Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Designing Resilient Multi-Tenant Platforms: The Role of AI in Scalable Cloud-Native SRE Pipelines

Susanta Kumar Sahoo
Independent Researcher, USA

ARTICLE INFO

Received: 15 July 2025

Revised: 22 Aug 2025

Accepted: 07 Sept 2025

ABSTRACT

Multi-tenant architectures present unique reliability challenges for Site Reliability Engineering teams, requiring solutions beyond traditional manual interventions and static rules. This article explores the integration of artificial intelligence into cloud-native SRE pipelines to enhance fault prediction, incident management, and automated remediation in distributed environments. The architecture encompasses time-series models for anomaly detection, NLP systems for incident classification, reinforcement learning for automated remediation, and adaptive resource management across tenant boundaries. The implementation strategies and real-world applications, the paper demonstrates how ML-augmented SRE practices transform reliability operations while addressing challenges including model drift, interpretability, data quality, and fairness considerations. The integration of machine learning with established reliability practices creates a foundation for autonomous, self-healing platforms that maintain resilience at scale while delivering consistent experiences across diverse tenant populations.

Keywords: Multi-tenancy, Site Reliability Engineering, Artificial Intelligence, Automated Remediation, Cloud-Native Architecture

1. Introduction

The emergence of multi-tenant architectures as the foundation for modern SaaS and PaaS offerings has created unprecedented challenges for Site Reliability Engineering (SRE) teams. As organizations serve multiple customers through a single software instance, they face complex reliability issues, including tenant isolation, varying SLAs, and dynamic failure modes. Traditional SRE approaches that rely on manual intervention and static rules are increasingly inadequate for maintaining resilience at scale. This paper explores how artificial intelligence, particularly machine learning models, can be integrated into cloud-native SRE pipelines to enhance fault prediction, incident management, and automated remediation in distributed multi-tenant environments.

The rapid evolution of cloud computing has fundamentally transformed software delivery paradigms, with multi-tenant architectures introducing unique reliability challenges beyond traditional infrastructure concerns. Research reveals these systems exhibit significantly more complex failure patterns with cascading effects that propagate across tenant boundaries when inadequately contained. The interdependencies between shared resources, network paths, and storage systems create vulnerability points that conventional monitoring approaches fail to address proactively [1]. Multi-tenant platforms require evolving SRE practices capable of handling these multi-dimensional reliability concerns, especially as business-critical workloads continue migrating to shared environments.

Multi-tenant reliability extends beyond technical challenges to encompass operational considerations, particularly regarding diverse Service Level Agreements. The variability between tenant tiers creates complex reliability matrices that must be simultaneously satisfied, introducing difficult prioritization decisions during incident response. Increasing tenant density compounds these challenges through

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

heightened resource contention and expanded failure impact zones. SRE practitioners report growing difficulty balancing consistent reliability metrics across tenant categories while preserving multi-tenancy's economic advantages [1]. These tensions highlight the need for more sophisticated reliability engineering approaches capable of dynamically balancing competing objectives.

Artificial intelligence presents promising solutions to these multi-tenant reliability challenges. Machine learning models can process extensive telemetry datasets to identify subtle patterns preceding service degradations, enabling truly proactive interventions. AI systems can automate incident classification, dramatically reducing operator cognitive load while accelerating triage processes. The application of reinforcement learning to remediation workflows represents a significant advancement, allowing systems to develop optimal recovery strategies based on historical outcomes rather than static runbooks [2]. These approaches demonstrate substantial improvements in key reliability metrics, suggesting AI integration represents a fundamental shift in resilience engineering for modern distributed systems.

The integration of machine learning within SRE workflows introduces implementation challenges requiring careful consideration. Data quality remains paramount, as models depend entirely on their training data. Multi-tenant environments magnify this challenge through the need to maintain tenant data separation while extracting meaningful reliability insights. Additionally, the opacity of advanced ML algorithms can impede troubleshooting efforts, potentially undermining trust in automated systems. Organizations implementing AI-enhanced SRE practices must develop new competencies bridging data science, software engineering, and operational excellence [2]. Despite these challenges, the potential benefits of intelligent automation in multi-tenant reliability engineering justify continued investment in this emerging field.

2. Challenges in Multi-Tenant SRE

Multi-tenant architectures introduce unique reliability challenges that exceed the capabilities of conventional SRE practices. The "noisy neighbor" problem occurs when one tenant's workload degrades performance for others. Enforcing diverse SLAs across tenants requires sophisticated monitoring and prioritization. Fault containment becomes critical to prevent cascading failures across tenant boundaries. Additionally, the volume and heterogeneity of telemetry data in multi-tenant environments necessitate advanced signal processing techniques. This section examines these challenges and establishes why ML-driven automation has become essential for modern SRE teams operating cloud-native platforms.

