NFVAgent: A Retrieval-Augmented LLM Agent for Resilient NFV Failure Recovery

1st Yeunjin Woo
Department of DMC Engineering
Sungkyunkwan University
Suwon, Republic of Korea
yeunjin.woo@g.skku.edu

2nd Honguk Woo
Department of Computer Science and Engineering
Sungkyunkwan University
Suwon, Republic of Korea
hwoo@skku.edu

***Abstract*—As modern networks continue to grow in scale, heterogeneity, and complexity, automation in Network Function Virtualization (NFV) management has become increasingly critical. Traditional NFV management systems, which rely on static policies and manual interventions, often fail to respond effectively to dynamic network conditions. This leads to delayed or suboptimal failure recovery. Moreover, the increasing diversity of NFV deployments, spanning heterogeneous configurations, tenants, and operational standards, restricts the flexibility and scalability of existing policy-based solutions. Considering these limitations, this work explores Large Language Model (LLM) agent approaches by harnessing the advanced reasoning capabilities of LLMs within NFV operational context. While LLM-based approaches are structurally promising, LLMs often lack explicit exposure to the diverse and domain-specific characteristics of NFV environments, resulting in limited generalization and reliability in real-world scenarios. We introduce NFVAgent, an LLM-driven NFV recovery framework built on Retrieval-Augmented Generation (RAG). It overcomes adaptation limitations by continuously updating its knowledge base through experiences gathered from both testbed and deployment environments. This evolving knowledge, validated across diverse NFV environments, is seamlessly integrated into the agent's decision-making loop, enabling it to generate appropriate recovery actions. The framework supports dynamic reasoning over up-to-date knowledge, allowing the agent not only to react to failures but to continually refine its understanding of NFV failure patterns and effective recovery strategies. We evaluate NFVAgent on an alarm dataset comprising over 500 failure events with diverse NFV environments, each configured based on industry standards (e.g., ONAP, ETSI, O-RAN). Experimental results show that NFVAgent achieves up to 99.8% recovery accuracy, outperforming existing policy-based methods by an average margin of 37.7% in multi-tenant environments. This highlights the practical viability and performance benefits of integrating LLM agents with retrieval mechanisms that leverage NFV-specific operational knowledge in real-world recovery tasks.**

***Index Terms*—network function virtualization (NFV), fault recovery, retrieval-augmented generation (RAG), large language models (LLMs), AIOps**

I. Introduction

Network Function Virtualization (NFV) significantly enhances scalability and resource efficiency in modern networks. However, this advancement also increases management complexity, as highlighted by ETSI [1], [2], making NFV fault management more demanding.

AI-driven automation is gaining traction across industries [3]–[5]. In particular, Large Language Models (LLMs) have been employed in cloud operations to analyze logs, detect anomalies, and generate automated recovery strategies [5]–[7]. Motivated by this, we investigate LLM agents for NFV fault recovery in complex, multi-tenant environments.

In NFV management, recovery has traditionally relied on policy-based approaches, where experts define failure scenarios and corresponding actions during system design. Such approaches have inherent limitations, especially in adapting to environmental changes, as static policies often fail to generalize to evolving conditions.

We introduce NFVAgent, an LLM-driven recovery framework designed to enhance adaptability in dynamic and heterogeneous NFV environments. Unlike rigid, predefined policies, NFVAgent analyzes failure events, retrieves domain-specific knowledge, and generates context-aware recovery actions, addressing the limitations of policy-based methods.

Although LLMs can perform semantic interpretation and action selection, their effectiveness in NFV remains constrained by limited exposure to operational data. While fine-tuning can embed domain-specific knowledge, it incurs high computational and maintenance costs. To address this, NFVAgent adopts a retrieval-augmented generation (RAG) mechanism that dynamically injects external knowledge during inference. The retrieval leverages a continuously evolving knowledge base spanning testbed and deployment environments.

NFVAgent comprises four modules: the Context Manager, Reasoner, Feasibility DB, and Site DB. Together, these enable adaptation across diverse NFV environments.

