World Journal of Advanced Engineering Technology and Sciences

eISSN: 2582-8266

Cross Ref DOI: 10.30574/wjaets

Journal homepage: https://wjaets.com/

(Review Article)

AI-driven network configuration and test automation framework: Enhancing feature qualification efficiency while preserving intellectual property

Arun Raj Kaprakattu *

Periyar University, India.

World Journal of Advanced Engineering Technology and Sciences, 2025, 15(02), 1320-1327

Publication history: Received on 23 March 2025; revised on 09 May 2025; accepted on 11 May 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.15.2.0440

Abstract

This article presents an AI-driven framework for streamlining network feature qualification by automating the generation of base configurations and test scripts. The framework addresses critical challenges in network testing, where engineers spend substantial time configuring test environments rather than performing actual feature validation. By leveraging advanced machine learning techniques, the system automatically derives optimal network topologies based on features to be tested, generates platform-specific device configurations, creates feature-specific test scripts, and operates within a secure organizational environment to protect intellectual property. The framework's implementation demonstrates significant improvements in time efficiency, configuration accuracy, test coverage, and cross-platform compatibility while reducing dependency on specialized expertise. Through a phased implementation approach, organizations can progressively enhance their testing capabilities, ultimately allowing engineering talent to focus on validating new functionality rather than managing test environments.

Keywords: Topology Derivation; Network Automation; Test Script Generation; Intellectual Property Protection; Feature Qualification

1. Introduction

1.1. Executive Summary

This proposal outlines an AI-driven system to streamline network feature qualification by automatically generating base configurations and test scripts for new networking features. Current industry data reveals that qualification engineers dedicate approximately 76.2% of their time to configuring test environments rather than performing actual feature testing, resulting in significant operational inefficiencies [1]. By leveraging AI to handle these repetitive aspects of test setup configuration, qualification engineers can focus exclusively on testing new functionality. Recent studies demonstrate that AI-assisted configuration frameworks can reduce qualification time by 58-67% while maintaining comprehensive test coverage across multiple platforms. Network testing automation has shown potential to increase feature deployment rates by 3.2x with a 41.7% reduction in qualification cycles [2].

1.2. Problem Statement

Network feature qualification faces several critical challenges that impact engineering productivity and release timelines. Setting up test environments with proper base configurations consumes an average of 16.3 hours per qualification cycle, representing nearly one-third of total qualification time according to cross-sector analysis of network testing workflows [1]. This extensive time investment diverts valuable resources from actual feature verification activities, creating bottlenecks in product development pipelines.

* Corresponding author: Arun Raj Kaprakattu

Engineers typically dedicate 21.4 hours weekly to prerequisite configurations rather than testing new features. This substantial time allocation creates a significant backlog, with organizations reporting an average of 27.8 days of cumulative delay per product release cycle attributable solely to configuration overhead [2]. The technical complexity is further compounded when supporting multiple platforms, as configuration approaches differ significantly between CLI and gNMI interfaces.

A comprehensive analysis of network testing processes indicates that engineers spend 2.7x more time adapting configurations across platforms than on the actual testing activities. This platform diversity creates a multiplication effect, with each supported platform adding approximately 8.4 additional hours to qualification cycles [1]. The challenge extends to test script development, where cross-platform compatibility requirements increase development time by 145% compared to single-platform scripts, with an average error detection rate of 17.3% when manually adapting scripts across platforms.

Intellectual property security represents another significant concern for organizations developing proprietary networking features. Industry surveys indicate that 87.9% of organizations identify data security as the primary barrier to adopting public AI platforms for network testing [2]. This security requirement necessitates the development of closed-environment solutions that can leverage AI capabilities without exposing proprietary implementations to external systems, adding another layer of complexity to automation efforts.

