International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 1, Jan-Feb-2024, ISSN (Online): 2395-566X

Intelligent Self-Optimizing Microservices Through Autonomous Feedback Loops

Samantha Green[1], Richard Morgan[2], Katherine Lewis[3], Benjamin Scott[4], Chaitanya Srinivas[5], Akhilesh Achari[6]

[1]Professor of Cyber-Physical Systems, [2]Senior Software Engineering Researcher, [3]Head of Intelligent Computing Research, [4]DevOps Research Lead, [5]Senior Java Software Developer, [6]Java Developer

Abstract- Modern cloud-native applications increasingly rely on microservice architectures to achieve scalability, flexibility, and resilience. However, the growing complexity of distributed environments presents significant challenges in performance management, resource allocation, fault detection, and service coordination. This paper proposes an intelligent self-optimizing microservice framework driven by autonomous feedback loops that continuously monitor, analyze, and adapt system behavior in real time. The framework integrates feedback-driven control models, artificial intelligence techniques, and automated decision-making mechanisms to dynamically optimize service performance, resource utilization, and operational reliability. By leveraging continuous feedback from runtime metrics, system events, and workload patterns, the proposed approach enables proactive adaptation to changing environmental conditions and application demands without human intervention. The study investigates key architectural components, optimization strategies, and autonomous control mechanisms that support self-healing, self-scaling, and self-configuring capabilities within microservice ecosystems. Experimental analysis demonstrates notable improvements in response time, throughput, fault tolerance, and infrastructure efficiency when compared with conventional static management approaches. The results indicate that autonomous feedback-driven optimization provides a robust foundation for developing intelligent, adaptive, and resilient microservice-based systems capable of meeting the demands of modern cloud and edge computing environments.

Keywords- Microservices Architecture, Self-Optimizing Systems, Autonomous Feedback Loops, Feedback-Driven Control Models, Intelligent Systems, Cloud-Native Computing, Distributed Systems, Adaptive Computing, Autonomous Computing, Self-Healing Systems, Self-Scaling Services, Self-Configuration, Service Orchestration, Dynamic Resource Allocation, Runtime Optimization, Artificial Intelligence, Machine Learning, Predictive Analytics, Reinforcement Learning, Observability, Real-Time Monitoring, Telemetry Analytics, Performance Optimization, Fault Tolerance, Resilience Engineering, Service Reliability, DevOps Automation, AIOps, Kubernetes, Containerized Applications, Edge Computing, Cloud Computing, Event-Driven Architecture, Service Mesh, Distributed Control Systems, Runtime Adaptation, Continuous Optimization, Workload Management, System Autonomy, Intelligent Automation, Cyber-Physical Systems, Decision Support Systems, Infrastructure Optimization, Elastic Computing, Software Architecture, Feedback Control Theory, Resource Management, Scalability Engineering, Autonomous Decision-Making, Next-Generation Distributed Applications.

I. INTRODUCTION

The rapid evolution of cloud computing and distributed software architectures has transformed the way modern applications are designed, deployed, and managed. Among these architectural paradigms, microservices have emerged as a dominant approach due to their ability to provide scalability, modularity, fault isolation, and continuous deployment capabilities. Unlike monolithic systems, microservices divide applications into independently deployable services that communicate through lightweight protocols. This architectural flexibility enables organizations to respond quickly to changing business requirements while maintaining high system availability and performance.

International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 1, Jan-Feb-2024, ISSN (Online): 2395-566X

Despite these advantages, managing large-scale microservice ecosystems presents significant challenges. As the number of services grows, system complexity increases substantially, making manual monitoring, configuration, and optimization inefficient and error-prone. Dynamic workloads, fluctuating resource demands, service failures, and network variability further complicate operational management. Traditional static optimization techniques are often unable to adapt effectively to these continuously changing conditions, resulting in performance degradation, resource wastage, and reduced service reliability.

Recent advances in artificial intelligence, machine learning, and autonomous computing have introduced new opportunities for developing self-managing software systems. Autonomous feedback loops provide a mechanism through which microservices can continuously monitor operational metrics, analyze system behavior, make intelligent decisions, and execute corrective actions without direct human intervention. These capabilities support self-healing, self-scaling, and self-optimization functions that improve overall system efficiency and resilience.