II. Related Work

LLMs have recently advanced AIOps tasks such as anomaly detection, RCA, and remediation, surpassing traditional ML approaches that require labeled data and feature engineering [5], [8], [9]. They can infer causal links and propose recovery without explicit rules [6], [10], [11], while also enabling tool-augmented reasoning and script generation [7], [12]. Applications span log-based RCA [13], [14], time-series anomaly detection [15], recovery agents [16], [17], multi-modal orchestration [18], agent collaboration [19], [20], and prompting-based fault inference [21].

978-3-903176-75-1 ©2025 IFIP

In NFV, fault management still relies on policy-based orchestration in ETSI MANO, ONAP, and O-RAN [22]–[24], which struggle with novel faults and heterogeneous deployments. AI-driven orchestration and intent research [25]–[27] mainly target service configuration or traffic analysis, with extensions to 5G intent [28], healthcare and industry [29], [30], private 5G [31], and multimodal automation [32]. LLMs for real-time NFV fault recovery remain unexplored; NFVAgent fills this gap with adaptive, context-aware recovery beyond static policies.

III. Our Approach

A. NFVAgent Framework

NFVAgent operates within an NFV management system, responding to failure events by autonomously planning recovery actions. It consists of four modules: the Context Manager, Reasoner, Feasibility DB, and Site DB. The Context Manager structures raw alarms into prompts, the Reasoner generates candidate actions, the Feasibility DB validates their executability across standards, and the Site DB ensures deployment-specific grounding. Together, they enable NFVAgent to remain robust in heterogeneous and evolving NFV environments.

B. Context Manager

The **Context Manager** retrieves metadata from the Site DB and validated patterns from the Feasibility DB, assembling them into structured prompts for the Reasoner.

The following is an example of such a prompt:

> **[Role declaration and action preference]**
> You are an NFV Management System. Generate recovery actions that can be executed automatically by the system in response to the failure event input. Respond with a feasible recovery action based on the retrieved documents, preferably one that can be executed automatically without human intervention. If the optimal action is explicitly marked as unavailable in the retrieved documents, choose an alternative action. If the retrieved documents do not explicitly mention feasibility, do not assume feasibility; prefer a safe fallback or a conservative probe validated by retrieved evidence.
> **[Failure event context and Site specific metadata]**
> Alarm : The CPU workload in the VNF is exceeding safe thresholds in Serval NFVO v3.0, Elephant ADF 24A, and OpenShift v11.1.
> **[Feasibility data]**
> Retrieved documents :
> { "Scale_Out": "valid", "Restart": "valid", "Migrate": "invalid (incompatible host resources)", "Throttle_Traffic": "valid", ... }
> **[Response format guide]**
> Provide your answer as a single, concise interface-level action name (e.g., Restart VNFC, Scale Out).

C. Reasoner

The **Reasoner**, powered by an LLM, generates recovery actions from the structured prompt. In our reference implementation, we used `gemini-1.5-flash` via API, but the backend model is swappable without altering the pipeline. To ensure safe execution, the Reasoner automatically retries inference if the response format deviates from the expected structure. It interprets the failure event in light of both feasibility constraints and site-specific metadata, preventing speculative or unsafe actions. The Reasoner benefits from the structured prompt design of the Context Manager, which encodes operational roles, failure descriptions, and feasibility boundaries. This integration ensures that the LLM outputs are not only semantically appropriate but also operationally valid.

D. Feasibility DB

The **Feasibility DB** encodes recovery feasibility as action–environment combinations validated in both testbed and deployment environments. For generalization, instance-specific identifiers (e.g., UUIDs, resource names) are abstracted into schema-level representations. These structured entries support consistent encoding and efficient similarity-based retrieval, ensuring that recovery knowledge can transfer across tenants and standards. It is continuously updated by the Context Manager, which records both successful and failed recovery attempts for future reference. By reflecting empirical outcomes, the Feasibility DB mitigates LLM hallucinations and enforces grounding in operational constraints. This evolving knowledge base is critical to sustaining high accuracy under diverse and dynamic NFV conditions.

E. Site DB

The **Site DB** maintains deployment-specific metadata, such as NF instance IDs, host resource availability, and current service states. It is incrementally updated when infrastructure changes occur, ensuring that recovery actions are grounded in live operational conditions. This enables deployment-aware decision-making without the overhead of continuous polling. The Site DB captures runtime configurations, platform versions, and topology information that vary across tenants. These details allow NFVAgent to specialize generalized feasibility patterns for concrete deployments. In combination with the Feasibility DB, it ensures that recommended recovery actions are both technically executable and contextually aligned with the active site.