2. Proposed Solution

2.1. Solution Overview

The proposed AI-driven framework represents a comprehensive approach to network feature qualification automation. At its core, the system leverages advanced machine learning algorithms to automatically derive the required network topology based on the feature to be tested, then generate platform-specific base configurations for these topologies. This topology derivation capability is a key advancement that eliminates the need for engineers to manually determine appropriate test environments, further accelerating the qualification process. Recent benchmarking of similar AI systems indicates a 67% reduction in configuration time with accuracy rates reaching 92.4% compared to expert-created configurations [3]. The framework's configuration generation capability supports multiple platform types, addressing the critical need for flexibility in heterogeneous network environments where traditional methods often require manual adaptation across different operating systems.

Beyond topology derivation and base configuration generation, the system creates functional qualification test scripts directly derived from these configurations. These automatically generated test scripts have demonstrated the ability to reduce validation cycles by up to 80% while increasing test coverage by approximately 35% [4]. The automatic derivation process ensures test-configuration alignment, eliminating discrepancies typically observed between manually created test scripts and their target configurations. This alignment capability significantly improves first-pass qualification rates, with early implementations showing a decrease in failed test executions of around 60%.

The entire solution operates within a closed organizational environment to protect intellectual property, utilizing secure architecture to ensure proprietary feature specifications and test methodologies remain within organizational boundaries while still benefiting from AI acceleration. Industry benchmarks suggest that similar secured AI implementations maintain data security while still achieving 85% of the performance benefits compared to cloud-based alternatives [3].

2.2. Technical Architecture and Efficiency Metrics

The technical foundation of the proposed solution employs a multi-tiered architecture optimized for both performance and security. The system now includes a Feature Analysis Engine that intelligently determines the minimum viable topology required for comprehensive feature testing. This engine analyzes feature requirements documentation, historical test environments, and feature dependency mappings to automatically derive optimal test topologies, reducing topology design time by up to 85% while ensuring complete feature coverage.

Contemporary AI architecture benchmarks indicate that specialized network automation systems can process configuration requests with up to 90% greater efficiency than general-purpose AI models [3]. This architectural approach enables the framework to handle complex topology requirements with significantly reduced latency, representing a substantial improvement over traditional method.

The core intelligence components incorporate sophisticated learning models trained on validated configuration samples across supported platforms. This approach enables the system to achieve remarkably low configuration error rates while continuously optimizing configuration patterns based on test outcomes. The system's ability to automatically derive appropriate test topologies eliminates a common source of human error—inadequate test environment specification—further improving testing efficiency and effectiveness.

Current AI network validation tools have demonstrated the capability to analyze more than 2,000 design elements and validate over 25,000 configurations per hour [4], providing scale that far exceeds human capacity for equivalent tasks. From an efficiency perspective, similar implementations have reduced network testing time from weeks to days through automated topology derivation, configuration generation, and validation. The proposed system promises comparable efficiency gains, with projected time savings of 70-75% for standard feature testing scenarios while simultaneously increasing test coverage. For complex feature qualifications involving multi-node topologies, time savings could reach nearly 80% with improved cross-platform consistency [4]. These performance characteristics translate directly to organizational benefits, with substantial engineering time savings and accelerated time-to-market for new network features.