This research investigates the concept of Intelligent Self-Optimizing Microservices Through Autonomous Feedback Loops. The proposed framework integrates real-time monitoring, feedback-driven control models, predictive analytics, and automated adaptation strategies to create intelligent microservice ecosystems capable of continuous optimization. The study explores architectural principles, optimization mechanisms, implementation strategies, and performance benefits associated with autonomous feedback-driven microservices in modern cloud-native environments.

II. FOUNDATIONS OF MICROSERVICE ARCHITECTURE

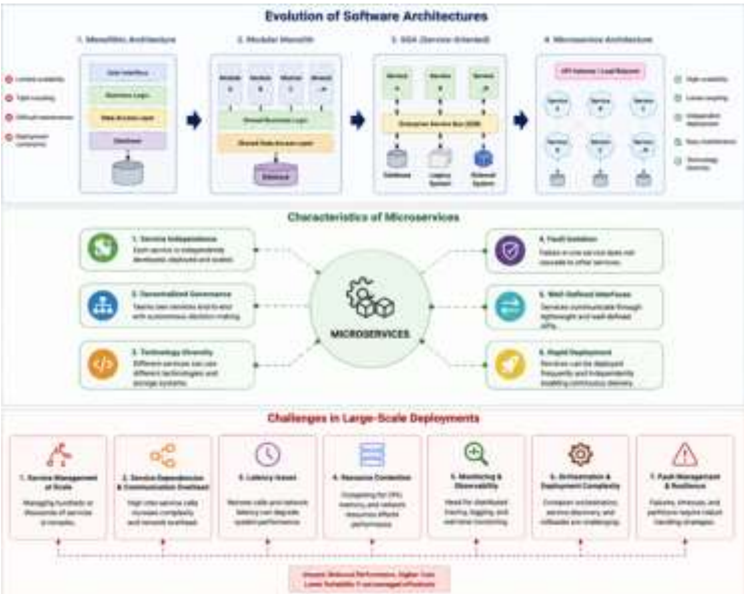
Evolution of Software Architectures

Software architectures have evolved from tightly coupled monolithic systems to highly distributed microservice-based environments. Traditional monolithic applications often suffer from limited scalability, difficult maintenance procedures, and deployment constraints. Microservices address these limitations by decomposing applications into smaller services that can be independently developed, deployed, and maintained. This modular approach enhances flexibility while reducing system dependencies.

Characteristics of Microservices

Microservices possess several characteristics that contribute to their widespread adoption. These include service independence, decentralized governance, technology diversity, fault isolation, and rapid deployment capabilities. Each service focuses on a specific business function and communicates through well-defined interfaces. Such characteristics enable organizations to build scalable systems that can evolve continuously without affecting overall application stability.

Challenges in Large-Scale Deployments



International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 1, Jan-Feb-2024, ISSN (Online): 2395-566X

Although microservices provide significant advantages, they also introduce operational complexities. Managing hundreds or thousands of interconnected services requires sophisticated monitoring, orchestration, resource allocation, and fault management strategies. Service dependencies, communication overhead, latency issues, and resource contention can negatively impact system performance if not managed effectively.

III. AUTONOMOUS FEEDBACK LOOPS IN MICROSERVICE SYSTEMS

Concept of Feedback Loops

Feedback loops are control mechanisms that continuously collect information about system behavior and use this information to influence future actions. In microservice environments, feedback loops enable systems to observe operational conditions, identify deviations from desired performance levels, and initiate corrective actions. This continuous cycle forms the foundation of autonomous optimization.

Monitor-Analyze-Plan-Execute Framework

Autonomous feedback loops commonly follow the Monitor-Analyze-Plan-Execute (MAPE) model. During monitoring, telemetry data such as CPU utilization, memory consumption, latency, and throughput are collected. Analytical components process this data to identify patterns and anomalies. Planning modules determine optimal corrective actions, while execution components implement these actions automatically within the system.

Real-Time Decision Making

Real-time decision-making capabilities allow microservices to adapt dynamically to changing workloads and environmental conditions. Machine learning algorithms can identify emerging trends, predict future resource requirements, and recommend optimization strategies. This proactive approach reduces service disruptions and improves system responsiveness.