The "noisy neighbor" phenomenon represents one of the most persistent challenges in multi-tenant reliability engineering. In cloud-native architectures, resource sharing across computational, network, and storage layers creates numerous opportunities for tenant interference. Despite advances in container orchestration and virtualization, complete workload isolation remains elusive. The challenge manifests across multiple dimensions: CPU contention when tenants compete for processing cycles, memory pressure when applications exceed allocations, network saturation from bandwidth-intensive operations, and I/O contention during concurrent storage access. Contemporary platforms implement various mitigation strategies, including resource quotas, quality-of-service policies, and dynamic throttling, yet these approaches often prove insufficient in high-density environments optimized for cost efficiency. The complexity increases exponentially in microservice architectures where service dependencies create intricate resource utilization patterns that defy simple isolation policies [3].

Service Level Agreement management presents multifaceted challenges where customers occupy different service tiers with varying reliability expectations. The fundamental complexity stems from maintaining multiple, sometimes conflicting, service guarantees on shared infrastructure. This challenge becomes particularly acute during degraded system states when resource constraints force difficult prioritization decisions that may adversely affect lower-tier tenants to preserve service quality for premium customers. Multi-tenant platforms typically define distinct service tiers with progressively stricter reliability

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

commitments, creating a matrix of availability targets, performance thresholds, and recovery time objectives that must be simultaneously satisfied. Traditional monitoring approaches often fail to incorporate business context, leading to situations where technical severity assessments misalign with actual business impact across the tenant spectrum [4].

Fault containment emerges as a critical concern where the failure impact extends beyond individual customer boundaries. Effective fault isolation requires consideration at multiple abstraction layers, from infrastructure partitioning to application-level bulkheads. Platforms employ various isolation patterns, including network segmentation, computational isolation through dedicated node pools, and logical separation through tenant-aware middleware. Despite these measures, research reveals isolation boundaries frequently fail during complex outages, particularly when underlying infrastructure components experience degradation. Failure propagation often follows unexpected routes through shared dependencies not immediately apparent in architecture diagrams [3].

The telemetry challenge stems from both volume and the contextual complexity of observability data. Modern platforms generate massive quantities of logs, metrics, and traces across distributed components. Observability signals must be accurately attributed to specific tenants, workloads, and service tiers for meaningful analysis. The cardinality explosion when tenant identifiers are added to metrics creates performance challenges for traditional systems. Establishing baseline behavior becomes significantly more difficult as each tenant exhibits unique usage patterns, workload characteristics, and growth trajectories. Anomaly detection requires tenant-aware models that can distinguish between normal variation and genuine service degradation within specific customer contexts [4].