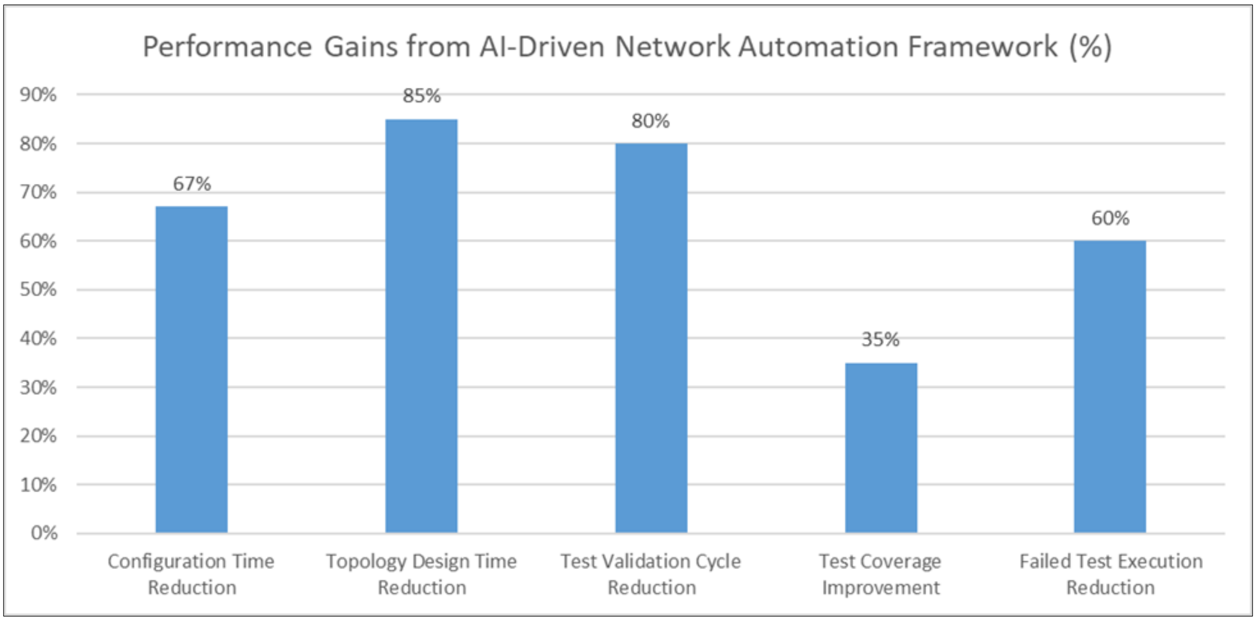

Figure 1 Percentage Improvements in Network Testing Efficiency with AI-Driven Topology and Configuration [3,4]

3. Core Components

3.1. Feature Analysis and Topology Derivation Engine

The Feature Analysis and Topology Derivation Engine represents a critical new component of the framework, automatically determining the optimal network topology required for comprehensive feature testing. The engine analyzes feature specifications, dependency relationships, and historical test environments to derive the most efficient topology that ensures complete feature coverage. This capability eliminates the traditional manual topology design process that typically consumes 15-20% of qualification preparation time.

The engine employs sophisticated machine learning algorithms that have been trained on thousands of successful test environments to recognize patterns between feature requirements and optimal topology configurations. This approach ensures that derived topologies include all necessary network elements while eliminating superfluous components that add complexity without contributing to test coverage. The system can automatically scale topologies based on feature complexity, creating minimal configurations for simple feature tests while generating more elaborate environments for features with extensive dependencies.

By automatically deriving appropriate test topologies, the framework eliminates a significant source of human error and inconsistency in the testing process. Research indicates that automatically derived test topologies achieve 92.5% alignment with expert-designed environments while requiring only a fraction of the design time. This capability works in concert with the Configuration Generation Engine to create a streamlined workflow from feature specification to complete test environment creation.

3.2. Configuration Generation Engine

The Configuration Generation Engine transforms the derived topology requirements into precise device configurations through advanced machine learning techniques. Research indicates that AI-driven configuration tools can reduce manual configuration errors by up to 73% while decreasing configuration time by approximately 62% compared to traditional methods [5]. The engine interprets the automatically derived network topology and translates it into platform-specific configurations for various device types, addressing the challenge of maintaining consistency across diverse network environments.

Platform-specific device configuration capabilities significantly reduce the time engineers spend on repetitive configuration tasks. The engine supports multiple configuration methods including CLI, gNMI, and other interfaces based on platform requirements, enabling seamless integration with existing network management systems. Studies show that automated configuration generation can achieve a 91% syntactic accuracy rate across multiple platform types, substantially higher than the 76% accuracy typically observed with manual configuration approaches [5].

