been a great deal of operational complexity with the increasing adoption of cloud-native technologies. One business operation can pass through hundreds of loosely coupled microservices being hosted on a variety of clusters, cloud service vendors, and locations. It means that the incidents involving the infrastructures-, networking-, application services-, database-, and third-party APIs can potentially impact the whole ecosystem swiftly, and thus, it is more difficult and more difficult to find, determine, and reclaim cases.

The conventional IT Operations (ITOps) and monitoring planning mostly rely on predefined thresholds, alerting plans triggered by guidelines and manual response plans. Although they have been applied to serve-enterprise systems over the years, they are now inappropriate to address the extremely dynamic cloud-native environments of elastic workloads, distributed dependency, continuous deployments and dynamic changing service topology. An ever-growing stream of telemetry data is generated by modern observability platforms: infrastructure metrics, application logs, distributed traces, events, performance metrics, customer experience metrics, synthetic monitoring results, and business performance indicators. Whilst these sources of data can provide in-depth information on how systems act, it is also a challenge to extract out of this heterogenous and high velocity information meaningful operational intelligence. The alert storms and numerous alerts, lack of contextual data, and surgery of dependencies commonly overburden the operations teams and result in increased Mean Time to Detect (MTTD), increased Mean Time to Resolve (MTTR), service downtimes, and bad customer experiences.

The introduction of Artificial Intelligence has made the analytics of operations automated and made incident control more efficient to support IT Operations (AIOps). Available AIOps solutions take advantage of machine learning solutions in anomaly detection, event correlation, predictive analytics, and root cause analysis. However, most of the existing resolutions are limited by isolated analysis models that carry out the various functions in isolation without a combined thinking process along the entire incident lifecycle. Moreover, classical machine learning models do not have contextual knowledge of how they work; they include truly huge amounts of feature engineering, and they cannot be described satisfactorily in terms of how they arrive at their predictions. These restrictions impair their ability to support independent functionality with more and more advanced cloud-native environments.

The recent advances of the Generative Artificial Intelligence (Generative AI) and the Large Language Models (LLM) have been accompanied by the new features of interpretative, unstructured operational data and the generation of operational intelligence that can be read by humans. Generative AI can learn all operational logs, incident reports, configuration files, runbooks, knowledge articles, change requests and engineering documentation and generate contextual explanations, incident summary, remediation latter, and decision support, compared to traditional predictive models. Such characteristics can contribute significantly towards serving the liaison of artificial intelligence systems and Site Reliability Engineering (SRE) teams since it facilitates immeasurable volumes of technical objects, transforming them into action. Nonetheless, implementing a single LLM as its own assistant is inadequate to handle complex operating conditions where several tasks requiring analytical operations have to be done in parallel, with strict reliability and latency constraints.

Recently, the term agentic Artificial Intelligence has proven to be an effective reference frame for determining autonomous intelligent systems, able to plan, reason, cooperate and perform more intricate workflows. Multi-agent systems are systems that decentralize the functionality of operations within them, assigning particular aspects of it to dedicated intelligent agents, each of which is given the responsibility of some aspect of the operation, exchanges information and coordinates actions via collaborative mindsets. In addition, a purpose-analog agent is specially deployed in the cloud-native incident management to establish a vigil in the infrastructure's health, detect anomalies, and correlate around an alert, initiate a root cause analysis, gauge the implications on customers, propose remediation steps, confirm recovery plans, and learn continuously based on incident results. These collaborative types of intelligence can greatly benefit compared to centralized monolithic types of AI systems in terms of scalability, fault tolerance and transparency in the operations of the system.

At the same time, machine learning has facilitated predicting of incidents, and it has transformed the operations' operating strategy to be pro-reactive instead of proactive for service assurance. Predictive models utilise past occurrences, resource access, application telemetry, customer behaviour and business key performance indicators, instead of responding to a damaged service; blue-ocean officers can be employed to determine probabilities of a system running poorly, prior to customers realising that their service was affected. Early prediction enables an early preparation, such as the diversion of the workload, the scaling of the capacity, fitness of the configuration, traffic rerouting or automatic recovery operations, hence the reduction of operational risk and service continuity. Independent operations can also be reinforced with the help of reinforcement learning, where autopiloting schemes will

be able to maximize the remediation system by a continuous stream of input feedback, and the effects of the occurrences of earlier events will be collected. Instead of runbooks, reinforcement learning agents will have learned the optimal recovery policies through trial and error in real workplaces, and it will continue to improve the quality of the judgments and minimise the recovery time.

In the midst of these changes in technology, there are two major research tasks that are yet to be determined. The operational intelligence systems either in use are highly responsive to infrastructure health, performance of applications and customer experience as well as business impact individually, and lack a unifying connection between these areas. It is also hard to carry out the root cause analysis due to the existing enterprise applications, which are microservices, APIs, databases, message brokers and external cloud services, as they are massively coupled in their integration dependencies. Also, attempting to remove redundant notifications is not a common operation of an alert correlation system due to the cascading failures that not only increase the operational load, but also slack the speedy response. Uniting Generative AI, multi-agent coordination, reinforcement learning, explainable artificial intelligence, knowledge graphs and customer experience intelligence together into a single autonomous operational system has been little aided by the existing frameworks.

The other prominent drawback is that it will lose touch with the technical incident management and the business-oriented service assurance. Traditional monitoring systems are mainly concerned with achieving maximum infrastructure and performance of applications without consideration of the reality customer experience and business impact due to service degradation. The decisions of operating a business online need to place more importance on incidents, customer experience, transaction importance, revenue ramifications, Service Level Agreement (SLA), and Service Level Objectives (SLOs). The Customer Experience Intelligence, integrated into operations, will help organizations manage their engineering resources more efficiently, reduce business losses, and increase user satisfaction with smart ordering of critical incidents.

Despite the significant strides achieved by AIOps, Generative AI, agentic AI, and predictive analytics, it is evident that there exists a gap in research to establish a unified autonomous framework that can unify these characteristics in the entire incident-management and service-assurance life cycle. The current state of AIOps tends to be focused on one of the operational procedures, such as anomaly detection, event correlation, root cause analysis, or even predictive monitoring. These functions are likely to be actualized as non-integrative units of analytics, and therefore they are not able to marshal reasoning, judgment, remediation and continuing learning between multiple stages of an affair. Operational assistants created using generative AI have been shown to aid in understanding logs, incidents, and technical documentation, yet most of the current methods are more advisory and lack organizational multi-agent reasoning, autonomous remediation and feedback-optimal decisions. Also, machine learning predictive models are more prone to be infrastructure or application level centered, and fail to include customer journey behaviour, business KPIs, service dependencies and SLA/SLO data into a single predictive and prioritization workflow.

The lack of integration of technical service health, customer experience and business impact is another critical gap in the research. Infrastructure monitoring and traditional monitoring and AIOps tools are typically used to monitor infrastructure availability, application performance, technical probes and the actual impact of service degradation on end users, key transactions, money-generating services, and business objectives is typically considered separately or post-incident. Moreover, microservice dependencies can change over time, failures can be cascaded, and cloud-native topologies might change swiftly, and current event-correlations seems incapable of discerning between primary causality and secondary symptoms. This can lead to operations teams getting huge amounts of correlated or redundant alerts that lack enough contextual information to help them decide on the most correct course of action. The lack of an ever-evolving knowledge representation and feedback-driven remediation cycle compounds the capacity of up-to-date systems to learn about past recovery successes and enhance future incident-response choices.

The resulting problem can be put as follows: How can a cloud-native digital service platform autonomously monitor and anticipate service degradation, correlate heterogeneous operational indicators, pinpoint root causes, assess customer and business impact, propose and execute suitable remediation, and understand remediation outcomes, continuously maintaining SLA/SLO targets? The solution to this issue does not lie in an isolated monitoring system or one Generative AI assistant. It requires an integrated framework that has the potential to integrate multi-agent intelligence, Generative AI, predictive machine learning, knowledge graphs, reinforcement learning, explainable AI, observable data, and customer experience intelligence to form a united functioning decision-making mechanism.

In this way, the main research need that the study satisfies is the lack of a single, customer-conscious, predictive and continuously learning cohesive autonomous incident intelligence framework connecting detection, diagnoses, impact assessment, remediation and learning to the lifecycle of the whole cloud-native services. The proposed research fills this gap since it develops Multi-

Agent Generative AI, and a Machine Learning Framework of Autonomous Incident Intelligence, Predictive Service Assurance and Customer Experience Optimization. This framework encompasses domain-specific intelligent agents, a Generative-AI Operational intelligence solution, predictive machine learning models, a service-dependency knowledge graph, remediation via reinforcement learning, explainable decision support, and a Customer Experience Intelligence layer to deliver progressively autonomous service operations.