F. NFV Simulator and Dataset

To enable reproducible evaluation, we implemented an **NFV simulator** that emulates diverse infrastructures and fault conditions. It supports (i) configurable interface capabilities, (ii) concurrent multi-tenant deployments, (iii) fault injection, and (iv) API-based validation of actions. The simulator mimics realistic orchestration while providing ground truth, allowing us to assess both executability and operator alignment.

For benchmarking, we curated an **alarm dataset** of over 500 events across heterogeneous NFV environments, organized into three subsets:

- **BaseSet** (100 canonical events) aligned with ETSI NFV categories such as compute, network, and storage.
- **TestSet** (500 variants) generated by varying configurations, parameters, and topology.

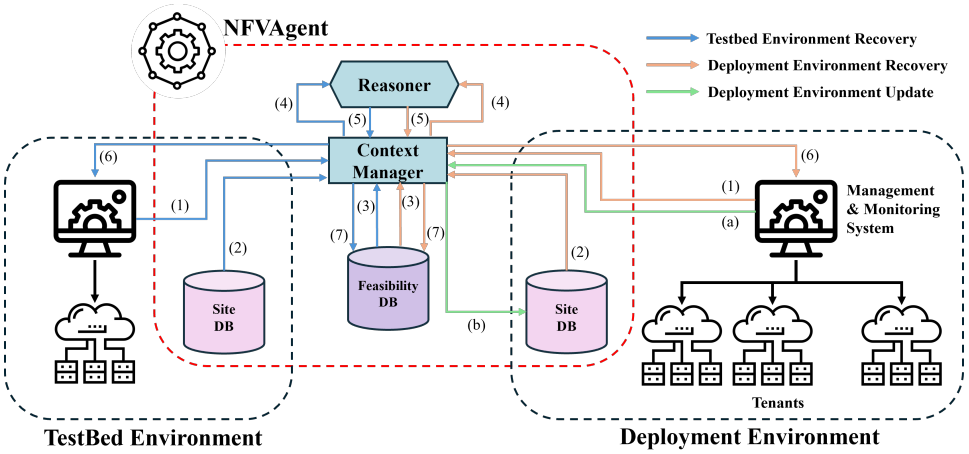

Fig. 1: NFVAgent Architecture: The recovery process operates consistently across testbed and deployment environments. (1) A failure event is received by the Context Manager, which (2) retrieves site data from the Site DB and (3) queries the Feasibility DB for validated operations. (4) These are combined into a structured prompt for the Reasoner, (5) which generates recovery actions. (6) The actions are executed by the NFV management system, and (7) the results are recorded in the Feasibility DB for continuous refinement. When a configuration change occurs, (a) the update is propagated to the Context Manager and (b) stored in the Site DB to keep metadata synchronized.

- **NewSet** (100 unseen events) representing novel alarms for testing generalization.

Example alarms include *"VNFC Memory Overload"*, *"NFVI Storage Performance Degradation (IOPS < threshold)"*, and *"Network Switch Link Failure"*, spanning ONAP, ETSI, and O-RAN to capture intra- and cross-standard variation.

IV. Evaluation

We evaluate NFVAgent to answer four research questions:

- RQ1: Does NFVAgent outperform policy-based methods?
- RQ2: How well does it adapt to heterogeneous multi-tenant environments?
- RQ3: Can it generalize to unseen failure events?
- RQ4: What is the impact of each architectural module?

A. Experiment Setup

1) NFV Environments: Single-tenant environments represent ONAP, ETSI, and O-RAN standards. Multi-tenant settings (3, 6, 9 tenants) emulate heterogeneous deployments with mixed compliance. These settings emulate the complexity of practical NFV deployments.

2) Baselines: We implement keyword- and alarm-code-based policy approaches aligned with NFV standards [33], [34]. Each baseline is optimized for its respective standard.

3) Evaluation Metrics: We use Accuracy (Acc) and Proactive Accuracy (PAcc), as defined in Fig. 2. Acc reflects the share of feasible and resolvable actions, while PAcc captures proactive or fallback actions judged acceptable.

