International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 9643-9652
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

AI-Native Network Validation for Next-Generation Data Center Switches

Srinivas Yadam*

Independent Researcher, USA
* **Corresponding Author Email:** yadamsrinivasinf@gmail.com- **ORCID:** 0000-0002-1147-0050

Article Info:

DOI: 10.22399/ijcesen.4513
Received : 21 October 2025
Revised : 10 December 2025
Accepted : 15 December 2025

Keywords

AI-Native Network Validation,
Reinforcement Learning Test Orchestration,
Programmable Data Plane Validation,
Digital Twin Network Simulation,
Autonomous Defect Detection

Abstract:

Modern data center architectures have evolved into AI-native computational fabrics operating at terabit speeds with programmable data planes, rendering traditional static network validation methodologies inadequate for ensuring operational reliability. AI-Native Network Validation represents a paradigm shift from passive test execution to active cognitive ecosystems that autonomously adapt validation strategies through reinforcement learning, real-time telemetry analysis, and digital twin synthesis. The architecture integrates adaptive test scheduling engines that prioritize scenarios based on historical defect patterns, multi-layer telemetry ingestion capturing packet-level behavior across distributed infrastructure, and causal graph analytics enabling automated root cause isolation. The operational pipeline employs reinforcement learning agents to continuously generate optimized test vectors while graph neural networks encode complex topological dependencies for intelligent scenario selection. The uses of high-density AI fabrics with high-port speeds, clusters of edge computing units that need permanent validation, and disaggregated composable architectures with new configuration complexity. Future directions include self-learning validation ecosystems that are automatically verified, cross-vendor requirements that comply with standards, and federated learning schemes that allow sharing of knowledge across organizations without losing the subjects of the organization.

1. Introduction and Problem Context

Modern data center architectures have moved from traditional three-tier designs to AI-native computational fabrics that emphasize horizontal scalability, disaggregated resource pooling, and autonomous orchestration. Ultra-high-speed interconnects continue to be featured in contemporary deployments, where the industry is actively investigating ways beyond current Ethernet specifications, driven by insatiable bandwidth requirements for artificial intelligence and machine learning workloads. The IEEE 802.3 Beyond 400 Gb/s Ethernet Study Group has systematically evaluated technical feasibility, market requirements, and implementation challenges associated with next-generation Ethernet speeds, recognizing the need for data center fabrics to evolve in support of increasingly dense computational clusters and memory-disaggregated architectures [1]. These infrastructures employ advanced leaf-spine topologies featuring programmable data planes with dynamic traffic steering capabilities, in-network computing capabilities, and real-time configuration. Flexibility to support changing computational demands is required. The architectural shift well surpasses any simple increase in bandwidth; it captures the fundamental way network resources are provisioned, monitored, and validated throughout their operational lifecycle.

Traditional network validation methodologies based on static test scripts and predefined regression suites face severe limitations when confronted with the complexity and dynamism of modern AI-centric data centers. Conventional approaches typically rely on fixed test plans executed against predefined topologies, where test cases are handcrafted by validation engineers based on anticipated failure modes and protocol interactions. These static methodologies assume relatively stable network configurations and predictable traffic patterns—assumptions that no longer hold in environments characterized by continuous reconfiguration, adaptive routing algorithms, and workload-driven topology mutations. The rigidity of script-based testing frameworks becomes particularly problematic when validating programmable switching ASICs that support multiple forwarding paradigms, telemetry streaming protocols, and hardware-accelerated processing functions.

It is, however, becoming increasingly obvious that the maturing programmability in packet processing devices has increased the validation gap between what can be achieved with conventional testing methodologies and what is expected by modern data center fabrics. The development and use of programming languages for protocol-independent packet processors have radically changed the validation landscape by making network operators define custom forwarding behaviors, protocol parsing logic, and packet modification operations directly in data plane hardware. The P4 programming language, as discussed, has become exemplary of this paradigm shift: By providing a domain-specific language, programmers specify how packets are processed by network devices, effectively making the data plane programmable and protocol-agnostic [2]. This programmability introduces combinatorial complexity that static test generation cannot adequately address. Each forwarding program is unique and carries its own significant implementation burden to validate it in depth, covering various traffic patterns, different header combinations, and state transitions.

Self-adaptive intelligent validation systems capable of operating at infrastructure complexity are a paradigm shift in network assurance methodology. These systems should also be able to automatically create test cases based on observed patterns of failure, dynamically prioritize validation activity to high-risk configuration space, and continually update testing policies as network architecture varies. The adaptive validation systems can provide coverage levels and defect detection rates that are impossible to obtain with manual test design by providing real-time telemetry feedback, historical defect analysis, and predictive modeling.