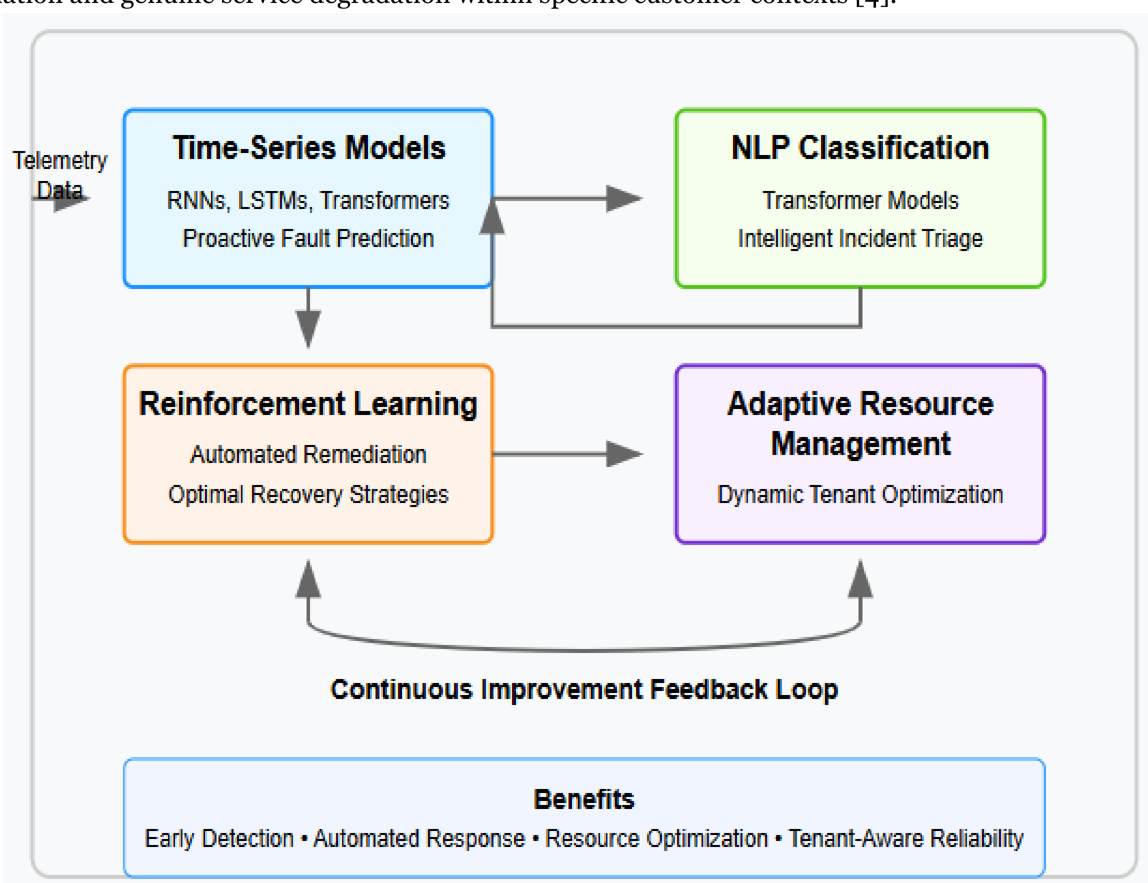

Fig 1: ML-Augmented SRE Pipeline Architecture [3, 4]

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

3. ML-Augmented SRE Pipeline Architecture

A comprehensive architecture for integrating machine learning into SRE workflows offers transformative capabilities for modern cloud-native platforms. Time-series models using RNNs, LSTMs, and Transformer-based architectures enable proactive fault prediction by analyzing historical metrics and identifying emerging anomalies. Intelligent incident classification leverages NLP to transform unstructured data into actionable insights, reducing alert fatigue and accelerating triage. Automated remediation systems employ reinforcement learning to execute optimal recovery actions based on contextual understanding. Finally, adaptive resource management uses ML to dynamically adjust tenant resources, optimize autoscaling, and enforce admission control policies. This architecture creates a feedback loop that continuously improves platform resilience.

The foundation of ML-augmented SRE pipelines lies in advanced time-series analysis for anomaly detection and predictive alerting. Contemporary architectures leverage deep learning models that process vast quantities of telemetry data to establish normal operating patterns and identify subtle deviations that often precede service disruptions. These models function across multiple timescales, detecting both immediate anomalies and gradual drift patterns that might otherwise escape human attention. The effectiveness stems from sophisticated feature engineering that extracts meaningful signals from high-dimensional metric spaces. Successful implementations typically combine multiple model architectures to balance sensitivity and specificity, with ensemble approaches demonstrating particular effectiveness in multi-tenant environments where workload patterns vary significantly across customer segments. This capability proves especially valuable in complex distributed systems where traditional threshold-based monitoring generates excessive noise and fails to capture subtle interdependencies between system components [5].

Intelligent incident classification systems address the growing complexity of operational data by applying natural language processing techniques to unstructured logs, alerts, and system events. These classification engines transform chaotic incident data into structured, actionable insights by identifying patterns across historical incidents and mapping new events to known categories. The most effective implementations employ transformer-based models that capture contextual relationships within log data, extracting meaningful signals from operational telemetry. Beyond basic categorization, advanced systems perform causal analysis to identify root factors and dependencies, dramatically accelerating the triage process. This capability proves particularly valuable in multi-tenant environments where a single underlying issue might manifest differently across tenant workloads, creating the appearance of multiple distinct incidents [5].

Automated remediation through reinforcement learning represents perhaps the most ambitious component of ML-augmented SRE architectures. These systems move beyond simple runbooks to develop sophisticated recovery strategies based on observed outcomes across thousands of incidents. The reinforcement learning approach frames incident remediation as a sequential decision problem, where agents learn optimal action sequences by maximizing rewards associated with successful recovery while minimizing service disruption. Advanced implementations employ hierarchical reinforcement learning to decompose complex remediation workflows into manageable sub-tasks, allowing more efficient learning and better generalization to novel failure scenarios [6].