To fill these research gaps, this paper has suggested a Multi- Agency Generative AI and Machine learning Framework of Autonomous Incident Intelligence, Predictive Service Assurance and Customer experience Optimization in cloud-native Digital service platforms. The proposed architecture is a teaming ecosystem of specialty intelligent agents in order to identify malfunctions, triage accidents, underlying cause, dependency, business impact and recovery strategy and to self-train. A Generative AI engine inputs heterogenous observability data, such as continuous telemetry, logs, distributed traces, dashboards, past incidents, engineering documentation and runbooks, and provides contextual operational intelligence, automatic incident recommendations and explainable remediation recommendations. The machine learning models are proactive to predict a bad service, on the basis of infrastructure indicators, application observability, customer journey analytics and business performance indicators. Knowledge graph can be used to translate changing service dependencies to improve event correlation, minimize the number of redundant alert triggers and quickly find root causes. Reinforcement learning can be utilized to continually optimize automated remediation plans by contrasting new information with the current state, thus providing adaptive feedback. Additionally, the Customer Experience Intelligence layer also analyses the real-time impact of the business and assists with prioritization of the intelligent customer experience incident based on the importance of the operations, experience to the customer, and beyond the intended customers by the organization.

The suggested architecture is geared towards providing autonomous AIOps, which engages agentic AI, Generative AI, predictive machine learning, reinforcement learning, explainable artificial intelligence, knowledge graphs, cloud observability, and customer-centered operational intelligence into a single architecture. The framework will aim at ensuring a significant reduction in Mean Time to Detect (MTTD) and Mean Time to Resolve(MTTR), greater resiliency of operational processes, increased reliability of digital service delivery, improved compliance with SLA/SLO, and improved customer experiences. With the ongoing growth of businesses' digital ecosystem to cloud native, the proposed research offers a reconfigurable, smart and scalable base of the next-generation autonomous service operations that is capable of delivering a customer sensitive and business-conscious platforms of digital services.

2. Background and Related Work

2.1. Background

The implementation of cloud-native technologies within a relatively limited time has essentially affected the provision and patrolling of the digital services. Modern systems are becoming more and more based on microservices, containers, orchestration via Kubernetes, serverless computing, distributed databases, application programming interfaces, and multi-cloud systems to offer scalable and always-available services. Although they provide some flexibility and increase the agility of deployments, they introduce a lot of operational complexity as well. A transaction involving a customer might go through numerous different related services, and locating, identifying and correcting service degradation becomes difficult. Conventional means of control that are largely dependent on predetermined set points and regulations prescribed by human factors are therefore likely to result in large volumes of alert signals with little contextual information to immediately operate on.

Along with the emergent alternatives of anomaly detection, event correlation, predictive maintenance and automatic incident resolution, has come the advent of the concept of Artificial Intelligence to accomplish IT Operations (AIOps). Both past and current telemetry allow machine learning patterns to identify abnormal behaviours of operation, as well as predict potential failures. Generative AI and Large Language Models have enabled operational teams to process unstructured logs, incident reports, runbooks, and technical documentation and generate contextual explanations and remediation proposals in more modern times. Most of the methods available, however, consider a merger of functions of the operational components, contrary to providing built-in intelligence across the entire incident lifecycle.

Reliability of the cloud-native services should also be aware of how health relates to the technical system and customer/business impacts. A relatively small reduction in a high-value transaction customer can have a significant business outcome, despite an anomaly in infrastructure, potentially or not signifying a critical incident. In this way, the next-generation service operations should include observability analytics, predictive analytics, machine learning, Generative AI, knowledge graphs, autonomous remediation, and customer experience intelligence.

In this context, the customer-conscious and autonomous operational intelligence is a significant research direction. This may be accomplished through the integration of a unified multi-agent structure that can bring a coordinated

use of specialized AI agents to carry out detection, diagnosis, impact evaluation, remediation, and ongoing learning, thereby providing proactive assurance of services, faster incident response, generalized SLA/SLO compliance, and enhanced customer encounter in complex cloud-first digital service environments.

2.2. Related Work

The most centered use of cloud-native computing, micro services, distributed applications and artificial intelligence has transformed the modern IT operations significantly. Old methods of monitoring and incident management have been transformed to Artificial Intelligence: IT Operations (AIOps), artificial cloud management and smart service assurance. However, more recently involving the use of Generative Artificial Intelligence (GenAI), Large Language Models (LLMs), multi-agent systems and predictive analytics, the project of building an intelligent operational platform that could be leveraged to automate the process of incident detection, diagnosis, remediation and lifelong learning was advanced to a more advanced degree. The section conducts a literature survey on the problem of cloud operations, AI-based incident management, multi-agent intelligence, and autonomous cloud services and outlines gaps existing in the literature, which the proposed framework fills.

Platforms based on the cloud have brought about new levels of scalability and flexibility in deployment, but this has also brought about a new level of complexity in the operations with the advent of distributed services, non-homogeneous environments and dynamic provisioning of resources. Alonso et al. [1] provided the CloudOps reference framework as the shift in the traditional cloud administration towards intelligent operations operationalization in a cloud-continuum environment. They have appeared talking about architectural concepts, realization difficulties, resource control, surveillance, automatization, and life cycle control of cloud services. Though the suggested framework offers an outstanding basis of cloud operations, it is more based on infrastructure handling and coordination rather than incorporating clever incident rationale, autonomous decision-making, analysis of customer experience, or cooperative artificial intelligence agents.

There is a vastly increased impact on software engineering and IT activities with the fast development of Large Language Models. In a vast scale systematic survey on the utilization of LLM in software engineering (Hou et al. [2]), it comes out that the application of LLM can be used in code generation, software maintenance, testing, documentation and software development. These issues that they listed were: the reliability problem, the necessity of having hallucination, explainability and deployment, as more underlying models are employed in the creation of software. Nonetheless, the review mostly dwells on the software engineering activities instead of cloud operations, predictive service assurance or the autonomous incident intelligence.

The concept of Retrieval-Augmented Generation (RAG) is now one of the most practical ways of providing factual consistency and situational awareness of systems on the Lisp machine. The combination of the extrinsic knowledge retrieval model and generative language model proposed by Lewis et al. [3] is the RAG model, which is intended to stimulate the type of reasoning based on knowledge. In addition to dynamically updating sources of knowledge by adding retrieval procedures to generate responses based on constant parameters, language models can produce responses based on context by updating the pre-trained parameters dynamically. Despite its introduction into the natural language processing context, the RAG can be used to offer a robust basis to operational intelligence systems that must then reason in telemetry, incident catalogues, operational runbooks and organizational knowledge bases.

One of the current trends in approaching the target of multi-agent systems in recent history has been a very strong paradigm towards the development of autonomous AI-ecosystems with the capacity of rationale communally and subsequently make decisions in a decentralized approach. A survey of the research in the area of LLM-based multi-agent systems led by Han et al. [4] established that there were numerous research issues which they incorporated: agent coordination, the communications protocols, planning, memory management, task decomposition and collaboration reasoning. Their work has shown that their various specialized agents outperform single-agent structures at the complex tasks of distributed expertise. However, the survey itself is still theoretical and is not aimed at cloud-native incident management or operational intelligence.

Similarly, Wu et al. [5] suggest AutoGen, an extendable multi-agent conversation head, capable of autonomously collaborating with multiple agents, being driven by an LLM. AutoGen progress plan work, role division, repetitive reasoning and multi-user performance that makes AutoGen extremely appropriate in the complex process of work of the enterprise. Even though AutoGen may be relevant in terms of the architectural foundation of agent collaboration in its use, it is not that concerned about the cloud observability, incident intelligent and service reliability engineering as compared to generic AI applications.

The increased intelligence applications complications have made the software engineering of AI-driven systems interesting and noticeable. Martinez-Fernandez et al. [6] conducted a complete survey where the software engineering methodologies of AI-based systems reported on

the development processes, the testing strategies, the lifecycle management, deployment and quality assurance. The need to introduce AI engineering principles to the already existing software development was also highlighted in the survey, yet it did not explore the autonomous cloud operations and AI-based operational intelligence.

There has been great research in the application of Generative AI to the discipline of software engineering as well. General Generative AI should boost human knowledge in the field of software engineering instead of substituting it, as suggested in the Copenhagen Manifesto by Russo et al. [7]. Their labor advocates the fact that AI systems based on human centricity are more productive to developers, as well as transparent, accountable, and human-controlled. Though the specified principles could be directly translated to the operational environment, neither self-reliant incident management nor ingenious cloud operations are examined in the literature.