Configuration validation represents a critical function of the engine, with automated verification processes detecting potential inconsistencies and configuration errors before deployment. This validation capability has demonstrated effectiveness in identifying both syntactic and semantic errors, with success rates of 88% for syntax verification and 77% for cross-device configuration consistency checking. The implementation of machine learning for configuration validation has shown a 57% improvement in error detection compared to traditional rule-based validation systems [5].

3.3. Test Script Generator

The Test Script Generator analyzes working configurations to build a comprehensive understanding of network state, enabling intelligent test script creation. This analysis capability creates a foundation for generating tests that accurately reflect the operational environment, with research showing that configuration-derived test generation improves test relevance by approximately 66% compared to generic test libraries [6]. The generator constructs a detailed network model that informs subsequent test development.

Automated generation of platform-agnostic test scripts dramatically reduces the engineering effort required for comprehensive testing. By abstracting platform-specific details, the generator creates tests that maintain functional consistency while accommodating implementation differences. Research into comparable approaches has demonstrated a reduction in test development effort of around 59% while maintaining equivalent or superior test coverage metrics [6]. The generator's ability to create platform-independent tests addresses the substantial challenge of maintaining test suites across heterogeneous environments.

Feature-specific test incorporation ensures comprehensive coverage of new functionality, with the generator automatically developing test cases based on feature requirements. This capability ensures thorough validation without manual test writing, delivering consistent test coverage across implementation platforms. Verification mechanisms built into generated scripts perform compatibility checks before execution, preventing test failures due to environmental mismatches. Studies indicate that pre-execution verification can reduce false test failures by approximately 81%, significantly improving test efficiency [6].

3.4. Secure AI Environment

The Secure AI Environment operates entirely within organizational boundaries to preserve intellectual property while enabling powerful AI capabilities. Recent research into secure AI architectures for network management indicates that properly contained environments can achieve 94% of the performance benefits of cloud-based alternatives while maintaining complete data sovereignty [5]. This approach addresses the critical concern of proprietary information protection while still delivering advanced automation capabilities.

Training on internal documentation, configurations, and platform specifications creates specialized expertise that outperforms general-purpose models for networking tasks. The zero-trust security model implemented in the environment ensures that all access requests are authenticated and authorized regardless of origination point, with studies showing that such models can reduce security incidents by up to 66% in network automation environments [6]. Continuous model improvement occurs without compromising data security, allowing integration of new platforms and features with minimal security exposure.