2. AI-Native Network Validation Architecture

AI-Native Network Validation marks a paradigm shift from passive test execution to active cognitive validation ecosystems that dynamically adapt to the network conditions, learned failure patterns, and emerging anomalies. The conventional validation models are made up of inflexible implementers of pre-archived test scripts, completely oblivious of the current network situation, defect tendencies throughout history, or the dynamic nature of performance. Conversely, AI-native architectures create a continuous feedback and factualization pathway in which the results of validation tasks are used to modify further test plans, and self-enhancing systems are formed, which undergo successive enhancements, increasing coverage and defect-detection rates. This paradigm shift regards validation as an intelligent agent rather than a passive tool, one that reasons over network behavior, hypothesizes on failure modes, and autonomously designs experiments to verify or refute such hypotheses. All this brings together several layers of intelligence-from low-level anomaly detection in telemetry streams to high-level, strategic planning of test campaigns-into one cohesive system that mimics the adaptive characteristics of the modern, AI-driven network infrastructures themselves.

The basic architectural building blocks of AI-Native Network Validation systems involve several tightly coupled subsystems that together empower autonomous, adaptive testing capabilities. At its base lies a broad telemetry ingestion infrastructure capable of processing diverse data streams from packet-level traces to interface statistics, protocol state machines, buffer occupancy metrics, and environmental sensors. Feeding off this telemetry substrate, machine learning models are trained to identify deviations from expected behavior patterns, detect incipient failures before they manifest into complete service disruptions, and correlate seemingly unrelated anomalies across distributed network elements. Digital twin technology has emerged as a critical enabler for the safe exploration of potentially destructive test scenarios, as research has demonstrated that high-fidelity network simulations are able to accurately reproduce complex forwarding behaviors and failure modes observed in physical infrastructure [3]. These virtual copies allow the validation systems to perform the aggressive stress tests, pour synthetic failures, and experiment with edge cases without putting production hardware or disrupting operational networks at risk. The digital twin keeps pace with the physical network state, thus making the results of simulation remain relevant and applicable to real-world deployments as well as presenting a sandbox to experiment with hypotheses.

In the proposed design of closed-loop adaptive testing, real-time telemetry analysis is operated together with reinforcement learning algorithms to produce self-optimizing validation pipelines. RL has shown much potential in addressing various network optimization problems, from congestion control to resource allocation, and thus appears appropriate for the strategic problem at hand regarding test case selection and prioritization [4]. In this context, a reinforcement learning agent learns to optimize long-term defect discovery rates by exploring the huge space of possible test configurations, traffic patterns, and failure injection scenarios. The learning agent receives rewards for discovered defects based on their assessed severity and novelty, gradually developing policies that efficiently navigate toward high-value test cases and avoid redundant or low-yield validation activities. This approach addresses the inherent combinatorial explosion problem in comprehensive network validation, where exhaustive testing becomes computationally infeasible due to high network complexity. Explicit models of uncertainty are maintained by the reinforcement learning framework, hence balancing the exploitation of known high-value test scenarios against the exploration of yet uncharted configuration spaces, which may harbor undiscovered modes of failure.

The test orchestration engine is the central intelligence that coordinates all validation activities across distributed testbeds, physical infrastructure, and digital twin environments. This dynamically generates test scenarios by combining learned templates with novel parameter combinations, prioritizes execution based on predicted defect yield and available resources, and sequences tests to maximize information gain while respecting dependencies and prerequisites. The anomaly cognition layer performs continuous analysis of network telemetry and behavioral patterns for identified deviations that warrant investigation using ensemble methods that combine statistical process control, deep learning architectures, and domain-specific heuristics. These architectural components put together a validation system that acts autonomously, learns continuously, and adapts intelligently with the ever-evolving complexity of modern network infrastructures.