A number of recent studies have focused particularly on the use of LLMs in incident management. One such recommendation system is an XPERT, suggested by Jiang et al. [8], that is an LLM proposal generator, which aids engineers in incident investigation by intelligent query recommendations. The proposed approach is time-saving in case of investigations, as it will suggest valid diagnostic questions, based on knowledge of previous occurrences. XPERT is primarily an incident exploration tool, though it is highly beneficial, unlike incident prediction and subsequent incident intelligence, analysis of customer impact, and automatic remedies.

The language models have added to diagnostic cloud incidences. One possible automated root cause resolution system described by Chen et al. [9] makes use of the LLMs to analyze cloud events based on data in the operational logs, telemetry, and past experience. The methodology portrays enormous advancements in the root cause identification as compared to the mainstream rule-based mechanisms. However, this framework is more based on the post-incident diagnosis and lacks predictive analytics, reinforcement learning and multi-agent orchestration.

A second promising field of research is to comprehend large-scale outages. Jin et al. [10] created a framework, which was used to evaluate and generalise the production outage automatically with the help of the LLM. It also allows engineering teams to communicate more effectively with the other teams, and the speed of decision-making since it creates short notices, which are formed on detailed information about their activities. The idea provided, however, is good, but instead of operating on independent operation intelligence across the incident lifecycle, the provided approach tends to concentrate more on reporting about the incident after its occurrence.

The emergence of standalone clouds has promoted the investigation of intelligent AI agents that are able to perform the operational activities themselves. AIOpsLab proposed by Chen et al. [11] is a benchmarking scheme of the whole-of-cloud to evaluate the performance of AI agents. Their work offers standardization of the assessment process to identify agent performance in terms of monitoring, diagnosing, planning and operational decision-making. Even though AIOpsLab formulates fruitful evaluation benchmarks, they fail to formulate an end-to-end operational framework that is developed out of customer experience intelligence, knowledge graphs, and reinforcement learning.

Following this orientation, Shetty et al. [12] have factored in the issues and design considerations of creating AI agents in an autonomous cloud environment. Their article discussed the requirements of architecture, scalability, reliability, safety, co-operation, and human-AI interaction, which should be acquired by operational AI agents. Despite these design principles informing the sound architectural direction urge, it still maintains a more conceptual character and does not suggest a combination of operational frameworks that will provide presumptions of predictive service and business-conscious incident priority.

The help of LLM has also been enjoyed by the SM Cloud monitoring. Yu et al. suggest MonitorAssistant, which simplifies the monitoring of cloud service since engineers can interact with the monitoring platforms in terms of natural language [13]. The framework will help users to get oriented with the dashboards, get operational implications and interpret monitoring representation using conversational interfaces. Autonomous multi-agent orchestration, or autonomous self-healing, is not considered autonomous, but human-assisted monitoring is.

One of the main challenges with cloud operations is the identification of the root cause. Dogga et al. [14] introduced the AutoARTS, a taxonomic structure for labeling and classifying the root cause of incidents that take place when the production of Microsoft Azure is involved. Their work can be utilized as useful datasets and input to the analysis of incident classification and implemented in machine learning analysis in operational analytics. Nevertheless, the proposed framework focuses on the rank of events, instead of predictive intelligence or thinking with AI.

The empirical test on production incidents has also been researched. Ghosh et al. [15] investigated the consequences of the massive cloud service outages in order to understand the common causes of failures, work bottlenecks, and how to address them. They find that they need greater automation, awareness of what is happening and smart operational support in the successful incident management. Nevertheless, their work mostly offers empirical

measurements, instead of an independent AI system that can ensure services proactively.

Based on the literature reviewed, gaps in research are apparent and very significant. Most of the literature discusses one of the components of smart operations, such as cloud management [1], software engineering [2], retrieval-augmented reasoning [3], multi-agent collaboration [4], [5], AI engineering [6], human-industrial Generative AI [7], incident query assistance [8], automated root cause analysis [9], incident summarization in the cloud [10], AI agent evaluation [11], autonomous cloud architectures [12], monitoring assistants [13], labeling root causes [14], or empirical incident analysis [15]. Nonetheless, none of the current models entail a mixture of the multi-agent collaboration, Generative AI, predictive machine learning, reinforced-based Heal, self-learning, knowledge graph dependency, customer experience intelligent, business impact prediction and AI-based copilots within a framework of a cloud-native digital service platform.

The proposed study is capable of addressing these limitations through the advent of Multi-Agents Generative AI and Machine Learning Framework, where autonomous incident intelligence, predictive service assurance, explainable operational reasoning, intelligent remediation planning, customer-centric prioritization and constant operational learning are unified. The proposed framework will bring the state of the art in autonomous cloud operations and next-generation AIOps to the modern digital service platforms by means of a single incarnation of integrating such complementary technologies into one intelligent network.

3. Methodology

3.1. Overview of the Proposed Framework

The growing trends of cloud-native and orchestration systems, micro-services with containers, and distributed applications have greatly enhanced the scalability and responsiveness of contemporary internet-based service platforms. Nonetheless, such environments create very dynamic behavior of operations, which do not go hand in hand with the traditional system of monitoring and management of incidents. The conventional AIOps engines largely depend on rule-of-thumb notifications, custom-made alert notifications, solitary cloud learning models featuring slight to no contextual information, autonomous thinking and decision making. This leads to alert fatigue, lack of analysis of root causes, lengthy service outages and inadequate customer experiences in engineering teams.

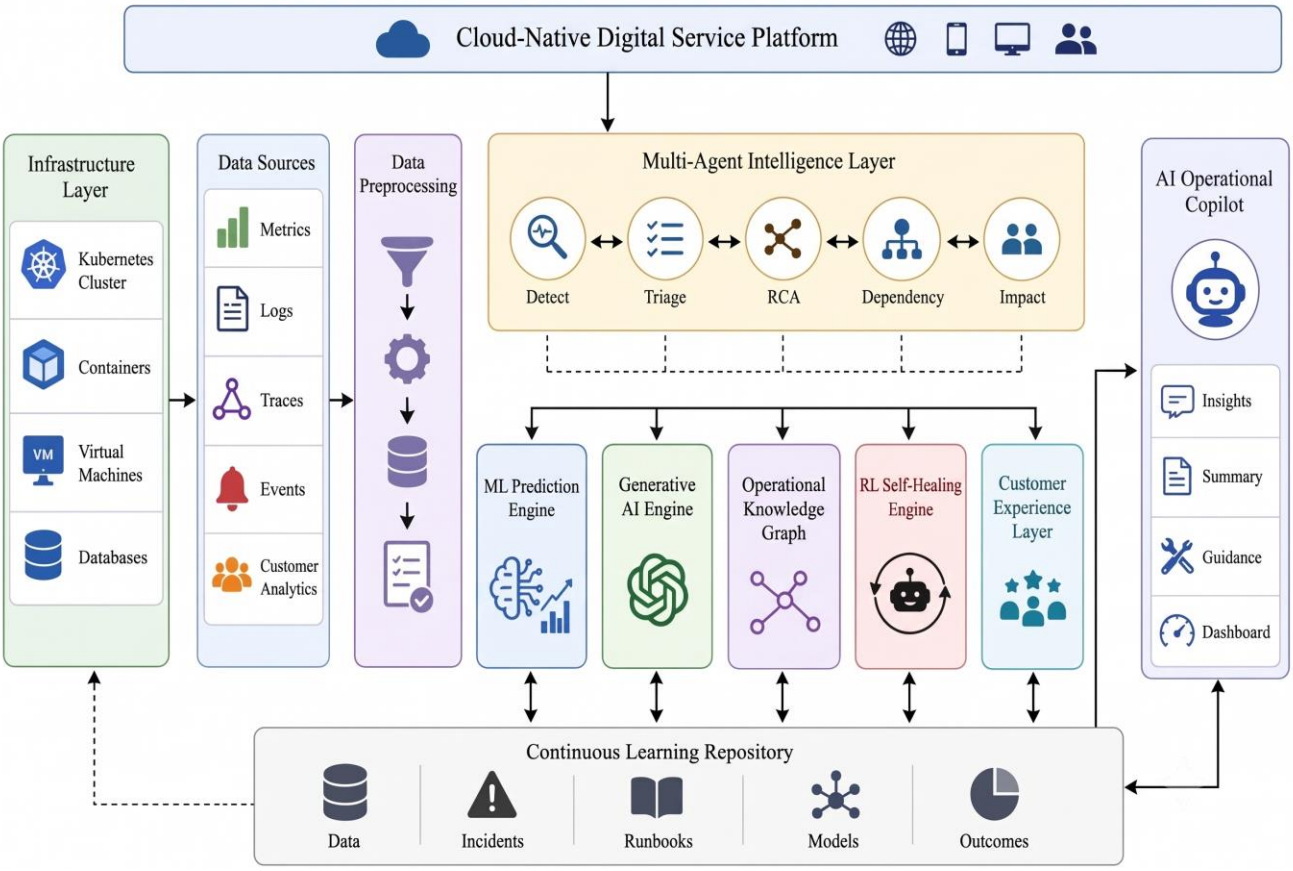

Fig. 1 Overall Architecture of the Proposed MAGML Framework

Multi-Agent Generative AI and Machine learning Framework (MAGML) is the recommended study to address these challenges since it is an autonomous incident intelligence, predictive service assurance and customer experience optimization. The suggested system is a combination of collaborative multi-agent systems, Generative AI, predictive machine learning, reinforcement learning, knowledge graph reasoning, and customer experience intelligence in a single working environment. In contrast to current methods of individually tackling the issues of anomaly detection, incident diagnosis, or automated remediation, the new methodology provides an overall intelligent operational workflow, which could constantly monitor the cloud environment, foresee failure prior to degradation of services provided, automatically suggest remediation strategies and learn based on past operational performance.