4) Evaluation Scope: Simulations focus on semantic validity and contextual appropriateness of recovery actions.

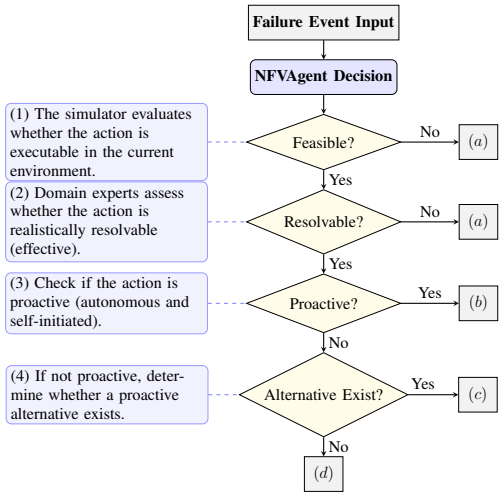

Fig. 2: Accuracy (Acc) and Proactive Accuracy (PAcc) computation. Acc = $\frac{(b+c+d)}{(a+b+c+d)}$, PAcc = $\frac{(b+d)}{(a+b+c+d)}$. (a) Invalid actions, (b) Valid proactive actions, (c) Valid non-proactive alternatives, (d) Justified fallback.

B. Performance in Single-Tenant Settings (RQ1)

As shown in Table I, NFVAgent consistently outperforms policy-based baselines across ONAP, ETSI, and O-RAN environments. It achieves Proactive Accuracy above 96%, while the best baselines remain below 84%. These results demonstrate NFVAgent's robustness in proactive recovery, unlike static policies defaulting to fallbacks.

C. Performance in Multi-Tenant Settings (RQ2)

Table II shows that baseline performance degrades sharply as the number of tenants increases. In contrast, NFVAgent

TABLE I: Performance in Single-Tenant Settings

Method	ONAP		ETSI		O-RAN	
	Acc(%)	PAcc(%)	Acc(%)	PAcc(%)	Acc(%)	PAcc(%)
Policy-Based						
Keyword (ONAP)	92.2	72.0	74.8	54.6	71.2	51.0
Keyword (ETSI)	92.2	66.4	92.2	66.4	69.2	43.4
Keyword (O-RAN)	92.2	72.0	76.2	62.2	94.0	80.0
AlarmCode (ONAP)	98.0	77.0	80.0	59.0	76.0	55.0
AlarmCode (ETSI)	98.0	71.0	98.0	71.0	74.0	47.0
AlarmCode (O-RAN)	98.0	77.0	81.0	65.0	100.0	84.0
NFVAgent	98.4	98.4	96.8	96.8	97.8	97.8

maintains over 95% Proactive Accuracy even in 9-tenant settings, outperforming baselines by up to 60 points. This highlights scalability and adaptability to heterogeneous deployments.

TABLE II: Performance in Multi-Tenant Settings

Method	3-Tenant		6-Tenant		9-Tenant	
	Acc(%)	PAcc(%)	Acc(%)	PAcc(%)	Acc(%)	PAcc(%)
Policy-Based						
Keyword (ONAP)	80.4	60.2	57.8	37.6	57.2	37.0
Keyword (ETSI)	83.6	57.8	61.8	36.0	60.2	34.4
Keyword (O-RAN)	88.4	74.4	55.4	41.4	51.2	37.2
AlarmCode (ONAP)	85.8	64.8	61.8	40.8	61.4	40.4
AlarmCode (ETSI)	89.2	62.2	66.2	39.2	64.6	37.6
AlarmCode (O-RAN)	94.0	78.0	59.8	43.8	55.2	39.2
NFVAgent	100.0	99.6	98.2	95.4	99.8	96.0

D. Generalization Performance (RQ3)

On NewSet (100 unseen events), NFVAgent achieves 99% Accuracy, while baselines drop significantly (Fig. 3), showing that retrieval-grounded reasoning effectively handles anomalies absent from testbed or baseline mappings.