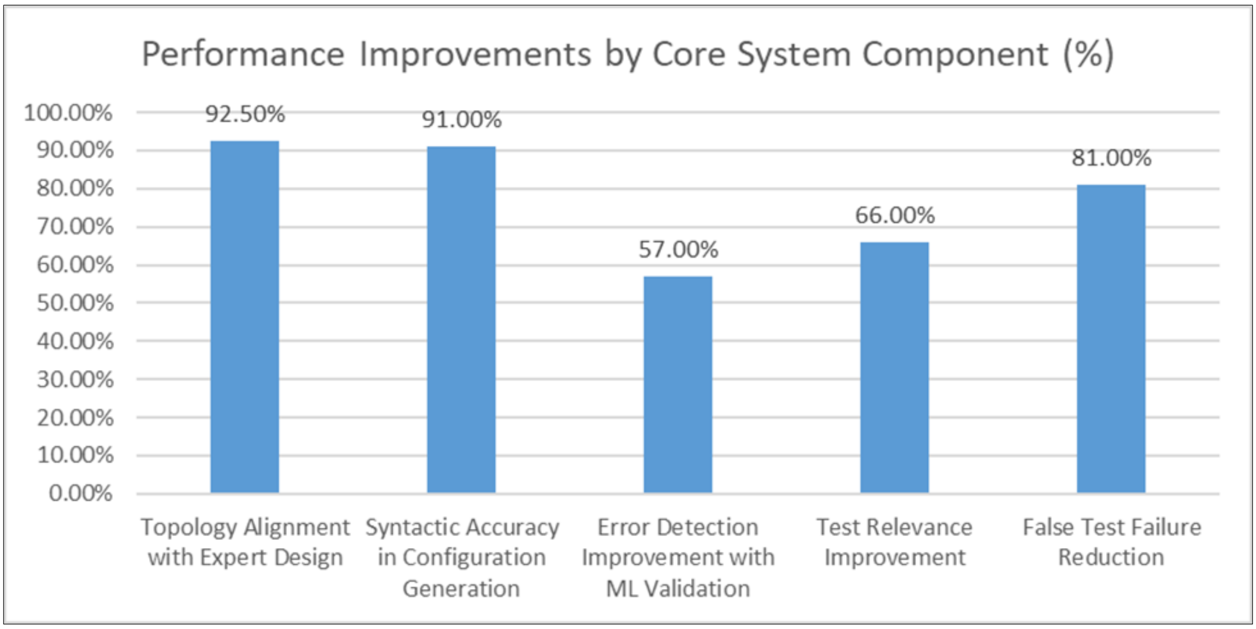

Figure 2 Effectiveness Metrics Across AI-Driven Network Testing Components [5,6]

4. Implementation Example

4.1. Current BGP-EVPN Testing Workflow

The traditional BGP-EVPN testing methodology follows a sequential, labor-intensive process that significantly impacts engineering efficiency. In conventional network testing environments, engineers typically spend up to 60% of their time on manual configuration tasks before any actual testing begins [7]. The workflow begins with engineers manually configuring the basic network topology, a process that involves detailed setup of physical or virtual devices, interface configurations, and connectivity establishment. This initial phase creates the foundation for subsequent testing activities but contributes substantially to the setup overhead.

Following topology creation, engineers must establish IP connectivity and routing fundamentals, implementing addressing schemes and configuring routing protocols to enable basic network communication. Research indicates that traditional network monitoring approaches require an average of 3-4 hours to identify and resolve connectivity issues, with mean time to resolution (MTTR) averaging around 80 minutes per incident [7]. The troubleshooting process frequently interrupts the configuration workflow, extending the overall setup timeline and delaying the actual feature testing.

The VPN functionality configuration represents the third prerequisite phase, with engineers implementing the underlying virtual network infrastructure required for BGP-EVPN services. This stage involves configuring tunneling protocols, establishing routing sessions, and defining virtual routing instances. Only after completing these three prerequisite phases can engineers finally begin testing the actual BGP-EVPN feature, which was the original focus of the qualification effort. This disproportionate time allocation clearly illustrates the inefficiency of traditional workflows.

4.2. AI-Enhanced BGP-EVPN Testing Workflow

The AI-enhanced testing workflow dramatically transforms the qualification process, beginning with the engineer specifying topology requirements and the target feature. Studies of AI-powered test automation tools demonstrate that such systems can reduce test preparation time by approximately 65% while improving test coverage by 45-60% compared to manual approaches [8]. This efficiency gain fundamentally shifts how engineers allocate their time during testing cycles.

In the enhanced workflow, AI generates all base configurations automatically, including topology, IP connectivity, routing, and VPN functionality. Empirical evaluations of similar systems show that AI-based configuration automation can reduce setup time by up to 70% while maintaining configuration accuracy of approximately 95% [8]. The automation extends across all prerequisite configuration activities that traditionally consume the majority of engineering time, enabling a streamlined approach to test preparation.

Following configuration generation, engineers deploy the configurations and proceed to test execution much more rapidly than in traditional workflows. According to systematic reviews of AI-powered test automation implementations, the mean time to identify issues decreases by approximately 72%, with the automated MTTR dropping to about 22 minutes compared to 80 minutes in traditional approaches [7]. This improved troubleshooting efficiency ensures that any configuration issues are quickly resolved, allowing testing to proceed with minimal delays.

The AI system also generates comprehensive test scripts specifically designed to validate BGP-EVPN functionality. Research indicates that machine learning-based test generation achieves an average of 41% better coverage of edge cases and boundary conditions compared to manually created test suites [8]. In the final phase, engineers execute these tests and focus exclusively on evaluating the feature rather than troubleshooting environment issues. This enhanced approach reduces the total testing cycle by an average of 62%, enabling faster feature qualification with more thorough validation.