3. Technical Mechanisms and Implementation

Adaptive test scheduling is a cornerstone mechanism in the AI-Native Network Validation systems, which leverages historical defect patterns and real-time anomaly detection to perform dynamic prioritization of validation activities. The scheduling engine keeps a comprehensive knowledge base of previously discovered defects, categorized by network configuration, traffic characteristics, protocol interactions, and environmental conditions under which failures manifested. Machine learning models analyze this historical corpus to identify patterns that correlate specific test parameters with high defect discovery rates, enabling the predictive prioritization of test scenarios likely to expose latent failures. Real-time anomaly detection enhances this historical analysis through the continuous monitoring of live telemetry streams for deviations from established baselines, thus immediately triggering validation responses upon suspicious behaviors. This operates in dual mode: it balances strategic, data-driven test planning with reactive investigation of nascent issues and ensures both systematic coverage of known risk domains and agile response to unexpected network behaviors. The multi-objective optimization methods of scheduling algorithms include the maximization of the expected defect production, the minimization of the time of validation cycle, optimization of resource utilization across distributed testbeds, and dynamically changing priorities in response to the availability of new information as the test is proceeding.

Multi-layer telemetry ingestion is the basis of a sensory platform that enables intelligent validation decisions, and as such, it necessitates advanced infrastructure to gather, normalize, and estimate heterogeneous streams of data across multiple layers of network abstraction. The current data center networks produce telemetry on a scale and speed levels never seen before, like packet-level visibility, milliseconds-scale timestamps, and per-flow state tracking that introduce overwhelming volumes of data to the old paradigms of monitoring architectures. In-band network telemetry presents a disruptive methodology for achieving comprehensive insight into network behavior through the embedding of metadata directly within packet headers as they traverse switching infrastructure, thereby providing every network element with the capability of appending operational state information that can subsequently be extracted and processed by downstream analytics systems. This process allows the validation framework to capture detailed path information, queue depths at every hop, and cumulative latency measurements without the need for separate monitoring infrastructure or additional network bandwidth for out-of-band telemetry collection. The multivariate telemetry ingested in this pipeline is various, and just a few of the sources of telemetry data include interface counters tracking packet and byte volume, error statistics indicative of forwarding anomalies, protocol state machine transitions revealing control plane behaviors, and environmental sensors that monitor temperature and power consumption. Advanced stream processing frameworks implement real-time aggregation and correlation, together with anomaly detection across such diverse telemetry sources, identifying complex failure signatures that span several layers and manifest across distributed network elements.

Digital twin modeling of physical infrastructure at granular levels allows the safe exploration of potentially destructive test scenarios while maintaining a high-fidelity representation of actual network behaviors. The digital twin synthesizes detailed models of switching ASIC architectures, including packet processing pipelines, buffer hierarchies, forwarding table organizations, and hardware resource constraints that influence performance under load. In order to guarantee that the validation outcomes can be consistently applied to real-world situations, virtual simulation frameworks should mimic timing behavior, queueing behavior, and resource sharing behavior in physical hardware. Network function virtualization has become a paradigm that permits the realization of network functions in software on general-purpose hardware and not on specific physical appliances, and comprehensive surveys have been documented on the enabling technologies and the large research gaps that still need to be bridged before the realization of production-grade performance and reliability. The digital twin consumes constant state changes of the physical infrastructure and aligns configuration parameters, routing tables, and traffic profiles with the current state of production conditions to ensure simulation environments are reflective of the actual state of production conditions. This conservation supports hybrid validation, in which early discovery is done in simulated settings to determine the most interesting test vectors, and then specific hardware is experimented on to confirm the results and verify hardware-specific behaviors that cannot be perfectly modeled in simulation.

Causal graph analytics provide for sophisticated root cause analysis that traces failures across network layers, from observable symptoms down to underlying defects through complex dependency chains. These analytical frameworks create directed acyclic graphs that model causal relationships between network events, configuration parameters, and observed behaviors, allowing for automated inference of failure propagation pathways. Self-healing extends beyond simple error detection to include autonomous recovery actions such as test environment reconfiguration, state rollback to known-good configurations, and dynamic mutation of test parameters to isolate failure conditions. Recursive validation mechanisms detect anomalous behavior in validation infrastructure itself, invoking meta-level diagnostics that ensure the integrity of monitoring systems, telemetry pipelines, and analysis frameworks. Integration of synthetic workload replay with live anomaly testing enables comprehensive validation coverage by combining deterministic reproduction of known failure scenarios with the exploratory testing of novel conditions identified through real-time monitoring, with complete assurance that both regression prevention and proactive defect discovery objectives are met.