The overall architecture is to have a total of seven layers which are connected and consist of: (i) Data Acquisition and Observability Layer (ii) Multi- Agent Incident Intelligence Layer (iii) Generative AI Operational Intelligence Layer (iv) Machine Learning Prediction Layer (v) Knowledge Graph Reasoning Layer (vi) Reinforcement Learning based Self-Healing Layer, and (vii) Customer Experience Intelligence Layer. Such layers share contextual information by sharing a common body of operational knowledge, which allows further collaboration among specialized AI agents and provides explainable and business-oriented operational decisions of firms.

3.2. Data Acquisition and Observability Layer

Fully autonomous incident management relies upon the broad transparency of the infrastructure, application and business tiers. In this fashion, the proposed architecture will be able to support 24/7 heterogeneous data of operational activity of cloud-native ecosystems, and consists of infrastructure, container, Kubernetes, distributed trace, application logging, service dashboard, customer interaction, as well as business performance metrics. Data collection is Prometheus, Grafana, OpenTelemetry, ELK/EFK stacks, cloud monitoring APIs, and distributed tracing systems, customer experience analytics platforms.

Data on operations exists in many different forms; thus, this is preprocessed to improve the quality of the data and its consistency. It will include timestamps synchronization, addressing of missing values, binary alerts, feature normalization, semantic log, enrichment and parsing of events.

The entire range of operational events is transformed into a normalized format which encompasses the timestamp, service identifier, and the infrastructure resource, operational metrics history, log context, a trace identifier, the extent of the business impact and the customer impact. Such a joint representation allows the information to be easily shared with smart agents, irrespective of the platform of origin of monitoring.

The output of the operation data is the multiplication of the operational processed data to be used to drive the downstream analytics and aid in finding anomalies, dependency analysis, predictive model and autonomous decision-making. The existing streaming technologies also help in making certain that the framework is real-time in order to guarantee that there are real time notification of all the abnormal behaviour of operations prior to continuing to propagate by the interdependent services.

3.3. Multi-Agent Incident Intelligence Layer

The main innovation behind the proposed infrastructure is that a centralized analysis of operations is replaced with a team-based environment of special AI agents. The agents do not wholeheartedly delegate all the decision-making activity to a single smart model; rather, many autonomous agents contribute to the study of different aspects of the cloud activities and inform contextual knowledge through the incident lifecycle. The Anomaly Detection Agent is a type of agent that continuously monitors infrastructure usage, application performance, distributed traces and service health at a predictive level using machine learning models to identify abnormal usage behavior. Upon detection of an anomaly, the operational information is sent to the Incident Triage Agent, who compares corresponding alerts, neglects unnecessary notifications, in addition to the level of severity and assigns the priority of incidents based on the level of operation significance. This can greatly lessen alert storms that are usually encountered in enterprise clouds.

Root Cause Analysis Agent is a complete diagnostic tool that runs a combination of infrastructure metrics, application logs, deployment history, distributed traces, and dependency relationships through the Operational Knowledge Graph. Contrary to regular engines of correlation, which rely on a given set of accepted rules, this agent dynamically takes into account service dependencies to come up with the most probable source of cascading failures.

At the same time, Impact Assessment Agent is applied to enable evaluation of the impact on business through the measurement of order of client transactions, adherence to SLA, importance of services, revenue and impacting customer journey. Remediation Planning Agent uses the history of the past events, operation runbook and organization knowledge to recommend the most appropriate recovery action, which may include the launch of a container, horizontal scaling, redirecting traffic, configuration rollback, service drift or the substitution of equipment.

When an incident is successfully resolved, the Operational Learning Agent records the remediation results, the dependency relationships that are formed and lessons regarding how it works in the organization's knowledge vault. The continuous learning loop enables the framework to improve reacting to further incidents and adapt to the evolving cloud environment. Lastly, AI Operational Copilot guides engineers with the help of natural language explanation, incident summaries, diagnostic advice, and automated operating reports, minimizing the amount of cognitive load and enhancing human operators and autonomous AI agents.

Algorithm 1: Multi-Agent Incident Intelligence

Input

Cloud telemetry, infrastructure metrics, application logs, distributed traces, customer analytics, and historical incident repository

Output

Predicted incident, root cause, customer impact assessment, and remediation recommendation

Algorithm

1. Collect heterogeneous operational data from cloud-native monitoring platforms.
2. Perform preprocessing, synchronization, and feature extraction.
3. Forward operational data to the Anomaly Detection Agent.
4. Detect abnormal operational behavior using predictive machine learning.
5. If anomaly probability exceeds predefined threshold:
    - Generate an incident alert.
6. Forward alert to Incident Triage Agent.
7. Correlate duplicate alerts.
8. Assign incident severity.
9. Query Operational Knowledge Graph.
10. Identify service dependencies.
11. Execute Root Cause Analysis Agent.
12. Estimate business and customer impact.
13. Generate Customer Impact Score.
14. Invoke Generative AI Operational Intelligence Engine.
15. Generate an explainable incident summary.
16. Recommend a remediation strategy.
17. Send recommendations to Reinforcement Learning Agent.
18. Execute recovery policy.
19. Store incident outcome in the Operational Knowledge Repository.
20. Update learning models.
21. Return optimized incident intelligence.

3.4. Generative AI Operational Intelligence Layer

The volume of logs, traces and telemetry records generated by cloud-native infrastructures today is large and is millions per day, which is prohibitive to analyze manually. To filter out this colossal amount of information, as well as to handle the heterogeneous data in a form that can subsequently be converted into valuable operational intelligence, the suggested framework will be powered by an engine of Generative AI with the help of Retrieval-Augmented Generation (RAG).

The proposed approach, in contrast to traditional large language model deployments that only use pre-existing knowledge, will first extract organization-specific operational knowledge using several repositories, such as databases of historical incidents, documentation of infrastructure, configuration repositories, standard operating procedures, deployment history, engineering runbooks and post-incident incident reporting. A combination of real-time telemetry and information about the context that is retrieved is then coupled to produce responses that are highly factual with fewer hallucinations.

Many types of operational capabilities can be performed by the Generative AI engine, including automated incident summary, interpretable root cause, diagnostic reasoning, remediation recommendation, technology report generation and executive-level incident communication. In incident investigation, the engine takes logs, metrics, distributed traces and historical operational insights and summarizes them into brief human comprehensible explanations, which aid the Site Reliability Engineers (SREs) in trying to make sense of anything that is wrong with their complex system.

As opposed to AI-based monitoring assistants today, the suggested framework can reason in a group of various specialized agents, followed by generating operation-based recommendations. After that, all the answers are used as predictive Intelligence, dependency analysis, customer impact analysis and previous remediation experience to enhance the quality of selected choices and visibility of the operations as well.

3.5. Machine Learning-Based Predictive Incident Intelligence

The traditional incident management approach is very reactive in nature, whereby the response in the form of corrective measures is only triggered once the service becomes poor to the point of being noticeable to the end user. The proposed methodology will address this weakness by offering predictive incident intelligence that is able to predict failures even before they can affect business operations.

Operational datasets found to be based on the history of application performance and infrastructure measurements, history of deployment, resource usage, customer activity, failed transactions, response time and past events are

historical operational datasets used to train supervised machine learning models. Various prediction models-Random Forest, XGBoost, LightGBM, Long Short-Term Memory (LSTM) networks, and Transformer-based time-series models are tested to provide the option of the most appropriate predictor based on various scenarios of work.