Adaptive resource management completes the architecture by dynamically optimizing infrastructure allocation based on workload patterns and performance requirements. These systems leverage predictive models to anticipate resource needs across tenant workloads, enabling proactive scaling decisions that prevent both resource contention and wasteful over-provisioning. The effectiveness stems from sophisticated workload characterization models that identify patterns across multiple dimensions, including diurnal cycles, seasonal variations, and growth trends specific to individual tenants. Advanced implementations employ reinforcement learning to optimize complex resource allocation decisions,

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

learning effective policies through repeated interaction with the environment rather than relying on static rules [6].

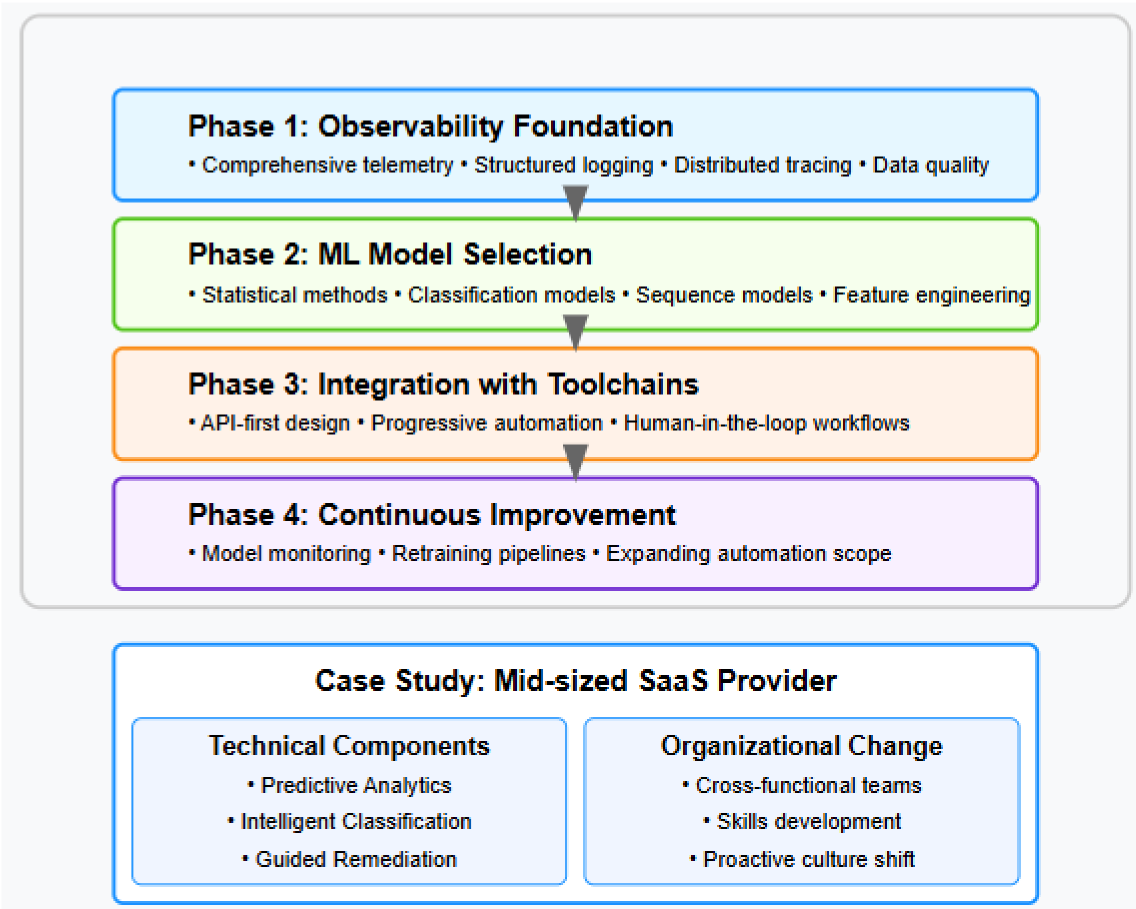

Fig 2: ML-Enhanced SRE Implementation Strategy [5, 6]

4. Implementation Strategies and Case Study

This section presents practical implementation strategies for organizations adopting ML-enhanced SRE practices. We discuss data collection requirements, model selection criteria, and integration patterns with existing cloud-native toolchains. A detailed case study of a mid-sized SaaS provider demonstrates how ML integration across hundreds of tenants led to significant improvements: 60% reduction in critical outages through predictive analytics, 70% faster incident triage, 90% success rate in automated remediation, and 25% cost savings through optimized resource allocation. We analyze the technical approaches and organizational changes that enabled these outcomes.