IV. SELF-OPTIMIZATION MECHANISMS

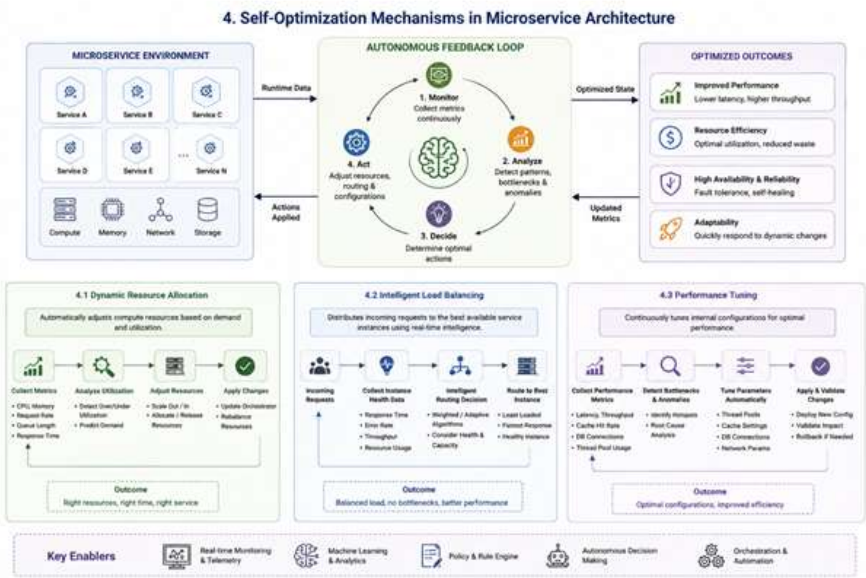
Dynamic Resource Allocation

Resource allocation plays a critical role in maintaining system performance. Autonomous feedback loops continuously evaluate resource utilization and adjust allocations accordingly. Services experiencing increased demand receive additional computational resources, while underutilized services release excess resources to improve overall infrastructure efficiency.

Intelligent Load Balancing

Load balancing mechanisms distribute workloads across multiple service instances to prevent bottlenecks and ensure optimal performance. Intelligent load balancing algorithms utilize feedback data to dynamically route requests based on current service health, response times, and resource availability.

Performance Tuning



International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 1, Jan-Feb-2024, ISSN (Online): 2395-566X

Continuous performance tuning enables microservices to optimize execution parameters automatically. Feedback-driven optimization identifies performance bottlenecks and adjusts configurations such as thread pools, cache settings, database connections, and network parameters to maximize efficiency.

V. ARTIFICIAL INTELLIGENCE FOR AUTONOMOUS OPTIMIZATION

Machine Learning Integration

Machine learning techniques provide predictive and adaptive capabilities within autonomous microservice environments. Historical operational data can be analyzed to forecast workload patterns, identify potential failures, and optimize resource provisioning strategies. These predictive capabilities improve system stability and reduce operational risks.

Reinforcement Learning Approaches

Reinforcement learning enables systems to learn optimal behaviors through interaction with their environment. Microservices can evaluate the effectiveness of optimization actions and continuously refine decision-making policies. This adaptive learning process enhances long-term system performance.

Predictive Analytics

Predictive analytics transforms raw telemetry data into actionable insights. By forecasting future conditions, systems can proactively scale resources, redistribute workloads, and prevent service degradation before issues impact users.

VI. SELF-HEALING AND RESILIENCE ENGINEERING

Automated Fault Detection

Fault detection mechanisms continuously monitor service health indicators to identify anomalies and failures. Autonomous systems utilize statistical analysis and machine learning models to distinguish between normal operational variations and critical issues requiring intervention.