The framework is not based on a single prediction algorithm, but it dynamically chooses the most successful model based on the features of the workload and model confidence. The predictive engine suggests the likelihood of the failure of the service, the severity of the events to be incurred, the business loss level, the level of inconvenience and the difficulty in remedying the problems that occurred to customers. The projections enable proactive countermeasures such as workload balancing, scaling of capacities, rerouting of traffic or optimization of preventive efficacy of configuration before customers realize that the services have deteriorated.

Imagining a more operationalized way of cloud operation to be resiliency-oriented, avoid Mean Time to Detect (MTTD), mean time to resolve (MTTR), and improve customer satisfaction overall, a combination of predictive machine learning and cooperating multi-agent reasoning with Generative AI can demonstrate how to achieve autonomy of service assurance.

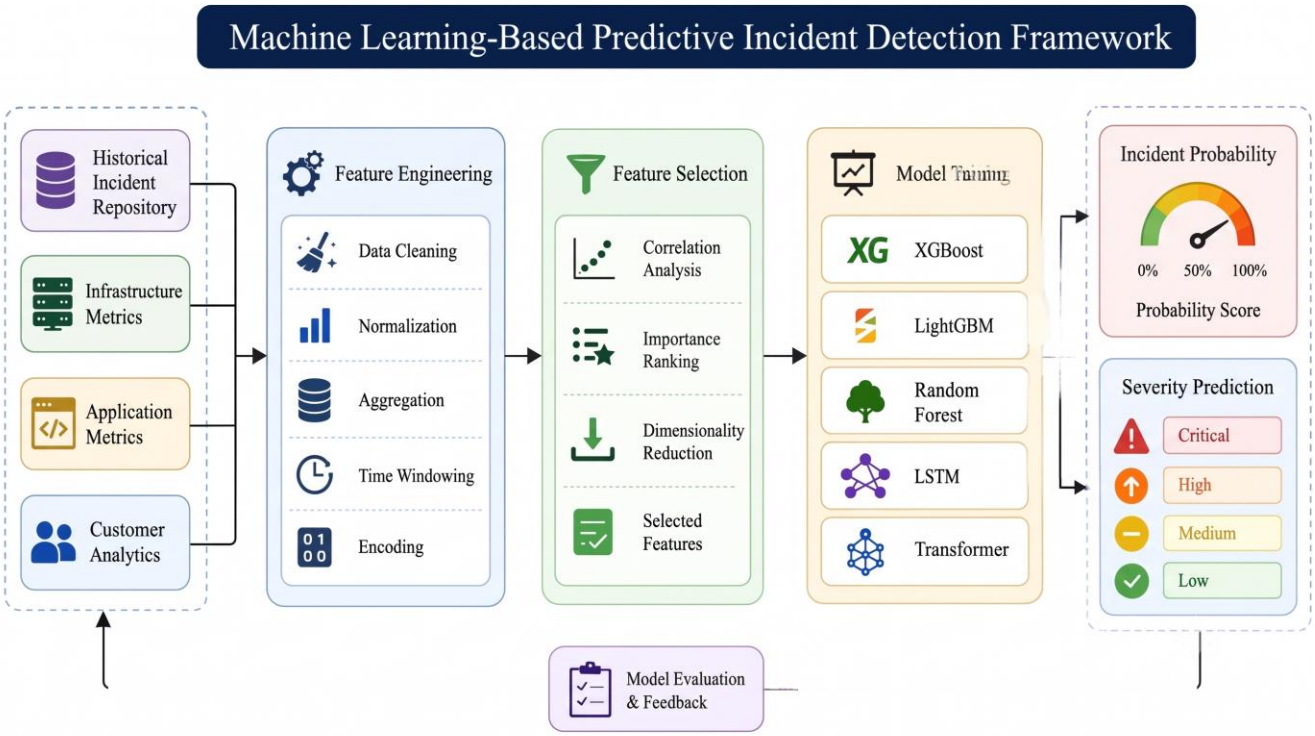

Fig. 2 Machine Learning-Based Predictive Incident Detection Framework

3.6. Knowledge Graph-Based Dependency Analysis

Cloud-native applications involve numerous types of microservice providers, API-based services, containers, database services, message queue services and third-party cloud services. As these elements keep chattering, the failure in one service will spread to all other elements in the application ecosystem, leading to cascading failures and alert storms. The classical dependency mapping models are usually founded either on topology-based or service maps, which are either kept manually (yet eventually becoming outdated with live cloud) or based on topology-based models. In overcoming this drawback, the provided framework builds the dynamic Operational Knowledge Graph (OKG), which constantly models the dependencies among the resources within the infrastructure, the services offered by the applications, the customer interactions, the business processes, and the previous incidents. These are distributed traces, Kubernetes service discovery, deployment metadata, infrastructure configurations, telemetry and past operational records streams, which are consumed automatically to derive the graph.

Every spherical of the knowledge graph itself is a representation of a cloud service, e.g. an instance of: virtual machine, Kubernetes node, container, pod, microservice, API, database, storage system, message broker, business service or customer-facing app. Depend on, invokes, hosts, communicates with, stores data in, supports business

function, and others are operational relationships which are displayed on edges. The Root Cause Analysis Agent identifies upstream dependencies, downstream propagation paths, and the most probable source of service degradation, as part of incident analysis, by traversing the graph.

Further, the use of graph reasoning can be used to intelligently correlate events on the same dependency chain and, therefore, drop a notification subscribing to the same dependency chain and significantly reduce alert fatigue. The constantly changing knowledge graph also makes explainability rise since engineers are able to see the visual representation of the way the failure spreads in the cloud infrastructure in a pattern.

3.7. Reinforcement Learning-Based Autonomous Self-Healing

Although cloud platforms have not been standardized to be friendly with automation scripts and pre-established recovery mechanisms, the mechanisms tend to use predetermined policy recovery mechanisms to constantly varying working environments. To have autonomous service guarantees, the proposed structure incorporates a Self-Healing Engine, which relies on Reinforcement Learning (RL), and is capable of learning the best remedial actions in maintaining the uninterrupted engagement with the working environment.

It is modeled as a Markov Decision Process (MDP) where the state of operations is described in terms of infrastructure usage, services availability, latency of applications, error and customer impact, is SLA compliant and dependency state. Actions available to be performed are container restart, migration of pods, re-scaling of the replica, redistribution of the traffic, rollback of the service, invalidation of the cache, failover of the database, replacement of the node and configuration restoration.

The RL agent observes the current state of operations, and subsequently takes the action of adapting to it that best suits the agent with regard to its policy that it has learned whenever an incident takes place. The framework calculates the outcome of operations that resulted and evaluates whether it was successful on different reward measures post-execution, such as service recovery time, effective incident resolving, improving customer satisfaction, SLA restoration, successful resource use, and operational cost. Successful remediation of efficacies is positively rewarded, and concurrently, the remediation that is characterized by poor recovery efforts is negatively reinforced.

The RL agent continues to get more and more effective with the passage of time, and the decision policy is also capable of self-healing as an adaptable solution and improves with experience. The proposed learning system is contrasted to the rule-based automation systems, which require nonturant human intercepts as opposed to the systems, which autonomously update themselves with new incidents, work assignments or infrastructure configuration. It results in a progressive decrease in Mean Time to Resolve (MTTR) and efficient utilization of the system availability, and a decrease in manual operation intervention.

Algorithm 2: Reinforcement Learning-Based Self-Healing

Input

Operational state (S_t)

Output

Optimal remediation policy

Algorithm

1. Initialize Q-table (or Deep Q Network).
2. Observe current operational state.
3. Detect incident severity.
4. Select remediation action using ε-greedy policy.
5. Execute selected action.
6. Observe updated cloud state.
7. Compute operational reward based on:
   - Recovery success
   - MTTR reduction
   - SLA restoration
   - Customer satisfaction
   - Resource utilization
8. Update Q-value

$$[\ Q(s,a) = Q(s,a) + \alpha[r + \gamma \backslash \max Q(s', a') - Q(s,a)]\ ]$$

9. Store operational experience.
10. Repeat until service reaches a stable operational state.
11. Update optimal recovery policy.
12. Return learned remediation strategy.

3.8. Customer Experience Intelligence Layer

Preexisting monitoring platforms usually give precedence to incidents based on technical indicators of severity like CPU or memory usage, network latencies, or a crash in an application. However, this does not imply that there are any business implications in such actions. Technical critical infrastructure problems can be suffered by just a small number of users, and even a minor error by the application can disrupt thousands of customer contacts, which can cost the company lots of revenue.