The implementation of machine learning within site reliability engineering workflows necessitates careful consideration of organizational readiness and data maturity. Successful adoption patterns begin with establishing comprehensive observability foundations that generate sufficiently rich telemetry data across the technology stack. This foundational step involves instrumenting applications and infrastructure to capture metrics at appropriate granularity, implementing structured logging practices that facilitate machine analysis, and deploying distributed tracing to understand request flows across service boundaries. Beyond basic data collection, mature implementations emphasize data quality attributes including consistent naming conventions, accurate timestamps, and appropriate context preservation.

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

The observability maturity model includes several progressive stages, from basic monitoring through advanced predictive capabilities, with organizations typically requiring deliberate advancement through each phase. Cross-functional collaboration proves essential during implementation, with effective programs establishing shared ownership between traditional SRE teams and data science specialists. This collaborative approach helps address common challenges, including data silos, inconsistent labeling practices, and the integration of domain expertise into model development. Organizations that successfully navigate these challenges typically establish dedicated machine learning platforms that standardize model lifecycle management, including versioning, validation, and deployment workflows specifically tailored to operational use cases [7].

The selection and evolution of appropriate machine learning methodologies represent crucial decisions in implementing enhanced reliability practices. Pattern recognition across successful implementations reveals a maturity progression that begins with statistical anomaly detection before advancing to more sophisticated techniques. Initial deployments frequently leverage univariate statistical methods and clustering algorithms to establish baseline detection capabilities, with these approaches offering advantages in explainability and modest data requirements. As implementations mature, more advanced techniques, including supervised classification for incident categorization and sequence models for pattern recognition, demonstrate increasing prevalence. The model selection process incorporates multiple factors beyond raw accuracy, including computational efficiency, interpretability requirements, and the ability to function effectively with limited labeled examples. Feature engineering remains critically important despite advances in representation learning, with domain-specific knowledge incorporation significantly improving model performance in operational contexts. Time-based cross-validation strategies prove particularly valuable in reliability applications, allowing organizations to evaluate how models would have performed historically while accounting for concept drift in operational patterns. The most sophisticated implementations establish continuous learning pipelines that automatically retrain models as new operational data becomes available, incorporating feedback loops that progressively improve detection and remediation capabilities [7].

The integration of machine learning capabilities with existing operational workflows and toolchains represents a significant implementation challenge that requires careful architectural consideration. Successful patterns emphasize augmentation rather than replacement, with ML systems enhancing human capabilities rather than attempting to fully automate complex reliability functions. The implementation architecture typically involves several distinct components: data pipelines that transform and normalize telemetry for model consumption, inference services that apply trained models to real-time data streams, notification systems that communicate predictions to appropriate stakeholders, and workflow engines that orchestrate potential automated responses. API-first designs facilitate integration with existing monitoring platforms, incident management systems, and communication tools, allowing incremental adoption without disrupting established processes. Progressive automation represents a common pattern, beginning with "human-in-the-loop" approaches where ML systems provide recommendations but require explicit approval before action. As confidence in model performance increases, organizations gradually expand the scope of autonomous operations while maintaining appropriate guardrails and oversight mechanisms. Integration challenges frequently arise around model deployment and operational monitoring, with successful implementations establishing robust observability for the ML systems themselves to detect issues such as prediction drift, data quality problems, or unexpected model behavior [8].

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Aspect	Summary
Data Foundation	Rich telemetry, structured logging, and observability maturity
Team Collaboration	SRE and data science partnership, shared model ownership
ML Techniques	Start with anomaly detection, evolve to classification and sequence models
Integration Approach	Augment existing tools, use APIs, and gradual automation with guardrails
Case Study Impact	60% outage reduction, 70% faster triage, 90% auto-remediation success, 25% cost savings

Table 1: Key Elements of ML-Enhanced SRE Implementation [7, 8]