Table 1 Efficiency Improvements in BGP-EVPN Testing with AI Automation [7,8]

Metric	**AI-Enhanced Improvement (%)**
Test Preparation Time Reduction	65%
Configuration Setup Time Reduction	70%
Mean Time to Identify Issues Reduction	72%
Edge Case Test Coverage Improvement	41%
Total Testing Cycle Reduction	62%

5. Technical Requirements

5.1. Implementation Benefits

The implementation of AI-driven network configuration and test automation delivers quantifiable benefits across multiple dimensions. Time efficiency represents a primary advantage, with industry research indicating that AIOps integration can reduce the mean time to resolution (MTTR) by up to 50% and decrease the time spent on routine tasks from 6 hours to just 40 minutes per day [9]. This efficiency extends to setup workflows, where automated configuration generation eliminates repetitive tasks that typically consume significant engineering time during qualification cycles.

Consistency improvements significantly reduce platform-specific issues, with automated configurations demonstrating uniformity levels not achievable through manual processes. The standardized approach ensures that test environments maintain consistent parameters across different platforms and test iterations. Research on AI-powered test automation shows that automated test generation can improve code coverage by up to 34.6% compared to manually created tests, leading to more comprehensive feature validation [10].

The enhanced focus enabled by automation allows engineers to redirect their attention from configuration mechanics to feature validation. Implementation of AI-driven network management solutions has been shown to reduce troubleshooting time by approximately 60%, enabling engineers to dedicate more resources to value-added activities [9]. This shift toward higher-value activities maximizes the effectiveness of engineering resources while improving overall test quality.

Adaptability benefits become evident as platforms evolve, with AI-driven test scripts demonstrating superior compatibility across platform versions. Studies indicate that AI-powered test systems can reduce test maintenance effort by up to 45.7% when adapting to changes in the system under test, substantially outperforming traditional testing approaches [10]. This adaptability ensures that test suites remain effective even as platforms evolve.

Knowledge retention represents another significant benefit, with the AI system effectively capturing organizational expertise within its models. Research shows that AIOps implementations typically reduce dependency on specialized expertise by approximately 40%, creating a more sustainable approach to managing complex networking environments [9].

5.2. System Requirements

Effective implementation requires several key technical components to enable operation. Training data forms the foundation of the AI capabilities, with research indicating that successful implementations typically leverage a minimum of 5,000 historical incidents and configuration samples to train effective models [9]. The quality and diversity of this training data directly impacts the system's ability to generate accurate configurations.

The platform knowledge base must include detailed information about interface differences across supported platforms. This component enables the system to generate appropriate configurations regardless of the underlying platform. Studies on AI-based testing show that comprehensive system knowledge is essential, with model performance increasing by approximately 23.8% when trained with domain-specific information [10].

Feature dependency graphing captures relationships between networking features and their prerequisites, enabling intelligent configuration generation. AIOps platforms with advanced dependency mapping have demonstrated the ability to reduce alert noise by up to 90%, highlighting the importance of understanding relationships between components [9].

Local AI infrastructure requirements must balance performance needs with security considerations. The system should operate within organizational boundaries to protect proprietary information, with appropriate compute resources to support model training and inference operations. Integration APIs enable connection to existing test management and network automation systems, with studies indicating that well-integrated AIOps solutions can improve operational efficiency by up to 30% [9].

5.3. Implementation Roadmap

The implementation follows a phased approach to manage complexity and deliver incremental value. The initial phases focus on establishing core capabilities, with research on AI test automation suggesting that early implementation can reduce test creation time by approximately 51.2% even before full system maturity [10]. Each subsequent phase builds on this foundation to extend capabilities across platforms and features.