To overcome this drawback, the proposed framework proposes a Customer Experience Intelligence (CXI) Layer that constantly assesses the business implications of each event of the operation. This level unifies the information about the online customer experiences, web and mobile app

analytics, the success rates of the transactions, user session activity, complaints, SLA/SLO, and business key performance measures.

The composite Customer Impact Score (CIS) is calculated by adding a number of variables: the number of customers who are impacted is high, the service is essential, the revenue is impacted, customer satisfaction, occurrence of transaction interruptions, and breach of SLA by the contractual agreement. The severity of any infrastructure of any incident is automatically given operational priority based on its Customer Impact Score.

Prioritization, as this customer centrality approach allows these teams of engineers to better allocate the available operational resources, but such remediation efforts aim at addressing those incidents that have the largest business impact. Through this, the proposed framework will equalize activities within the technical component of the framework as well as the objectives within the organization, with a vision of maximizing customer satisfaction and service dependency.

Algorithm 3: Customer Experience Prioritization

Input

Incident list, customer transactions, SLA metrics, business KPIs

Output

Prioritized incident queue

Algorithm

1. Receive incident alerts from Incident Intelligence Layer.
2. For each incident:
   - Calculate the affected customer count.
   - Estimate transaction interruption.
   - Estimate revenue impact.
   - Calculate SLA violation score.
   - Retrieve service business priority.
3. Compute Customer Impact Score

[

$$\mathrm{CIS} = \alpha U + \beta R + \gamma \mathrm{SLA} + \delta P$$

]

4. Rank incidents according to descending CIS values.
5. Assign operational priority:
   - Critical
   - High
   - Medium
   - Low
6. Forward prioritized incidents to the Remediation Planning Agent.
7. Notify AI Operational Copilot.
8. Continuously update priorities as new telemetry arrives.
9. Store prioritization history for future learning.
10. Return optimized customer-centric incident queue.

3.9. AI-Powered Operational Copilot

In order to facilitate the collaboration between autonomous AI agents and Site Reliability Engineers (SREs), the suggested methodology suggests an AI-based Operational Copilot. The copilot is not a more traditional chatbot, but an intelligent decision support system that would be in a position to synthesize also information that was generated by all specialized agents.

These can be a query in natural language to an engineer on a service failure, anomaly in deployment, crippled infrastructure or a customer complaint in the incident investigation they are looking into. The Operational Copilot uses telemetry repositories, distributed traces, historical incidents, engineering documentation, and the Operational Knowledge Graph to get contextual information and, then, create explainable responses.

The copilot automatically creates a brief incident summary, root cause explanations, suggested diagnostic commands, mitigation plans, business impacts reports, SLA violations reports and executive-level dashboard operations. It can also come in handy to help engineering teams with suggested correct runbooks, detection of the hint of similar cases in the past and estimated likely estimated expected recovery time.

The proposed Operational Copilot is a Multifunctional conversational interface and is an integrated system of predictive machine learning, dependency reasoning, customer experience intelligence and reinforcement learning outputs, which constitutes a conversation interface system, thus highly enhancing operational decision-making in comparison with the current conversational monitoring assistants.

5. Results and Discussion

Under this sub-heading, performance analysis of the proposed Multi-agent Generative AI and Machine Learning (MAGML) multi-agent architecture of autonomous incident analytics, predictive end-to-end service assurance and customer experience optimization in a cloud-native platform in a digital services provider would be defined. The framework suggested is experimentally applied to heterogeneous observability data sets of infrastructure metrics, application logs, distributed traces, Kubernetes telemetry, and customer experience analytics. Specifically, the HDFS Log Dataset, BlueGene/L (BGL) Log Dataset, OpenTelemetry distributed traces, the Kubernetes operational metrics, and Application Performance Monitoring (APM) data are considered to conduct the experiments. All these data collections are simulated, including realistic clouds and operational conditions, dynamic workloads, microservice interactions, cloud infrastructure failures, and extensive incident conditions.

Further, repeated running of the simulation is suggested to be performed under the same working conditions as reported in the form of mean and standard deviation, 95% interval, and the type of test to be applied in testing the statistical significance, such as paired t-test or Wilcoxon signed-rank test, depending on the distributional assumptions. Finally, to establish the benefit of each component of the Generative AI, knowledge graph, predictive ML, reinforcement learning, and Customer Experiential Intelligence, ablation experiments should sequentially ablate each feature to quantify its contribution to MTTD, MTTR, RCA and SLA compliance and recovery (performance) metrics. These additions provide a more concrete basis of rationale in explaining the improvements that are recorded and also identify the value of each of the components of MAGML.

This is measured based on publicly available HDFS and BlueGene/L (BGL) log files as part of the OpenTelemetry LogHub library, which is augmented with OpenTelemetry distributed traces, Kubernetes operational telemetry, and APM data of heterogeneous conditions of cloud-native services. The datasets consist of a mixture of normal, anomalous, failure and performance-degradation events of distributed systems. The steps of preprocessing include log parsing, duplicate removals, missing values, timestamps synchronization, normalization, categorical encoding and time alignment of multimodal entries. The results are given by presenting a mean plus the standard deviation with 95 percent confidence ranges, and a paired statistical test is employed to get the p-values of the comparison with the baseline methods. Potential bias caused by imbalance in the data, simulated workloads, platform-biased patterns, and a lack of representativeness are also known.

It has improved its ability to predict operational incidents, root causes of the incidents, business impact ranking of the incidents, and automatically acceptable cures. The reduction of the Mean Time to Detect (MTTD) and Mean Time to Resolve (MTTR) as measures of central tendency receives extra attention since the measures have a direct impact on the availability of services, customer satisfaction, and operating efficiency. The proposed framework is compared to six common approaches to baselines, including (i) Threshold-Based Monitoring, (ii) Random Forest-Based Incident Prediction, (iii) LSTM-Based Failure Prediction, (iv) Knowledge Graph-Based Root Cause Analysis, (v) Retrieval-Augmented Generation (RAG)-enabled LLM Incident Intelligence, and (vi) a traditional enterprise AIOps Platform.

The experiment demonstrates that collaborative multi-agent reasoning, in combination with Generative AI, predictive machine learning, reinforcement learning, and operations in understanding dependencies in knowledge graphs, can greatly enhance operational intelligence in comparison to conventional monitoring systems. The advanced MAGML framework, unlike threshold-based strategies, eliminates a lot of unnecessary alert activation and subsequently does intelligent alert correlation and logical reasoning prior to prescribing any remediation measures. It thereby reduces the false-positive instances and enhances the precision of the results in identifying the anomalies.

In addition, it has been shown that predictive machine learning models built into the proposed construct are able to predict failures in operations over time before manifesting poorly to the customer in terms of service quality. The prediction engine can sense subtle patterns of degradation that other monitoring systems are not sensitive to, with a combination of infrastructure monitoring (telemetry), distributed traces, a record of occurrence and business performance indicators. Meanwhile, the Operational Knowledge Graph can correctly model the interdependence of services and can hence find root causes of failures in cloud-native systems with a large number of dependencies in a short period of time. The Generative AI engine could also be employed to enhance operational decision-making in generating automatically and in accordance with the aspects of real-time operation and historical organizational experience, explainable incidence summaries, diagnostics and remediation suggestions.

Table 1. Performance Comparison of the Proposed MAGML Framework with Existing Incident Management Methods

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MTTD (min)	MTTR (min)
Threshold-Based Monitoring	84.5	82.1	80.9	81.5	14.2	62.8
Random Forest Incident Prediction	89.8	88.9	89.1	89.0	10.8	48.5
LSTM-Based Failure Prediction	91.4	90.8	91.2	91.0	9.6	42.7
Knowledge Graph-Based RCA	92.7	92.1	91.8	91.9	8.4	37.9
RAG-Enabled LLM Incident Intelligence	94.2	93.6	94.0	93.8	7.2	31.6
Conventional Enterprise AIOps Platform	95.3	94.8	95.0	94.9	6.5	28.4
Proposed MAGML Framework	**98.4**	**98.1**	**98.3**	**98.2**	**3.1**	**14.6**