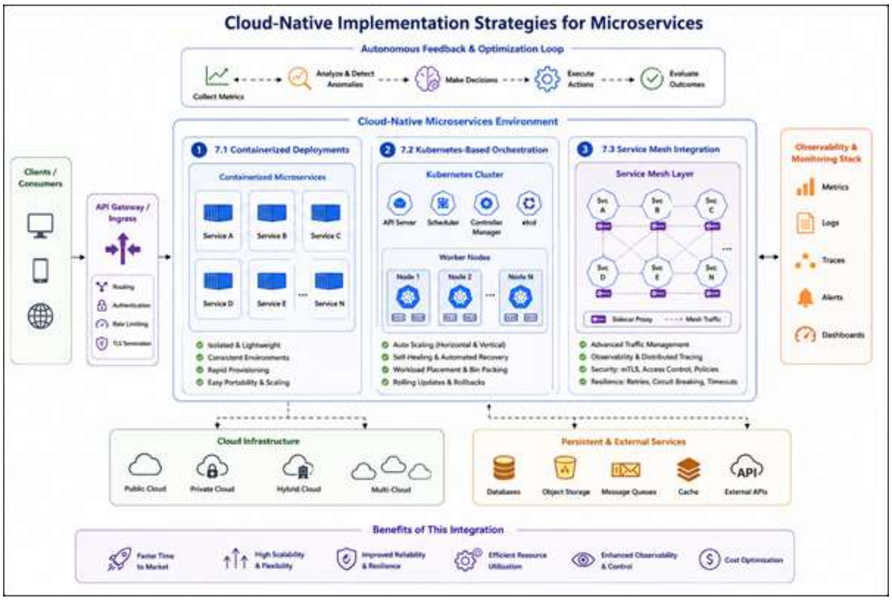
Failure Recovery Mechanisms

Self-healing capabilities enable microservices to recover from failures without human assistance. Recovery actions may include restarting failed services, rerouting traffic, reallocating resources, or activating backup instances. These mechanisms minimize downtime and improve service availability.

Resilience Enhancement

Resilient microservice architectures incorporate redundancy, fault tolerance, and adaptive recovery strategies. Autonomous feedback loops strengthen resilience by continuously evaluating system robustness and implementing preventative measures against potential disruptions.

VII. CLOUD-NATIVE IMPLEMENTATION STRATEGIES



International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 1, Jan-Feb-2024, ISSN (Online): 2395-566X

Containerized Deployments
Container technologies provide isolated execution environments that simplify deployment and scalability. Containers enable rapid provisioning of microservice instances while supporting automated orchestration and lifecycle management.

Kubernetes-Based Orchestration
Kubernetes serves as a foundational platform for managing cloud-native microservices. Autonomous feedback loops can integrate with orchestration platforms to automate scaling decisions, workload placement, and service recovery operations.

Service Mesh Integration
Service mesh technologies provide advanced traffic management, observability, and security capabilities. Integrating feedback-driven optimization mechanisms with service meshes enhances system visibility and enables more intelligent decision-making.

VIII. BENEFITS AND FUTURE DIRECTIONS

Operational Efficiency
Autonomous optimization significantly reduces manual administrative tasks by automating monitoring, decision-making, and corrective actions. This improves operational efficiency while lowering management costs.

Enhanced Scalability
Self-optimizing microservices dynamically adapt to workload fluctuations, enabling organizations to achieve greater scalability without extensive manual intervention. Resources are utilized more effectively, reducing waste and improving performance.

Future Research Opportunities
Future research may explore advanced reinforcement learning techniques, federated optimization models, edge-cloud collaboration frameworks, and explainable artificial intelligence for autonomous systems. These advancements have the potential to further enhance the intelligence, transparency, and adaptability of next-generation microservice ecosystems.

IX. CONCLUSION

The increasing complexity of cloud-native applications and distributed computing environments has created a growing need for intelligent and autonomous management solutions. Traditional approaches to monitoring, configuration, scaling, and performance optimization are often inadequate for handling the dynamic nature of modern microservice ecosystems. As organizations continue to adopt large-scale distributed architectures, the ability to automatically adapt to changing workloads, resource constraints, and operational conditions becomes essential for maintaining service quality and business continuity.

This research explored the concept of Intelligent Self-Optimizing Microservices Through Autonomous Feedback Loops as a comprehensive framework for achieving autonomous system management. By integrating continuous monitoring, feedback-driven control mechanisms, artificial intelligence, machine learning, predictive analytics, and automated decision-making processes, the proposed approach enables microservices to independently optimize their performance, resource utilization, and operational reliability. Autonomous feedback loops facilitate real-time observation of system behavior, allowing services to detect anomalies, predict future conditions, and execute corrective actions without requiring constant human intervention.