A representative case study from a mid-sized software-as-a-service platform illustrates the transformative potential of ML-enhanced reliability practices across the incident lifecycle. This multi-tenant platform, serving customers across financial services, healthcare, and retail sectors, implemented a comprehensive ML strategy to address growing operational complexity as tenant count and feature scope expanded. The implementation journey spanned multiple phases, beginning with consolidated observability across previously siloed monitoring systems. This foundation enabled the development of tenant-specific baseline models that captured normal operational patterns across diverse workload types. The ML implementation architecture consisted of four primary components: predictive analytics for early warning, intelligent classification for incident triage, guided remediation for operator assistance, and resource optimization for infrastructure efficiency. The predictive analytics system employed an ensemble approach combining statistical methods with deep learning models, while the classification system utilized natural language processing techniques to extract patterns from unstructured logs and incident reports. The remediation component incorporated reinforcement learning to suggest optimal recovery actions based on historical effectiveness, while the resource optimization module leveraged forecasting models to predict tenant-specific demand patterns and optimize infrastructure allocation. Beyond technical improvements, the organization reported significant cultural transformation as reliability practices evolved from reactive firefighting toward proactive, data-driven operations. This shift required substantial investment in skills development, with SRE teams developing new competencies in data analysis, model evaluation, and machine learning operations alongside traditional reliability engineering practices [8].

5. Challenges and Future Directions

Despite promising results, ML-augmented SRE faces significant challenges. Model drift requires regular retraining as workloads evolve. Black-box ML systems must be made interpretable to support root cause analysis and regulatory compliance. High-quality incident datasets remain scarce in production environments. Ensuring fairness across tenant tiers presents both technical and ethical considerations. Looking forward, we anticipate several developments: large language models generating natural language incident reports, federated learning preserving tenant data privacy, graph neural networks mapping dependency topologies, and AI-orchestrated failover mechanisms across multiple clusters.

Model drift emerges as a fundamental challenge in operational machine learning systems deployed for reliability engineering. In dynamic production environments, the statistical properties of telemetry data evolve continuously as application workloads change, infrastructure components are updated, and user behavior patterns shift. This evolution causes the performance of previously trained models to degrade over time as the underlying data distribution diverges from the training set. The phenomenon manifests particularly acutely in multi-tenant environments, where each tenant's workload may evolve independently, creating a complex landscape of shifting patterns that models must accommodate. Addressing model drift requires implementing sophisticated monitoring frameworks that continuously evaluate prediction quality against ground truth observations. Effective approaches incorporate automated drift detection mechanisms that analyze feature distributions and model performance metrics to identify when retraining becomes necessary. Organizations implementing ML-driven reliability systems must establish appropriate governance processes around model maintenance, including clear ownership definitions, validation procedures, and deployment pipelines that facilitate regular updates. Beyond reactive approaches, advanced implementations incorporate adaptive learning techniques that allow models to incrementally adjust to changing conditions without complete retraining. The challenge extends beyond technical considerations to operational processes, requiring close collaboration between data science teams developing models and reliability engineers responsible for production systems [9].

The interpretability of machine learning models represents a critical concern for reliability engineering applications, where understanding failure modes and root causes remains essential for effective incident management. Complex model architectures, particularly deep neural networks, often function as black boxes whose internal decision processes defy straightforward explanation. This opacity creates practical challenges during incident response, where operators need to understand not just what anomalies have been detected but why specific patterns triggered alerts and how they relate to underlying system components. The interpretability challenge extends to stakeholder communication, where technical and business leaders require clear explanations of automated decisions affecting service availability and performance. Post-incident analysis and continuous improvement processes similarly depend on understanding model behavior to identify potential weaknesses and refinement opportunities. Regulatory considerations further amplify the importance of interpretability, particularly in industries with compliance requirements around incident documentation and risk management. Several approaches show promise in addressing these challenges, including inherently interpretable model architectures, post-hoc explanation techniques that analyze model outputs, and hybrid systems that combine complex predictive components with explainable decision layers. The most effective implementations balance predictive performance with appropriate transparency, recognizing that complete explainability may sometimes require sacrificing some degree of model sophistication [9].

Data quality and availability present persistent challenges for machine learning applications in reliability engineering, particularly regarding labeled examples needed for supervised learning approaches. Incident data tends to be relatively sparse in well-managed production environments, creating a fundamental tension: the more successful reliability practices become at preventing incidents, the less training data becomes available for improving predictive models. This challenge intensifies for high-severity incidents, which occur infrequently but represent the most critical detection targets. The data scarcity problem extends beyond raw quantity to quality considerations, as incident documentation practices vary significantly in structure, completeness, and accuracy. Temporal gaps in historical data, inconsistent labeling practices, and limited contextual information further complicate model development efforts. Feature engineering presents additional challenges, requiring domain expertise to transform raw telemetry into meaningful inputs for machine learning algorithms. Several approaches help address these limitations, including synthetic data generation through simulation, transfer learning from adjacent domains, and semi-supervised techniques that leverage abundant unlabeled data supplemented by limited labeled examples. Data augmentation strategies, including time-series transformations and controlled perturbations, can expand limited training sets while improving model robustness. Beyond technical solutions, establishing rigorous data governance practices and standardized incident documentation workflows significantly enhances data quality for machine learning applications [9].