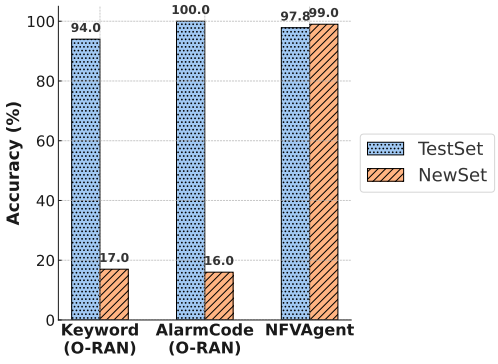

Fig. 3: Generalization Performance on NewSet (unseen events) in an O-RAN Single-Tenant Environment

Case Study. For the unseen event *"Dynamic resource scaling anomaly detected in auto-scaler"*:

- **NFVAgent:** Generated *"Invoke AutoScaler reset API"* by retrieving similar past cases from the Feasibility DB and verifying AutoScaler state via the Site DB, enabling successful resolution.
- **Policy-based:** Produced *"Trigger VM failover to backup host"* from a static keyword match, misidentifying the anomaly as host failure and causing ineffective recovery.

This case illustrates how NFVAgent leverages feasibility grounding and deployment metadata to generalize to novel failures, whereas static policies remain brittle.

E. Ablation Study (RQ4)

As shown in Table III, removing Feasibility DB drops Acc below 50%, removing Site DB reduces Acc to 75%, and removing Context Manager collapses performance (<20%). Structured context and feasibility grounding are essential.

TABLE III: Ablation Studies in Multi-Tenant Settings

Configuration	3-Tenant		6-Tenant		9-Tenant	
	Acc(%)	PAcc(%)	Acc(%)	PAcc(%)	Acc(%)	PAcc(%)
Ablated Components						
w/o Feasibility DB	49.0	49.0	45.2	45.2	48.2	48.2
w/o Site DB	80.4	78.8	73.8	72.8	75.0	73.2
w/o Context Manager	8.2	6.0	12.4	8.0	19.2	16.8
NFVAgent (Full)	100.0	99.6	98.2	95.4	99.8	96.0

We also test model substitution (Table IV): replacing the Reasoner's LLM with alternatives yields consistently >97% Acc.

TABLE IV: LLM Substitution (in 9-Tenant Setting)

LLM	Acc(%)	PAcc(%)
`claude-3-opus`	98.2	97.4
`o1-mini`	97.4	97.2
`gpt-4o-mini`	99.0	98.6
`gemini-1.5-flash`	99.8	96.0

V. Conclusion

We presented NFVAgent, an LLM-based NFV recovery framework with a deployment-aware RAG pipeline. It outperforms policy-based baselines across environments and generalizes to unseen failures. Future work includes multi-alarm correlation and tool-augmented LLM integration for greater autonomy.

Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-25442569, AI Star Fellowship Support Program, Sungkyunkwan Univ. and RS-2019-II190421, Artificial Intelligence Graduate School Program (Sungkyunkwan University)), IITP-ICT Creative Consilience Program grant funded by the Korea government (MSIT) (IITP-2025-RS-2020-II201821, 25%), and by Samsung Electronics Co., Ltd.

REFERENCES

[1] P. W. M. Chiosi, D. Clarke, and M. Fargano, "Network functions virtualisation: An introduction, benefits, enablers, challenges call for action," ETSI, White Paper 1, 2012.

[2] ETSI, "Nfv management and orchestration," ETSI, Tech. Rep. GS NFV-MAN 001 V1.1.1, 2014.

[3] S. Afrin, "Ai-enhanced robotic process automation," *IEEE Access*, vol. 12, pp. 1–1, 2024.

[4] W. Noonpakdee, "The adoption of artificial intelligence for financial investment service," in *Proc. 22nd Int. Conf. Adv. Commun. Technol. (ICACT)*, Phoenix Park, Korea, 2020, pp. 487–492.

[5] J. Zha, X. Shan, J. Lu, J. Zhu, and Z. Liu, "Leveraging large language models for efficient alert aggregation in aiops," *Electronics*, vol. 13, no. 22, p. 4425, Nov. 2024.