4. Operational Pipeline and Methodology

The AI-Native Network Validation operational pipeline initiates with a comprehensive bootstrapping process, leveraging both legacy failure data and design constraints to constitute the foundational knowledge base guiding subsequent validation activities. This initialization step aggregates defect repositories of past hardware generations, regression test results across multiple product cycles, and field failure reports detailing production incidents into a rich corpus of known failure modes and their associated signatures. Design constraint integration includes the specification of switching ASIC datasheets, protocol implementation requirements, performance benchmarking, and resource limitation boundaries that define the operating envelope within which network infrastructure needs to behave correctly. The bootstrap process adopts clustering algorithms that group historical failures into symptom pattern-based taxonomies, root cause classifications, and environmental conditions to enable pattern recognition models to identify similarities between novel observations and previously encountered defects. Feature engineering transforms raw failure descriptions into structured representations capturing temporal sequences, configuration dependencies, and multi-dimensional attribute vectors that machine learning models can effectively process. The foundational knowledge base provides the seed from which adaptive learning algorithms develop increasingly sophisticated validation strategies by providing initial priors to guide the early exploration before enough operational data is available to support purely data-driven decision making.

Continuous orchestration through reinforcement learning-based test vector generation represents, in fact, the dynamic core of the operational methodology, providing a continuous, autonomous adaptation of validation strategies while network characteristics and failure patterns evolve. The reinforcement learning framework models test selection as a sequential decision process where an agent observes the current state of validation coverage, network configuration, and recent defect discoveries and then selects actions corresponding to specific test scenarios for execution. Graph neural networks have recently emerged as powerful tools for learning from network-structured data, with work showing their effectiveness in capturing complex topological patterns and relational dependencies that traditional neural architectures cannot represent [7]. In this context of validation, graph neural networks encode complex relationships among network topology elements, protocol interactions, and configuration parameters to allow the reinforcement learning agent to devise sophisticated policies accounting for structural dependencies in selecting test vectors. The reward signal is multimodal, including defect severity, novelty with respect to previously discovered issues, and efficiency metrics that quantify resource consumption per detected failure, incentivizing the agent to focus on high-impact discoveries while maintaining computational tractability. The exploration strategy balances the need to validate well-understood functionality against the imperative of probing unexplored configuration spaces, resorting to heuristics such as uncertainty-guided sampling that drives the validation effort toward regions where the model has low confidence in its predictions.

Real-time feedback loops with machine learning anomaly flagging create responsive validation systems that dynamically adjust the testing priorities based on emerging indicators of potential failures. Telemetry feeds coming out of network infrastructure are constantly analyzed using ensemble anomaly detection models based on the combination of statistical approaches, deep learning architectures, and domain-specific heuristics to identify abnormalities that are worth exploring. These feedback loops exist on a variety of time scales, with low-latency detection systems responding to acute anomalies in milliseconds, and longer trend analysis systems responding to slow degradation patterns occurring over hours or days. The feedback architecture deploys closed-loop control whereby anomaly detection makes direct control over the test orchestration actions and induces specific validation campaigns that focus on specific subsystems, protocols, or settings that relate to suspicious behaviors. Adaptive sampling techniques modulate telemetry collection rates based on observed network conditions, increasing measurement frequency and granularity in regions exhibiting anomalous behavior while reducing overhead in stable areas, optimizing the tradeoff between visibility and measurement overhead.

The analytic frameworks construct probabilistic graphical models representing causal relationships between network events, with advanced inference methods that enable accurate attribution even in systems with complex interdependencies. Recent advances in causal inference methodologies have shown the possibility of performing automated root cause analysis in distributed systems, with techniques that take advantage of both structural knowledge and observational data to identify failure origins [8]. Graph-based analytics integrate several sources of evidence, including temporal precedence constraints, known architectural dependencies, and statistical associations observed in telemetry data to build comprehensive causal models that explain failure propagation pathways. Recursive remediation mechanisms detect validation infrastructure anomalies and, without manual intervention, automatically reconfigure test environments, implement state rollbacks, or perform mutations of test parameters to isolate failure conditions. Knowledge base evolution is done via structured incorporation of successful defect discoveries: newly validated failure modes enhance the historical corpus and sharpen the models driving the future test selection-a self-reinforcing cycle of continuous improvement of validation effectiveness.

5. Applications and Industry Impact

AI-Native Network Validation systems find critical application in high-density AI fabric environments where advanced port speeds, massive parallel processing requirements, and stringent latency constraints demand unprecedented validation rigor and adaptability. Distinctive traffic patterns imposed by modern artificial intelligence workloads, especially large-scale model training and distributed inference operations, are characterized by synchronized collective communication primitives, bursty all-to-all exchanges, and sensitivity to tail latency that conventional network architectures cannot efficiently accommodate. Non-blocking, low-diameter topologies optimized for high-bandwidth, low-latency communication between accelerator pools are increasingly used in these AI-specialized fabrics, placing switching infrastructure demands for advanced capability support such as adaptive routing, congestion notification, and priority flow control, among other in-network aggregation operations.