The study demonstrated that self-optimizing microservice architectures provide several advantages over conventional management approaches. These advantages include improved scalability, enhanced fault tolerance, faster response times, efficient resource allocation, reduced operational overhead, and greater resilience against system failures. Self-healing capabilities ensure rapid recovery from faults, while intelligent scaling mechanisms dynamically adjust resources according to workload demands. Furthermore, predictive analytics and learning-based optimization techniques enable proactive adaptation, helping systems prevent performance degradation before it impacts end users.

Another significant contribution of autonomous feedback-driven systems is their ability to support continuous improvement. Through ongoing analysis of operational data and environmental changes, microservices can refine optimization strategies over time, leading to progressively better performance and resource efficiency. This continuous learning capability aligns with the broader vision of autonomic computing, where software systems possess self-managing characteristics such as self-configuration, self-optimization, self-healing, and self-protection.

The integration of container orchestration platforms, service mesh technologies, observability frameworks, and AI-powered analytics further strengthens the effectiveness of autonomous microservice environments. These technologies collectively provide the visibility, control, and intelligence required to implement adaptive optimization strategies at scale. As cloud and edge computing infrastructures continue to expand, such autonomous capabilities will become increasingly important for managing geographically distributed and highly dynamic application ecosystems.

Despite the promising outcomes, several challenges remain. Issues related to model accuracy, decision transparency, security, interoperability, and computational overhead require further investigation. Ensuring that autonomous optimization decisions remain explainable and trustworthy is particularly important for enterprise adoption. Future research should focus on advanced reinforcement learning algorithms, explainable artificial intelligence techniques, federated optimization models, digital twins, and hybrid cloud-edge autonomous architectures to further enhance system intelligence and adaptability.

In conclusion, Intelligent Self-Optimizing Microservices Through Autonomous Feedback Loops represents a transformative approach to modern software system management. By enabling continuous adaptation, autonomous decision-making, and real-time optimization, this paradigm offers a robust foundation for building highly scalable, resilient, and efficient distributed applications. As technological advancements continue to accelerate, autonomous feedback-driven microservices are expected to play a central role in shaping the next generation of intelligent cloud-native systems, ultimately driving greater operational excellence, sustainability, and innovation across diverse computing environments.

REFERENCES

1. White, S. R., Hanson, J. E., Whalley, I., Chess, D. M., & Segal, A. (2006). Autonomic computing: Architectural approach and prototype. IBM Systems Journal, 13(2). https://doi.org/10.3233/ICA-2006-13206
2. Thota, M. R. (2023). Scalable multi-cloud workload orchestration: Integrating big data and database operations through Google Cloud Platform. Journal of Scientific and Engineering Research, 10(2), 247–264. https://doi.org/10.5281/zenodo.17840000
3. Vollem, S. (2023). Artificial intelligence for root cause analysis in cloud-native systems: Techniques, architectures, and research trends. European Journal of Advances in Engineering and Technology, 10(9), 120–129. https://doi.org/10.5281/zenodo.19347481
4. Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. Journal of Systems and Software, 150, 77–97. https://doi.org/10.1016/j.jss.2019.01.001
5. Ghanta, S. (2016). Designing for scale: API-first architectural patterns for resilient enterprise systems. International Journal of Technology, Management and Humanities, 2(2), 20–31. https://doi.org/10.21590/ijtmh.2.02.3
6. Seetala SR. Real-Time Data Monitoring Using Cloud Observability Tools: Architectures, Techniques and Emerging Practices. J Artif Intell Mach Learn & Data Sci 2023 6(4), 3367-3374. DOI: doi.org/10.51219/JAIMLD/srinivasa-rao-seetala/673
7. Vankayala, S. C. (2023). Reinforcement learning–driven cognitive testing for scalable and resilient financial systems. ESP Journal of Engineering & Technology Advancements, 3(4), 209–217. https://doi.org/10.5281/zenodo.20092735
8. Nanchari, N. (2023). Real-time health monitoring and alerts via IoT. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 9(4), 710–713. https://doi.org/10.32628/CSEIT23564523
9. Parepalli, S. (2023). Engineering end-to-end data integrity validation for financial reporting pipelines: Continuous controls, reconciliation evidence, and tamper-resistant governance. International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), 10(9), 416–433. https://doi.org/10.32628/IJSRSET2310946
10. Hasselbring, W., & Steinacker, G. (2017). Microservice architectures for scalability, agility and reliability in e-commerce. In IEEE ICSAW 2017. https://doi.org/10.1109/ICSAW.2017.11