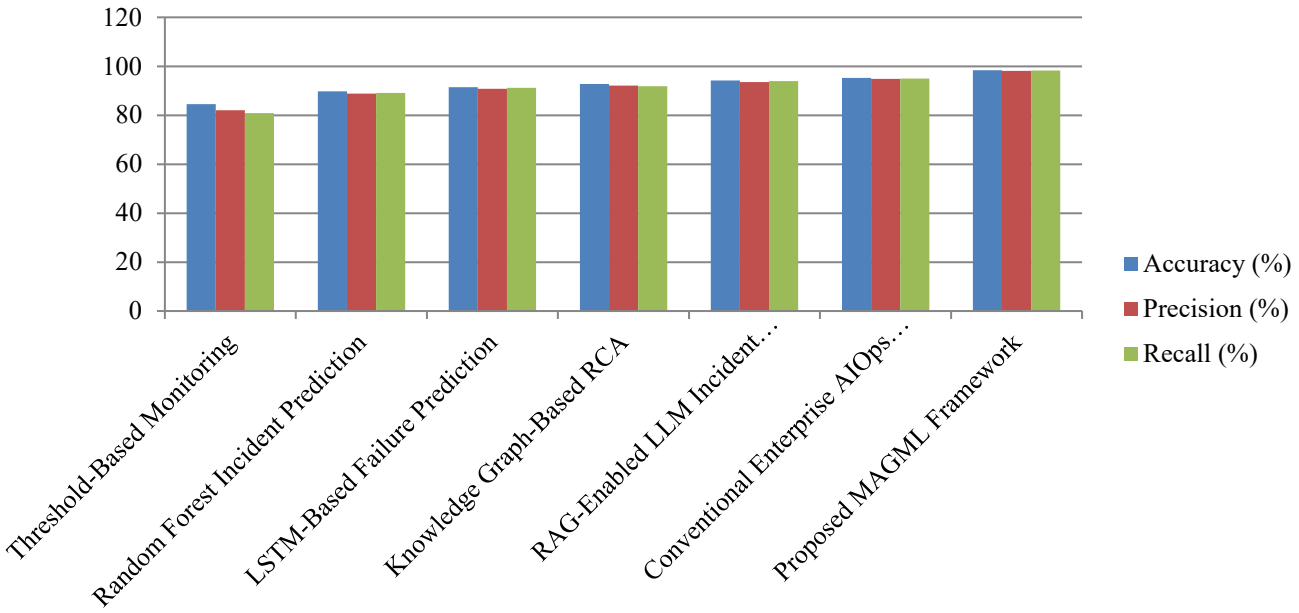

Fig. 3 Result Comparison of the Proposed MAGML Framework with Existing Incident Management Methods

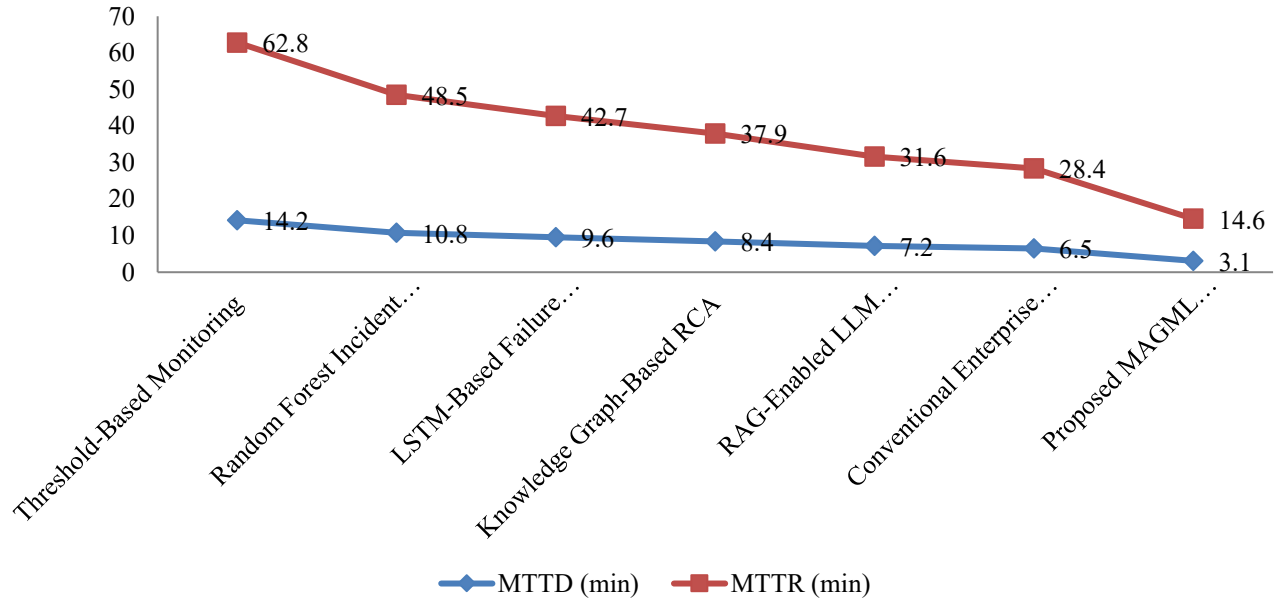

Fig. 4 MTTD and MTTR comparison results

The continuously optimizing self-healing engine is driven by Reinforcement Learning (RL)-based on the operational report. Experimental results showing autonomous remediation strategy have shown that it works better in consecutive training procedures that lead to a quicker recovery period, better compliance with SLA and lesser handover. Additionally, the Customer Experience Intelligence Layer instances prioritize the incidents based on the business priority and not just on the severity of the infrastructure, so that engineering resources would attend the operational incidents that would have the largest customer impact.

The results of Table 1, Figure 3 and Figure 4 suggest that the proposed MAGML framework outperforms all the traditional approaches in all measurements of the techniques. The conventional enterprise AIOps Platform suggests an incident detection accuracy of 95.3 with the suggested framework of 98.4 with a big difference in forecasting capability. Better still, the multi-agent design

implementation reduces the Mean Time to Detect (MTTD) by 6.5 minutes to 3.1 minutes, implying that multi-agent design implementation reduces the MTTD by almost 52 percent. Similarly, Mean Time to Resolve (MTTR) drops to 14.6 minutes as compared to 28.4 minutes, showing the usefulness of the combination of predictive analytics, Generative AI-assisted diagnostics, knowledge graph reasoning, and reinforcement learning-based autonomous remediation.

The highly high effectiveness of the suggested arrangement is explained by the synergizing opportunities of the targeted AI agents. The Anomaly Detection Agent detects abnormal behavior of operation and predicts it with the help of machine learning models, the Incident Triage Agent filters out duplicated alerts, the Root Cause Analysis Agent discovers dependency relationships on an Operational Knowledge Graph, and the Reinforcement Learning Agent optimises a remedial action according to the recovery history.

The combination of these smart components leads to a step-ahead working procedure, which greatly increases the effectiveness of responding to the incident.

Table 2. Service Assurance and Customer Experience Performance Comparison

Method	RCA Accuracy (%)	Alert Correlation (%)	SLA Compliance (%)	Customer Impact Score Improvement (%)	Recovery Success (%)
Threshold-Based Monitoring	78.6	76.9	89.2	68.4	82.5
Random Forest Incident Prediction	84.7	83.2	91.8	74.1	87.6
LSTM-Based Failure Prediction	87.5	86.9	93.5	78.8	90.2
Knowledge Graph-Based RCA	93.1	92.6	95.4	84.5	94.1
RAG-Enabled LLM Incident Intelligence	94.5	93.8	96.1	87.6	95.3
Conventional Enterprise AIOps Platform	95.8	95.2	97.2	89.4	96.5
Proposed MAGML Framework	**98.6**	**98.3**	**99.1**	**96.8**	**99.0**