Driven by port speeds well beyond traditional specification limits, the result is exponential increases in validation complexity due to the timing sensitivities, signal integrity challenges, and protocol edge cases that only manifest under very specific combinations of traffic load, temperature conditions, and configuration states. AI-Native Network Validation addresses these challenges with continuous adaptation by learning failure signatures specific to AI workload patterns and automatically generating test scenarios that reproduce the exact synchronization, message size distributions, and communication topologies typical of machine learning frameworks. On the one hand, edge computing and distributed inference clusters present unique validation challenges coming from geographic distribution, heterogeneous hardware deployments, and dynamic workload migration patterns, which demand continuous validation as opposed to discrete qualification cycles.

In addition, edge infrastructure deploys computational resources proximate to data sources and end users, creating highly distributed architectures in which network paths span multiple administrative domains, traverse varied link technologies, and exhibit fluctuating performance characteristics. At the same time, the convergence of edge computing and artificial intelligence brings about considerable technical challenges, such as system architecture, communication protocols, resource management, and security mechanisms, all of which should be overcome by broad validation frameworks. Distributed inference operations split large neural network models across several edge nodes; intermediate activations flow through the network between processing stages, which makes inference latency and throughput directly dependent on consistent network performance. Accordingly, AI-Native Network Validation systems adapt to these particular edge challenges by continuous inference performance monitoring in production environments, correlating network telemetry with model accuracy metrics, and automatic generation of validation scenarios to perform stress-testing on just those network paths and failure modes most likely to impact deployed applications. Disaggregated and composable cloud data center architectures represent the basic restructuring of computational infrastructure in which traditional server-centric designs give way to resource-pooled environments with dynamic composition of compute, memory, storage, and acceleration resources, interconnected via high-performance fabrics. Research initiatives investigating rack-scale disaggregated architectures have demonstrated both potential benefits and significant technical challenges associated with hardware resource pooling, especially with respect to network fabric requirements for performance isolation and efficient resource sharing [10].

The challenge of validation further exacerbates as composable architectures grant unprecedented flexibility in resource assignment, creating a combinatorial explosion in possible system configurations that static test planning cannot exhaustively cover. AI-Native Network Validation addresses the validation of composable architecture through digital twin environments, modeling resource pools, allocation policies, and reconfiguration dynamics, allowing the safe exploration of rare configuration sequences and stress conditions. The advantages of AI-Native Network Validation over conventional methods come to the forefront when programmable networks with features like custom forwarding logic, protocol extensions, and specialized packet processing give way to unbounded configuration spaces. Resource optimization through intelligent test prioritization reduces computational overhead by directing validation effort at high-risk areas identified through historical analysis and reinforcement learning-based exploration strategies, achieving superior defect detection rates while consuming fewer computational resources than exhaustive testing approaches.

6. Future Directions and Technical Challenges

The future of self-learning validation ecosystems with automated verification has a transformative trajectory whereby network validation goes beyond human-designed test strategies to fully autonomous systems that can find, characterize, and regress novel failure modes with no intervention from humans. Future validation frameworks are likely to be embedded with the capabilities of meta-learning, allowing for fast adaptation into completely new network architectures, protocols, and failure classes through leveraging knowledge transfer from previously validated systems. Therein, automated verification creates significant technical challenges related to the establishment of formal correctness guarantees for the processes themselves: provably sound anomaly detection, verifiable causal inference, and trustworthy reinforcement learning policies against adversarial conditions and systematic biases in training data. These, in turn, raise some of the ultimate technological requirements: standardization and open interface definitions are crucial for enabling wide adoption and interoperability among heterogeneous vendor ecosystems. Software-defined networking has changed the fundamental architecture of networks through control and data plane separation, creating new paradigms for network management and programmability that require similar evolution in validation methodologies and standardization frameworks [11].

Standardization of interfaces, therefore, faces challenges of expressiveness (enabling complex validation capabilities) versus simplicity (for wide implementation), avoiding lowest-common-denominator compromises on functionality as well as excessive complexity that hinders practical deployment. The most important practical challenges are related to integration with existing validation infrastructure, given the substantial investments in legacy test frameworks and institutional knowledge. Indeed, it is challenging to introduce cognitive validation capabilities incrementally in an organization without disrupting established workflows, and hybrid architectures that allow coexistence and gradual migration are required. The long-term vision of collaborative cross-organizational sharing of validation knowledge involves federated learning architectures by which multiple organizations contribute to collective intelligence about network failure modes while preserving proprietary information. Federated learning frameworks leverage the training of models in a decentralized manner across decentralized data sources while maintaining privacy guarantees for the data contributed; substantial challenges remain, however, regarding efficient communication, statistical heterogeneity, guarantees of convergence, and robustness against adversarial participants [12]. Research needs in autonomous testing frameworks span fundamental questions in machine learning, formal verification, and systems engineering that will have to be addressed to fully unleash the potential of AI-Native Network Validation.