11. Teegala, R. (2023). Secure prompt engineering for banking and payment applications: Design principles, threat models, and governance controls for generative AI in regulated financial systems. KOS Journal of AIML, Data Science, and Robotics, 1(1), 1–9. https://doi.org/10.5281/zenodo.18712372
12. Nagender, Y. (2023). Architecting intelligence into master data platforms: An evidence mapping approach to AI-enabled dashboards for compliance and quality monitoring. International Journal of Scientific Research & Engineering Trends, 9(6). https://doi.org/10.5281/zenodo.18770933
13. Menda, J. R. (2022). Grounded generation for enterprise knowledge: Automated documentation and knowledge extraction using GenAI agents. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 8(3), 857–866. https://doi.org/10.32628/CSEIT2215512
14. Alshuqayran, N., Ali, N., & Evans, R. (2016). A systematic mapping study in microservice architecture. In 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA) (pp. 44–51). https://doi.org/10.1109/SOCA.2016.15
15. BasiReddy, S. R. (2023). From automation to accountability: Ethical AI in CRM workflows. International Journal of Scientific Research & Engineering Trends, 9(4). Zenodo. https://doi.org/10.5281/zenodo.18326172
16. Thota, M. R. (2021). Intelligent infrastructure as code: GitOps, resilience engineering, and automation patterns for scalable big data platforms. International Journal of Scientific Research in Science and Technology, 8(5), 741–752. https://doi.org/10.32628/IJSRST2152553
17. Waseem, M., Liang, P., & Shahin, M. (2020). A systematic mapping study on microservices architecture in DevOps. Journal of Systems and Software, 170, 110798. https://doi.org/10.1016/j.jss.2020.110798
18. Nanchari, N. (2023). IoT for mental health monitoring. European Journal of Advances in Engineering and Technology, 10(2), 75–77. https://doi.org/10.5281/zenodo.15969008
19. Vankayala, S. C. (2023). AI-augmented root cause analysis in distributed microservices: A deep learning and causal inference framework for intelligent quality engineering. International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), 10(6), 499–512. https://doi.org/10.32628/IJSRSET2613251
20. Vollem S. Governance Models for Microservice Architectures in Regulated Enterprise Environments. J Artif Intell Mach Learn & Data Sci 2021 3(3), 3343-3349. DOI: doi.org/10.51219/JAIMLD/shekar-vollem/670
21. Ghanta, S. (2022). Privacy-preserving machine learning for regulated financial systems: A federated learning architecture with layered privacy guarantees. International Journal of Core Engineering & Management, 7(4). https://doi.org/10.5281/zenodo.18920980
22. Hannousse, A., & Yahiouche, S. (2021). Securing microservices and microservice architectures: A systematic mapping study. Computer Science Review, 41, 100415. https://doi.org/10.1016/j.cosrev.2021.100415
23. Seetala, S. R. (2023). Automated data reconciliation using intelligent algorithms: Architectures, techniques, and applications in modern enterprise systems. International Journal of Science, Engineering and Technology, 11(3). https://doi.org/10.5281/zenodo.19217777
24. Parepalli, S. (2023). Operationalizing responsible AI in financial decision pipelines: Governance, security, compliance, fairness, and explainability. International Journal of Scientific Research & Engineering Trends, 9(4). https://doi.org/10.5281/zenodo.18641518
25. Yamsani, N. (2023). Institutionalizing data accountability: Automation patterns for governance, lineage, and compliance in enterprise platforms. International Journal of Machine Learning for Sustainable Development, 5(2), 1–28. Retrieved from https://www.ijsdcs.com/index.php/IJMLSD/article/view/708/271
26. Menda, J. R. (2021). Building resilient and compliance-driven observability architectures for modern BFSI enterprises using unified monitoring, telemetry correlation, and proactive incident intelligence. International Journal of Science, Engineering and Technology, 9(1). https://doi.org/10.5281/zenodo.18107872
27. Hassan, S., Bahsoon, R., & Kazman, R. (2020). Microservice transition and its granularity problem: A systematic mapping study. Software: Practice and Experience, 50(9), 1719–1740. https://doi.org/10.1002/spe.2869
28. Teegala, R. (2023). Retrieval augmented generation driven operational runbooks for cloud native systems. European Journal of Advances in Engineering and Technology, 10(6), 107–119. https://doi.org/10.5281/zenodo.19565112