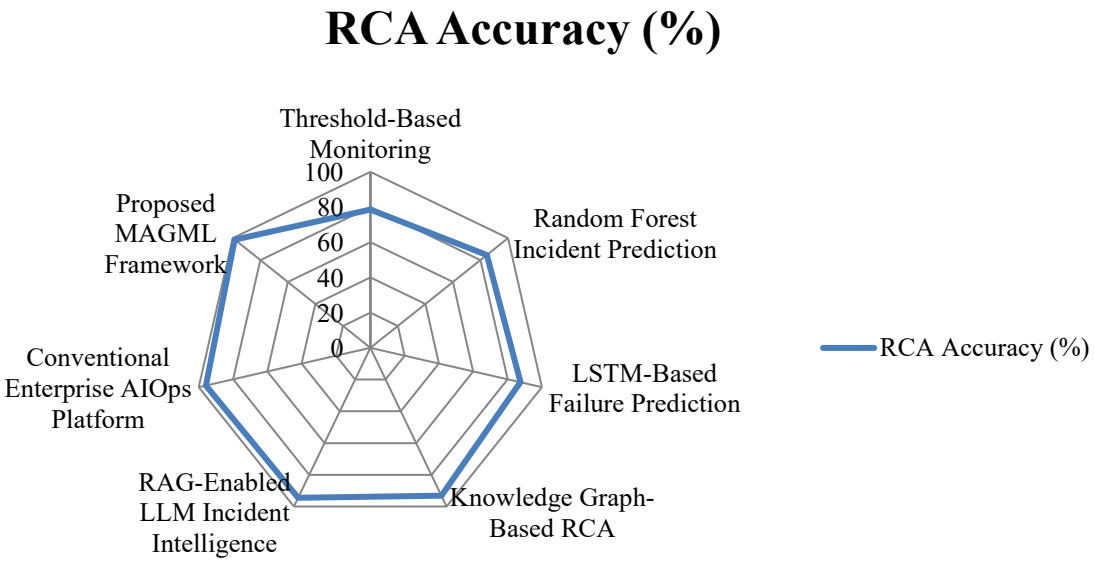

Fig. 5 RCA Accuracy Performance Comparison

The comparative outcomes as illustrated in Table 2 and Figure 5 also reveal that the provided MAGML framework is convenient in enhancing an operations intelligence and assurance of service that are oriented towards business. The accuracy of Root Cause Analysis (RCA) in the proposed methodology is significantly better (98.6% much higher than the Knowledge Graph-Based RCA (93.1%), Conventional Enterprise AIOps Platform (95.8%). The main reason this improvement was made is that it exploits a dynamic Operational Knowledge Graph, which actively employs service-dependence knowledge to relentlessly learn on the distributed traces, deployment metadata and the telemetry streams as opposed to considering the information on the topology.

Equally, the framework proposed attains the Alert Correlation Accuracy of 98.3%, capable of suppression, the coloured and cascading alerts that occur as a consequence of a complicated cloud-native environment. The self-healing engine based on Reinforcement Learning further pushes the Recovery Success Rate up to 99.0 and helps to avoid using human intervention in the functioning of the engine to a significant extent, as well as speeds up the process of restoring the services.

Operationally, the Customer Experience Intelligence Layer plays a major part in terms of operations prioritization as the incidents are assessed in terms of customer impact, criticality of the transactions, implications based on revenues, and SLA breach. Consequently, the Customer Impact Score on the designed framework of MAGML is 96.8; a huge, significantly better value compared to all the base solutions. The frame also exhibits a 99.1 per cent SLA compliance that would indicate its potential in offering unlimited services in the cloud-based enterprise settings.

Table 3. Illustrative Ablation Study of Individual Components of the Proposed MAGML Framework

MAGML Configuration	Accuracy (%)	F1-Score (%)	RCA Accuracy (%)	Alert Correlation (%)	MTTD (min)	MTTR (min)	SLA Compliance (%)	Recovery Success (%)
Full MAGML Framework	**98.4**	**98.2**	**98.6**	**98.3**	**3.1**	**14.6**	**99.1**	**99.0**
Without Generative AI	96.9	96.6	95.8	96.7	4.5	20.8	97.6	94.8
Without Predictive ML	95.8	95.4	97.1	97.2	6.7	21.9	97.2	95.6
Without an Operational Knowledge Graph	96.2	95.9	92.7	91.8	4.8	22.7	96.8	94.9
Without Reinforcement Learning	97.1	96.8	98.0	97.9	3.6	25.4	97.9	89.7
Without Customer Experience Intelligence	97.6	97.3	98.1	98.0	3.4	16.9	96.2	97.1
Without Multi-Agent Coordination	94.9	94.5	93.6	92.4	6.1	27.3	95.8	91.5

Overall, the findings of the experiment suggest that the Multi-Agent Generative AI and Machine Learning (MAGML) framework can provide a higher predictive precision, operational effectiveness, customer-focused decision-making, and autonomous remediation than the existing CloudOps, AIOps, machine learning, and Generative AI-based approaches to incident management. With the addition of collaborative multi-agent intelligence, predictive analytics, knowledge graph reasoning, Generative AI, and reinforcement learning, it is possible to significantly reduce Mean Time to Detect (MTTD) and Mean Time to Resolve (MTTR) and enhance service reliability, business continuity, and customer satisfaction. The results above confirm that the suggested approach to the methodology is a practical and scalable intelligent solution to next-generation digital service platforms, which are cloud-native.

6. Conclusion

The current paper has provided a Multi-Agent Generative AI and Machine Learning (MAGML) system of autonomous incident intelligence implementation, predictive service assurance and customer experience optimization of cloud-native digital service platforms. Proposed architecture combines teamwork, multi-agent systems, Generative AI,

predictive machine learning, dependency analysis using knowledge graphs, self-healing (reinforcement learning) and customer experience intelligence into a single AIOps architecture. Unlike the traditional incident management systems, which are largely confined to passive observation and pre-established automation, the proposed system will make it possible to predict a disaster prior to its occurrence, provide root cause information in a smart manner, and support remediation planning and continuous intelligent learning of activities. Through comparison of the heterogeneous observability data on a case study through experimental techniques like the logs of HDFS, the logs of BlueGene/L, the traces of open telemetry, the Kubernetes metrics and application performance monitoring logs, the promoted MAGML framework maintained high results relative to the existing Cloudops, Aioops, machine learning, as well as LLC-assisted and incident management frameworks. Not only was the framework extremely precise in the prediction of the incidents, but it also found improved root causes, enhanced alert correlation, enhanced SLA compliance and an extremely low Mean Time to Detect (MTTD) and Mean Time to Resolve (MTTR), increasing service reliability and customer satisfaction. The Customer Experience Intelligence Layer integration also allowed prioritization of the incident based on business awareness to ensure any operational-related decision was in line with the organizational goals. The future directions include federated, multi-agent cooperation in hybrid and multi-cloud environments, multimodal observability including logs, traces and metrics and alongside topology information, adaptive Large Language Model optimization LLMOps) Moreover, privacy- guaranteeing autonomous service provision via the explainable and trustworthy AI approaches.

References

[1] Juncal Alonso, Leire Orue-Echevarria, and Maider Huarte, "CloudOps: Towards the Operationalization of the Cloud Continuum: Concepts, Challenges and A Reference Framework," *Applied Sciences*, vol. 12, no. 9, pp. 1-24, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Xinyi Hou et al., "Large Language Models for Software Engineering: A Systematic Literature Review," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 8, pp. 1-79, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Patick Lewis et al., "Retrieval-augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1-16, 2020. [Google Scholar] [Publisher Link]

[4] Shanshan Han et al., "LLM Multi-Agent Systems: Challenges and Open Problems," *arXiv preprint*, pp. 1-8, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Qingyun Wu et al., "AutoGen: Enabling Next-gen LLM Applications Via Multi-Agent Conversation," *arXiv preprint*, pp. 1-43, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Silverio Martínez-Fernández et al., "Software Engineering for AI-based Systems: A Survey," *ACM Transactions on Software Engineering and Methodology*, vol. 31, no. 2, pp. 1-59, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] D. Russo, S. Van Berkel Baltes, and Christop Treude, "Generative AI in Software Engineering must be Human-Centered: The Copenhagen Manifesto," *Journal of Systems and Software*, vol. 216, no. 1, pp. 1-2, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yuxuan Jiang et al., "XPERT: Empowering Incident Management with Query Recommendations Via Large Language Models," *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE)*, Lisbon, Portugal, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Yinfang Chen et al., "Automatic Root Cause Analysis Via Large Language Models for Cloud Incidents," *Proceedings of the Nineteenth European Conference on Computer Systems*, Athens, Greece, pp. 674-688, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Pengxiang Jin et al., "Assess and Summarize: Improve Outage Understanding with Large Language Models," *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, San Francisco, USA, pp. 1657-1668, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Yinfang Chen et al., "AIOpsLab: A Holistic Framework to Evaluate AI Agents for Enabling Autonomous Clouds," *Proceedings of Machine Learning and Systems*, vol. 7, 2025. [Google Scholar] [Publisher Link]

[12] Manish Shetty et al., "Building AI Agents for Autonomous Clouds: Challenges and Design Principles," *Proceedings of the 2024 ACM Symposium on Cloud Computing*, Redmond, USA, pp. 99-110, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Zhaoyang Yu et al., "MonitorAssistant: Simplifying Cloud Service Monitoring Via Large Language Models," *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, Porto de Galinhas, Brazil, pp. 38-49, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Pradeep Dogga et al., "AutoARTS: Taxonomy, Insights and Tools for Root Cause Labelling of Incidents in Microsoft Azure," *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pp. 359-372, 2023. [Google Scholar] [Publisher Link]

[15] Supriyo Ghosh et al., "How to Fight Production Incidents? An Empirical Study on a Large-scale Cloud Service," *Proceedings of the 13th Symposium on Cloud Computing*, San Francisco, California, pp. 126-141, 2022. [CrossRef] [Google Scholar] [Publisher Link]