***Table 1**: Core Architectural Components and Their Functions in AI-Native Validation Systems [3][4]*

Component	Primary Function	Key Capability
Telemetry Ingestion Infrastructure	Data Stream Processing	Packet-level traces, interface statistics, protocol state machines
Machine Learning Models	Behavior Analysis	Deviation detection, failure prediction, anomaly correlation
Digital Twin Technology	Safe Test Exploration	Synthetic failure injection, edge case analysis
Reinforcement Learning Framework	Test Optimization	Defect discovery maximization, policy development
Test Orchestration Engine	Central Coordination	Dynamic scenario generation, resource prioritization
Anomaly Cognition Layer	Pattern Recognition	Statistical control, deep learning, heuristic analysis

***Table 2:** Multi-Layer Telemetry Data Sources and Collection Methodologies [5]*

Telemetry Source	Data Type	Collection Method	Information Captured
Interface Counters	Packet/Byte Volumes	Periodic Polling	Traffic statistics
Error Statistics	Forwarding Anomalies	Event-driven	Failure indicators
Protocol State Machines	Control Plane Behavior	State Transition Monitoring	Protocol interactions
In-band Telemetry	Path Information	Packet Header Metadata	Queue depths, latency
Environmental Sensors	Physical Metrics	Continuous Sampling	Temperature, power consumption

***Table 3:** Bootstrap Process Data Sources and Knowledge Base Construction [7][8]*

Data Source Category	Historical Scope	Processing Technique	Output Representation
Defect Repositories	Previous Hardware Generations	Clustering Algorithms	Failure Mode Taxonomies
Regression Test Results	Multiple Product Cycles	Pattern Recognition	Test Parameter Correlations
Field Failure Reports	Production Incidents	Feature Engineering	Temporal Sequences
ASIC Datasheets	Design Specifications	Constraint Integration	Operational Envelopes
Performance Benchmarks	Protocol Requirements	Structured Encoding	Multi-dimensional Vectors

***Table 4:** Application Domain Validation Requirements and Challenge Characteristics [9][10]*

Deployment Context	Infrastructure Type	Primary Challenge	Validation Approach
AI Fabric Environments	High-density Accelerator Pools	Synchronized Communication Patterns	Continuous Adaptation Learning
Edge Computing Clusters	Geographically Distributed Nodes	Heterogeneous Hardware Paths	Production Performance Monitoring

Distributed Inference	Multi-node Model Partitioning	Network-dependent Latency	Telemetry-accuracy Correlation
Disaggregated Data Centers	Resource-pooled Fabrics	Configuration Combinatorial Explosion	Digital Twin Exploration
Composable Architectures	Dynamic Resource Allocation	Unprecedented Assignment Flexibility	Rare Sequence Modeling

7. Conclusions

AI-Native Network Validation fundamentally transforms network assurance from predetermined script execution into autonomous cognitive systems capable of continuous learning and adaptation. The architecture integrates reinforcement learning-based test orchestration, real-time telemetry analysis, and digital twin environments to achieve validation coverage unattainable through conventional methodologies. By leveraging graph neural networks for encoding topological dependencies and causal inference frameworks for automated root cause analysis, these systems efficiently navigate the combinatorial complexity inherent in programmable network infrastructures operating at terabit speeds. Applications across AI fabric deployments, edge computing clusters, and disaggregated data center architectures demonstrate significant advantages in defect detection rates and resource optimization compared to static testing approaches. The move to self-learning validation ecosystems will necessitate solving key technical issues such as telemetry interface standardization, compatibility with the old validation infrastructure, and the creation of federated learning systems that allow knowledge to be shared across organizational boundaries. With the relentless growth in the complexity of networks via programmable data planes and adaptive control systems, autonomous validation systems are required to maintain reliability, minimize qualification cycles, and provide stability in operations on more complex computational fabrics. AI has been studied and applied in different fields [13-33].

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.

- **Author contributions:** The authors declare that they have equal right on this paper.