Fairness and equity across tenant tiers present complex challenges in multi-tenant reliability systems, where machine learning models must balance competing priorities and potentially conflicting service-level agreements. The fundamental challenge stems from potential biases in model training and operation that could disadvantage certain tenant segments, particularly smaller customers generating less telemetry

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

data or exhibiting atypical usage patterns. These biases can manifest in various ways, including differential accuracy in anomaly detection, varying lead times for predictive alerts, and inconsistent effectiveness of automated remediation actions. The challenge extends beyond technical considerations to ethical dimensions, raising questions about equitable resource allocation and appropriate service differentiation between tenant tiers. Addressing these concerns requires deliberate design choices throughout the machine learning lifecycle, from data collection and preprocessing through model selection, training, evaluation, and deployment. Effective approaches incorporate fairness metrics into evaluation frameworks, explicitly measuring performance across tenant segments to identify and mitigate potential disparities. Techniques including stratified sampling, tenant-specific feature normalization, and multi-task learning help ensure equitable treatment while respecting intentional service differentiation encoded in business agreements. The most sophisticated implementations employ constrained optimization approaches that explicitly model fairness requirements alongside performance objectives, allowing principled trade-offs when resources become constrained [10].

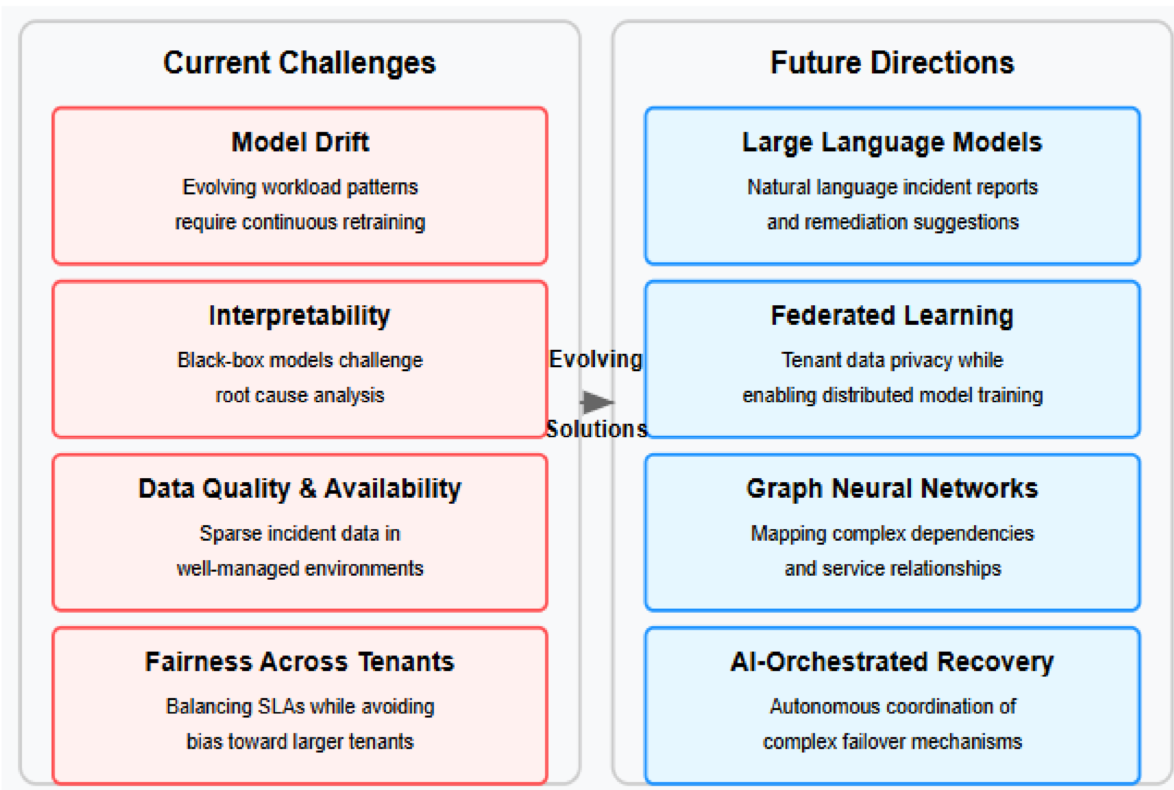

Fig 3: Challenges and Future Directions [9, 10]