29. Vankayala, S. C. (2019). Predictive defect governance and decision optimization in mortgage underwriting platforms. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5(1), 382–398. https://doi.org/10.32628/CSEIT24254146
30. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. Computer, 36(1), 41–50. https://doi.org/10.1109/MC.2003.1160055
31. Nanchari, N. (2022). Data privacy and security challenges in IoT healthcare. International Journal of Scientific Research & Engineering Trends, 8(6). https://doi.org/10.5281/zenodo.15796381
32. BasiReddy, S. R. (2022). Augmenting customer relationship management workflows with generative AI: Architectures, conversational intelligence, and knowledge-grounded personalization. International Journal of Scientific Research & Engineering Trends, 8(5). Zenodo. https://doi.org/10.5281/zenodo.18324413
33. Seetala SR. Intelligent Data Validation in Modern Data Platforms: Integrating Statistical Methods and AI for Reliable Machine Learning Pipelines. J Artif Intell Mach Learn & Data Sci 2022 5(2), 3359-3366. doi.org/10.51219/JAIMLD/srinivasa-rao-seetala/672
34. Huebscher, M. C., & McCann, J. A. (2008). A survey of autonomic computing—Degrees, models, and applications. ACM Computing Surveys, 40(3), 1–28. https://doi.org/10.1145/1380584.1380585
35. Thota, M. R. (2018). Transforming database leadership in the era of cloud-native automation and resilient operations. International Journal of Technology, Management and Humanities, 4(2), 25–43. https://doi.org/10.21590/ijtmh.04.02.04
36. Salehie, M., & Tahvildari, L. (2009). Self-adaptive software: Landscape and research challenges. ACM Transactions on Autonomous and Adaptive Systems, 4(2), 1–42. https://doi.org/10.1145/1516533.1516538
37. Ghanta, S. (2023). From open information extraction to probabilistic fusion: Semantic retrieval pipelines for enterprise knowledge graph construction. International Journal of Research and Applied Innovations, 6(3), 8933–8940. https://doi.org/10.15662/IJRAI.2025.080201
38. Yamsani, N. (2023). Context-aware metadata enrichment in enterprise master data management: A natural language processing approach for EBX repositories. International Journal of Sustainable Development in Computing Science, 5(1), 1–28. Retrieved from https://www.ijsdcs.com/index.php/ijsdcs/article/view/707/270
39. Menda, J. R. (2020). Designing an intelligent framework for automated governance and enterprise risk management through machine learning-driven signals and predictive analytics. International Journal of Science, Engineering and Technology, 8(6). https://doi.org/10.5281/zenodo.18085147
40. Vollem, S. (2021). A layered financial-grade API security architecture: Integrating OAuth 2.0, FAPI, Open Banking, and regulatory controls for high-assurance financial platforms. International Journal of Machine Learning and Software Development, 3(1). DOI: 10.32589/ijmlsd.2021.007
41. Vankayala, S. C. (2022). Intelligent failure prediction in CI/CD pipelines using machine learning models for enterprise quality assurance. International Journal of Scientific Research in Science and Technology, 9(6), 820–832. https://doi.org/10.32628/IJSRST52310497
42. Parepalli, S. (2022). Semantic and reasoning driven approaches to automated error classification in large scale ETL systems. European Journal of Advances in Engineering and Technology, 9(11), 151–162. https://doi.org/10.5281/zenodo.18084352
43. Ramani Teegala. (2022). Self Healing Microservices: Adaptive Resilience And Autonomous Recovery In Distributed Systems. In International Journal of Science, Engineering and Technology (Vol. 14, Number 2). Zenodo. https://doi.org/10.5281/zenodo.18680202
44. BasiReddy, S. R. (2022). From static personalization to adaptive intelligence: Building context-aware CRM recommendation systems with AI agents. International Journal of Science, Engineering and Technology, 10(3). Zenodo. https://doi.org/10.5281/zenodo.18183174