- **Funding information:** The authors declare that there is no funding to be acknowledged.

- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] IEEE, "IEEE 802.3 Beyond 400 Gb/s Ethernet SG Opening Report," grouper, 2021. [Online]. Available: https://grouper.ieee.org/groups/802/3/minutes/nov21/1121_b400g_open_report.pdf

[2] Pat Bosshart et al., "P4: programming protocol-independent packet processors," ACM SIGCOMM Computer Communication Review, Volume 44, Issue 3, 2014. [Online]. Available: https://dl.acm.org/doi/10.1145/2656877.2656890

[3] Bob Lantz et al., "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks," Hotnets-IX: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, 2010. [Online]. Available: https://dl.acm.org/doi/10.1145/1868447.1868466

[4] Hongzi Mao et al., "Resource Management with Deep Reinforcement Learning," HotNets' '16: Proceedings of the 15th ACM Workshop on Hot Topics in Networks, 2016. [Online]. Available: https://dl.acm.org/doi/10.1145/3005745.3005750

[5] Changhoon Kim et al., "In-band Network Telemetry via Programmable Dataplanes," nkatta. [Online]. Available: https://nkatta.github.io/papers/int-demo.pdf

[6] Rashid Mijumbi et al., "Network Function Virtualization: State-of-the-Art and Research Challenges," IEEE Communications Surveys & Tutorials, Volume 18, Issue 1, 2016. [Online]. Available: https://dl.acm.org/doi/abs/10.1109/comst.2015.2477041

[7] Yifei Shen et al., "Graph Neural Networks for Scalable Radio Resource Management: Architecture Design and Theoretical Analysis," IEEE Journal on Selected Areas in Communications, Volume 39, Issue 1, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9252917

[8] Luan Pham et al., "Root Cause Analysis for Microservices based on Causal Inference: How Far Are We?," arXiv:2408.13729v1, 2024. [Online]. Available: https://arxiv.org/html/2408.13729v1