[6] T. Ahmed, S. Ghosh, C. Bansal, T. Zimmermann, X. Zhang, and S. Rajmohan, "Recommending root-cause and mitigation steps for cloud incidents using large language models," in *Proc. IEEE/ACM 45th Int. Conf. Software Engineering (ICSE)*, Melbourne, Australia, May 2023, pp. 1737–1749.

[7] Y. Jiang, C. Zhang, S. He, Z. Yang, M. Ma, S. Qin *et al.*, "Xpert: Empowering incident management with query recommendations via large language models," in *Proc. IEEE/ACM 46th Int. Conf. Software Engineering (ICSE)*, Apr. 2024, pp. 1–13, article 92.

[8] L. Zhang, T. Jia, M. Jia, Y. Wu, A. Liu, Y. Yang *et al.*, "A survey of aiops for failure management in the era of large language models," Jun. 2024, arXiv preprint arXiv:2406.11213.

[9] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi *et al.*, "Time-llm: Time series forecasting by reprogramming large language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[10] Z. Wang, Z. Liu, Y. Zhang, A. Zhong, J. Wang, F. Yin *et al.*, "Rcagent: Cloud root cause analysis by autonomous agents with tool-augmented large language models," in *Proc. 33rd ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Birmingham, UK, Oct. 2024, pp. 4966–4974.

[11] D. Zhang, X. Zhang, C. Bansal, P. Las-Casas, R. Fonseca, and S. Rajmohan, "Pace-lm: Prompting and augmentation for calibrated confidence estimation with gpt-4 in cloud incident root cause analysis," Sep. 2023, arXiv preprint arXiv:2309.05833.

[12] J. Shi, S. Jiang, B. Xu, J. Liang, Y. Xiao, and W. Wang, "Shellgpt: Generative pre-trained transformer model for shell language understanding," in *Proc. 2023 IEEE 34th Int. Symp. Software Reliability Engineering (ISSRE)*, Florence, Italy, Oct. 2023, pp. 671–682.

[13] Z. Yu, M. Ma, C. Zhang, S. Qin, Y. Kang, C. Bansal, S. Rajmohan, Y. Dang, C. Pei, D. Pei, Q. Lin, and D. Zhang, "Monitorassistant: Simplifying cloud service monitoring via large language models," in *Companion Proceedings of the 32nd ACM International Symposium on Foundations of Software Engineering (FSE Companion '24)*. Pernambuco, Brazil: ACM, Jul. 2024, p. 12–23. [Online]. Available: https://doi.org/10.1145/3663529.3663826

[14] X. Zhang, S. Ghosh, C. Bansal, R. Wang, M. Ma, Y. Kang, and S. Rajmohan, "Automated root causing of cloud incidents using in-context learning with GPT-4," in *Proceedings of the 2024 ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE Companion)*. Portland, OR, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3663529.3663846

[15] Y. Chen, H. Xie, M. Ma, Y. Kang, X. Gao, L. Shi, Y. Cao, X. Gao, H. Fan, M. Wen, J. Zeng, S. Ghosh, X. Zhang, C. Zhang, Q. Lin, S. Rajmohan, D. Zhang, and T. Xu, "Automatic root cause analysis via large language models for cloud incidents," in *Proceedings of the 19th European Conference on Computer Systems (EuroSys '24)*. Athens, Greece: ACM, Apr. 2024, p. 674–688. [Online]. Available: https://doi.org/10.1145/3627703.3629553

[16] B. Paranjape, S. Lundberg, S. Singh, H. Hajishirzi, L. Zettlemoyer, and M. T. Ribeiro, "Art: Automatic multi-step reasoning and tool-use for large language models," *arXiv preprint arXiv:2303.09014*, 2023, preprint. [Online]. Available: https://arxiv.org/abs/2303.09014

[17] Q. Wang, X. Zhang, M. Li, Y. Yuan, M. Xiao, F. Zhuang, and D. Yu, "Tamo: Fine-grained root cause analysis via tool-assisted llm agent with multi-modality observation data," *arXiv preprint arXiv:2504.20462*, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2504.20462

[18] J. Singh, R. Magazine, Y. Pandya, and A. Nambi, "Agentic reasoning and tool integration for llms via reinforcement learning," *arXiv preprint arXiv:2505.01441*, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2505.01441