The future of ML-augmented reliability engineering points toward several promising developments that could significantly advance current capabilities. Large language models demonstrate remarkable potential for enhancing incident management workflows, from generating structured documentation to synthesizing knowledge from historical incidents and suggesting potential remediation strategies. These models can transform unstructured operational data into actionable insights, significantly reducing the cognitive load on human operators during complex incidents. Federated learning approaches offer a path toward addressing data privacy concerns in multi-tenant environments, enabling models to learn from distributed data sources without centralizing sensitive customer information. This capability proves particularly valuable for global platforms operating across diverse regulatory jurisdictions with varying data protection requirements. Graph neural networks represent another promising direction, offering sophisticated modeling of complex system dependencies that characterize modern distributed architectures. These models excel at capturing the intricate relationships between microservices, infrastructure components, and tenant workloads, enabling more accurate fault localization and impact prediction than traditional approaches. AI-orchestrated recovery mechanisms represent perhaps the most ambitious frontier, moving beyond detection and diagnosis toward autonomous remediation of complex failure scenarios. These systems could eventually coordinate sophisticated recovery operations across multiple clusters, availability zones, and cloud providers, dynamically adapting strategies based on evolving conditions while maintaining appropriate human oversight [10].

Conclusion

Machine learning is fundamentally transforming SRE practices in multi-tenant cloud-native systems. The integration of AI throughout the reliability lifecycle, from proactive failure prediction to automated remediation and resource optimization, provides a scalable foundation for resilient platforms. ML-enhanced SRE pipelines can significantly reduce operational toil, improve system uptime, and deliver consistent experiences across diverse tenant populations. As these technologies mature and adoption increases, organizations operating multi-tenant platforms will need to develop new competencies at the intersection of reliability engineering, data science, and cloud-native architecture. The future of SRE lies in this convergence, enabling autonomous, self-healing systems that maintain resilience at unprecedented scale.

References

[1] Sibaram Prasad Panda et al., "Benefits of Site Reliability Engineering (SRE) in Modern Technology Environments," SSRN, 2025. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5285768

[2] Phanish Lakarasu, "Designing Cloud-Native AI Infrastructure: A Framework for High-Performance, Fault-Tolerant, and Compliant Machine Learning Pipelines," SSRN, 2025. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5237913

[3] Hilkka Woods et al., "Multi-Tenant Architecture Design in Cloud-Native Applications," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/392163585_Multi-Tenant_Architecture_Design_in_Cloud-Native_Applications

[4] Harshkumar R. Sharma et al., "Design and Evaluation of a Multi-Tenant SaaS Platform for Scalable and Secure Application Delivery," Indian Journal of Computer Science and Technology, 2025. [Online]. Available: https://www.indjcst.com/archiver/archives/design_and_evaluation_of_a_multi_tenant_saas_platform_for_scalable_and_secure_application_delivery.pdf

[5] Ravikumar Perumallaplli, "Predictive Maintenance in Cloud Infrastructure: A Machine Learning Framework," SSRN, 2025. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5228213

[6] Naga Sai Bandhavi Sakhamuri, "Autonomous SRE: A Reinforcement Learning Approach to Proactive Incident Prevention in Cloud-Native Environments," Journal of Computer Science and Technology Studies, 2025. [Online]. Available: https://al-kindipublishers.org/index.php/jcsts/article/view/9843

[7] Hironori Washizaki et al., "Machine Learning Architecture and Design Patterns," IEEE, 2020. [Online]. Available: http://www.washi.cs.waseda.ac.jp/wp-content/uploads/2019/12/IEEE_Software_19__ML_Patterns.pdf

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[8] Maheeza Bhamidipati, "AI-Driven Automation and Reliability Engineering: Optimizing Cloud Infrastructure for Zero Downtime and Scalable Performance," Journal of Computer Science and Technology Studies, 2025. [Online]. Available: https://al-kindipublishers.org/index.php/jcsts/article/view/9725
[9] Zhaoyi Xu and Joseph Homer Saleh, "Machine learning for reliability engineering and safety applications: Review of current status and future opportunities," ScienceDirect, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0951832021000892
[10] Saba Zia et al., "The Role of Artificial Intelligence in Modern Engineering: Opportunities and Challenges," Research Corridor Journal of Engineering Science, 2024. [Online]. Available: https://researchcorridor.org/index.php/RCJES/article/view/5