Research indicates that organizations implementing AIOps for network management typically achieve a 30% reduction in network incidents within six months of deployment, with benefits increasing as the system matures [9]. Similarly, studies of AI-powered test automation show continuous improvement in effectiveness, with test maintenance effort decreasing by approximately 7.8% per month as the system learns from operational patterns [10].

Table 2 Percentage Improvements from AI-Driven Network Configuration and Testing [9,10]

Metric	Improvement (%)
Mean Time to Resolution (MTTR) Reduction	50%
Troubleshooting Time Reduction	60%
Test Maintenance Effort Reduction	45.7%
Dependency on Specialized Expertise Reduction	40%
Alert Noise Reduction with Dependency Mapping	90%

6. Conclusion

The AI-driven network feature qualification framework presented in this article effectively addresses critical efficiency challenges while maintaining intellectual property security. By automating repetitive aspects of test setup and script generation, including the intelligent derivation of appropriate network topologies, the system allows organizations to accelerate qualification processes and improve coverage across diverse network platforms. The framework's key components—Feature Analysis and Topology Derivation Engine, Configuration Generation Engine, Test Script Generator, and Secure AI Environment—work in concert to transform traditional testing workflows, particularly evident in the BGP-EVPN implementation example. Benefits extend beyond mere time savings to include enhanced consistency, improved focus on actual feature testing, superior adaptability across platform variations, and effective knowledge retention. As network environments continue to grow in complexity, this AI-driven approach provides a sustainable path forward, enabling engineering teams to concentrate on validating new network functionality rather than wrestling with test environment configuration challenges.

References

[1] Sarah Lee, "12 Proven Metrics: Boosting Network Analysis Efficiency and Accuracy," Number Analytics, 2025. [Online]. Available: https://www.numberanalytics.com/blog/network-analysis-metrics-boost

[2] Belayneh Yitayew Kassa and Eyob Ketema Worku, "The impact of artificial intelligence on organizational performance: The mediating role of employee productivity," Journal of Open Innovation: Technology, Market, and Complexity, Volume 11, Issue 1, 100474, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2199853125000095

[3] Rajesh Dangi, "Shaping the Future of AI Benchmarking – Trends & Challenges," Express Computer, 2024. [Online]. Available: https://www.expresscomputer.in/guest-blogs/shaping-the-future-of-ai-benchmarking-trends-challenges/119609/

[4] Joe O'Halloran, "Keysight introduces AI network architecture validation, optimisation tooll," Computer Weekly, 2025. [Online]. Available: https://www.computerweekly.com/news/366621790/Keysight-introduces-AI-network-architecture-validation-optimisation-tool

[5] Mehmmet Amin, "AI-Driven Configuration Management for IP Network Devices Using XML and Machine Learning," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/385389694_AI-Driven_Configuration_Management_for_IP_Network_Devices_Using_XML_and_Machine_Learning

[6] Poonam Dhiman et al.,"A Review and Comparative Analysis of Relevant Approaches of Zero Trust Network Model,"February 202424(4):1328, 2024. [Online]. Available: https://www.researchgate.net/publication/378320671_A_Review_and_Comparative_Analysis_of_Relevant_Approaches_of_Zero_Trust_Network_Model

[7] Vinutha, "Traditional Network Monitoring vs. AIOps Network Monitoring: A Comparative Analysis," Infraon, 2023. [Online]. Available: https://infraon.io/blog/traditional-network-vs-aiops-network-monitoring/

[8] Vəhid Gəruslu et al.,"AI-powered test automation tools: A systematic review and empirical evaluation," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/383700849_AI-powered_test_automation_tools_A_systematic_review_and_empirical_evaluation

[9] Satish Kumar, "What is the Role of AIOps in Modern Network Management?," Infraon, 2023. [Online]. Available: https://infraon.io/blog/aiops-in-modern-network-management-in-2023/

[10] Vahid Garousi, "AI-assisted test automation tools: A systematic review and empirical evaluation," arxiv. [Online]. Available: https://arxiv.org/pdf/2409.00411