[19] D. Roy, X. Zhang, R. Bhave, C. Bansal, P. H. Las-Casas, R. Fonseca, and S. Rajmohan, "Exploring llm-based agents for root cause analysis," in *Companion Proceedings of the 32nd ACM International Symposium on Foundations of Software Engineering (FSE Companion '24)*. Pernambuco, Brazil: ACM, Jul. 2024, p. 208–219. [Online]. Available: https://doi.org/10.1145/3663529.3663841

[20] Z. Wang, Z. Liu, Y. Zhang, A. Zhong, J. Wang, F. Yin, L. Fan, L. Wu, and Q. Wen, "Rcagent: Cloud root cause analysis by autonomous agents with tool-augmented large language models," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*. Boise, ID, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3627673.3680016

[21] D. Zhang, X. Zhang, C. Bansal, P. Las-Casas, R. Fonseca, and S. Rajmohan, "Pace-lm: Prompting and augmentation for calibrated confidence estimation with gpt-4 in cloud incident root cause analysis," *arXiv preprint arXiv:2309.05833*, 2023, preprint. [Online]. Available: https://arxiv.org/abs/2309.05833

[22] E. Coronado, R. Behravesh, T. Subramanya, A. Fernández-Fernández, M. S. Siddiqui, X. Costa-Pérez *et al.*, "Zero touch management: A survey of network automation solutions for 5g and 6g networks," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2535–2578, 2022.

[23] A. Hazra, A. Morichetta, I. Murturi, L. Lovén, C. K. Dehury, V. C. Pujol *et al.*, "Zero-touch networks: Towards next-generation network automation," Dec. 2023, arXiv preprint arXiv:2312.04159.

[24] H. Chergui, A. Ksentini, L. Blanco, and C. Verikoukis, "Toward zero-touch management and orchestration of massive deployment of network slices in 6g," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 86–93, Feb. 2022.

[25] J. Wang, L. Zhang, Y. Yang, Z. Zhuang, Q. Qi, and H. Sun, "Network meets chatgpt: Intent autonomous management, control and operation," *Journal of Communications and Information Networks*, vol. 8, no. 3, pp. 239–255, Sep. 2023.

[26] S. Cruzes, "Enhancing optical networks with large language models: An era of automated efficiency," Oct. 2024, techRxiv preprint.

[27] Y. Njah, A. Leivadeas, and M. Falkner, "An ai-driven intent-based network architecture," *IEEE Communications Magazine*, pp. 1–8, 2024, early access.

[28] D. M. Manias, A. Chouman, and A. Shami, "Towards intent-based network management: Large language models for intent extraction in 5g core networks," *arXiv preprint arXiv:2403.02238*, 2024, preprint. [Online]. Available: https://arxiv.org/abs/2403.02238

[29] Y. Njah, A. Leivadeas, J. Violos, and M. Falkner, "Toward intent-based network automation for smart environments: A healthcare 4.0 use case," *IEEE Access*, vol. 11, pp. 131 040–131 051, 2023. [Online]. Available: https://doi.org/10.1109/ACCESS.2023.3338189

[30] M. L. Romero and R. Suyama, "Agentic ai for intent-based industrial automation," *arXiv preprint arXiv:2506.04980*, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2506.04980

[31] J. McNamara, D. Camps-Mur, M. Goodarzi, H. Frank, and S. Yan, "Nlp powered intent based network management for private 5g networks," *IEEE Access*, vol. 11, pp. 84 756–84 769, 2023. [Online]. Available: https://doi.org/10.1109/ACCESS.2023.3301126

[32] K. Trantzas, D. Brodimas, B. Agko, G. C. Tziavas, C. Tranoris, S. Denazis, and A. Birbas, "Intent-driven network automation through sustainable multimodal generative ai," *EURASIP Journal on Wireless Communications and Networking*, 2025. [Online]. Available: https://doi.org/10.1186/s13638-025-02472-x

[33] ONAP, *APEX Policy Framework Overview*, ONAP Documentation, Jun. 2024.

[34] ETSI, "Network functions virtualisation (nfv) release 5; management and orchestration; os-ma-nfvo reference point – interface and information model specification," ETSI, Tech. Rep. GS NFV-IFA 013 V5.2.3, Sep. 2024.