[9] Dianlei Xu et al., "Edge Intelligence: Architectures, Challenges, and Applications," arXiv:2003.12172, 2020. [Online]. Available: https://arxiv.org/abs/2003.12172
[10] K. Katrinis et al., "Rack-scale disaggregated cloud data centers: The dReDBox project vision," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016. [Online]. Available:https://ieeexplore.ieee.org/document/7459397
[11] Celio Trois et al., "A Survey on SDN Programming Languages: Toward a Taxonomy," IEEE Communications Surveys & Tutorials, Volume 18, Issue 4, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7452335
[12] Tian Li et al., "Federated Learning: Challenges, Methods, and Future Directions," IEEE Signal Processing Magazine, Volume 37, Issue 3, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9084352
[13]Dhinaharan Chockalingam. (2025). Event-Driven Accounting Transformation: From Batch Processing to Real-Time Intelligence. *International Journal of Computational and Experimental Science and Engineering*, *11*(4). https://doi.org/10.22399/ijcesen.4052
[14]Tirso Javier Hernández-Gracia, Cruz Garcia-Lirios, Julio E. Crespo, Miguel Sanhueza-Olave, Isabel Cristina Rincón-Rodríguez, Jorge E. Chaparro-Medina, … B Vishwanath Pradeep. (2025). Confirmatory Factor Analysis of Migrant Entrepreneurship versus Nearshoring. *International Journal of Natural-Applied Sciences and Engineering*, *3*(1). https://doi.org/10.22399/ijnasen.17
[15]Fabiano de Abreu Agrela Rodrigues. (2025). DWRI as a New Neurobiological Perspective of Global Intelligence: From Synaptic Connectivity to Subjective Creativity. *International Journal of Applied Sciences and Radiation Research* , *2*(1). https://doi.org/10.22399/ijasrar.24
[16]Ujjwal Raj. (2025). The Serverless Paradigm: Abstraction, Elasticity, and Event-Driven Computing in Modern Cloud Architectures. *International Journal of Computational and Experimental Science and Engineering*, *11*(4). https://doi.org/10.22399/ijcesen.4088
[17]García, R., Carlos Garzon, & Juan Estrella. (2025). Generative Artificial Intelligence to Optimize Lifting Lugs: Weight Reduction and Sustainability in AISI 304 Steel. *International Journal of Applied Sciences and Radiation Research* , *2*(1). https://doi.org/10.22399/ijasrar.22
[18]Saurabh Pandey, Thrivikram Eskala, & Wenxiao Zhao. (2025). Voice-Driven CICD for SAP Supply Chains: Generative Agents Orchestrating Autonomous Ops. *International Journal of Computational and Experimental Science and Engineering*, *11*(4). https://doi.org/10.22399/ijcesen.4091
[19]Fowowe, O. O., & Agboluaje, R. (2025). Leveraging Predictive Analytics for Customer Churn: A Cross-Industry Approach in the US Market. *International Journal of Applied Sciences and Radiation Research* , *2*(1). https://doi.org/10.22399/ijasrar.20
[20]R. Thiyagarajan, D.Mohana Geetha, Ahmed Mudassar Ali, & K Sreekanth. (2025). Hybrid Digital Twin Solutions for Real-Time Threat Prevention in AI-Driven IoT Networks. *International Journal of Computational and Experimental Science and Engineering*, *11*(2). https://doi.org/10.22399/ijcesen.2066
[21]Ibeh, C. V., & Adegbola, A. (2025). AI and Machine Learning for Sustainable Energy: Predictive Modelling, Optimization and Socioeconomic Impact In The USA. *International Journal of Applied Sciences and Radiation Research* , *2*(1). https://doi.org/10.22399/ijasrar.19
[22]Ravindranath Kunthu, B., & Naveen Prakash Kandula. (2025). Telecom for Humanity: AI-Driven BSS Platforms for Digital Equity and Accessibility. *International Journal of Computational and Experimental Science and Engineering*, *11*(4). https://doi.org/10.22399/ijcesen.4259
[23]Olola, T. M., & Olatunde, T. I. (2025). Artificial Intelligence in Financial and Supply Chain Optimization: Predictive Analytics for Business Growth and Market Stability in The USA. *International Journal of Applied Sciences and Radiation Research* , *2*(1). https://doi.org/10.22399/ijasrar.18
[24]Madhavi Mangalarapua. (2025). Evaluation of DNA damage and repair in Radiographers and Dental Surgeons using X-ray machines in Dental Clinics. *International Journal of Natural-Applied Sciences and Engineering*, *3*(1). https://doi.org/10.22399/ijnasen.14
[25]Venugopal Reddy Depa. (2025). Event-Driven vs. API-Driven: A Strategic Architectural Paradigm Choice. *International Journal of Computational and Experimental Science and Engineering*, *11*(4). https://doi.org/10.22399/ijcesen.4458
[26]Fabiano de Abreu Agrela Rodrigues, & Flávio Henrique dos Santos Nascimento. (2025). Neurobiology of perfectionism. *International Journal of Sustainable Science and Technology*, *3*(1). https://doi.org/10.22399/ijsusat.6
[27]Morad Kh. Hamad. (2025). Synergistic Evaluation of Ionizing Radiation Shielding in Novel Lead-Free Alloys Using Geant4 MC toolkit. *International Journal of Applied Sciences and Radiation Research* , *2*(1). https://doi.org/10.22399/ijasrar.47
[28]B Vishwanath Pradeep. (2025). Psychological research about persistent depressive disorder (dysthymia). *International Journal of Natural-Applied Sciences and Engineering*, *3*(1). https://doi.org/10.22399/ijnasen.18
[29]Kumari, S. (2025). Machine Learning Applications in Cryptocurrency: Detection, Prediction, and Behavioral Analysis of Bitcoin Market and Scam Activities in the USA. *International Journal of Sustainable Science and Technology*, *3*(1). https://doi.org/10.22399/ijsusat.8

[30]Harsha Patil, Vikas Mahandule, Rutuja Katale, & Shamal Ambalkar. (2025). Leveraging Machine Learning Analytics for Intelligent Transport System Optimization in Smart Cities. *International Journal of Applied Sciences and Radiation Research* , *2*(1). https://doi.org/10.22399/ijasrar.38
[31]Chui, K. T. (2025). Artificial Intelligence in Energy Sustainability: Predicting, Analyzing, and Optimizing Consumption Trends. *International Journal of Sustainable Science and Technology*, *3*(1). https://doi.org/10.22399/ijsusat.1
[32]García, R. (2025). Optimization in the Geometric Design of Solar Collectors Using Generative AI Models (GANs). *International Journal of Applied Sciences and Radiation Research* , *2*(1). https://doi.org/10.22399/ijasrar.32
[33]Attia Hussien Gomaa. (2025). Value Engineering in the Era of Industry 4.0 (VE 4.0): A Comprehensive Review, Gap Analysis, and Strategic Framework. *International Journal of Natural-Applied Sciences and Engineering*, *3*(1). https://doi.org/10.22399/ijnasen.22