Global Academic Journal of Economics and Business

Available online at https://www.gajrc.com
DOI: https://doi.org/10.36348/gajeb.2026.v08i05.042

OPEN ACCESS

ISSN: 2706-9001 (P)
ISSN: 2707-2584 (O)

Original Research Article

AI-Powered Intelligent Network Automation Framework for Future-Ready Enterprise Computing Systems

Vibin James[1*]
[1]Independent Researcher

***Corresponding Author**
Vibin James
Independent Researcher

Article History
Received: 29.07.2026
Accepted: 25.09.2026
Published: 28.09.2026

Abstract: Enterprise networks are evolving from relatively static connectivity fabrics into distributed computing systems that span campuses, data centers, clouds, edge nodes, remote users, software-defined overlays, and programmable transport. This heterogeneity makes manual configuration and reactive operations increasingly inadequate. This review synthesizes recent research on artificial intelligence for network automation and develops an AI-powered intelligent network automation framework for future-ready enterprise computing systems. The review integrates evidence from machine learning for networking, intent-based networking, zero-touch management, network telemetry, network digital twins, programmable data planes, anomaly detection, federated learning, and explainable AI. The proposed framework organizes automation as a governed closed loop that translates business intent into machine-verifiable objectives, collects multi-layer telemetry, constructs contextual network state, applies predictive and decision models, validates actions in a digital-twin or policy sandbox, executes changes through software-defined and programmable infrastructure, and continuously verifies outcomes. The synthesis shows that effective autonomy depends less on a single algorithm than on the coordination of data quality, model lifecycle management, policy constraints, observability, security, and human oversight. Key research gaps concern cross-domain state representation, safe reinforcement learning, explainability, concept drift, trustworthy intent translation, multi-vendor interoperability, energy-aware optimization, and methods for measuring automation risk. The review contributes a practical architecture, an evaluation logic, and research directions for enterprises seeking higher resilience, agility, and computing efficiency without surrendering operational accountability.

Keywords: Artificial Intelligence, Network Automation, Intent-Based Networking, Zero-Touch Management, Network Digital Twin, Telemetry, Programmable Networks, Enterprise Computing.

INTRODUCTION

Enterprise computing has become inseparable from networking. Applications now traverse virtual private clouds, software-defined wide area networks, container platforms, edge sites, branch networks, identity services, programmable switches, and public cloud interconnects. The operational consequence is a sharp increase in state variables, dependencies, policies, and failure modes. Traditional management based on device-by-device configuration, threshold alarms, and manually coordinated change windows does not scale well to this environment. Surveys of machine learning in networking therefore increasingly frame intelligence

as a mechanism for classification, prediction, optimization, and adaptive control rather than as an isolated analytics function (Ridwan *et al*., 2021; Kanakis *et al*., 2023). In parallel, network softwarization has created the technical means to act on those insights through software-defined networking, virtualization, APIs, and programmable data planes (Hussain *et al.,* 2022; Kfoury *et al*., 2021; Kaur *et al*., 2021).

The conceptual shift is from automation of individual tasks to automation of operational intent. Intent-based networking formalizes desired outcomes and separates them from low-level implementation, while zero-touch management extends this logic toward self-configuration, self-healing, self-optimization, and continuous service assurance (Leivadeas and Falkner, 2023; Coronado *et al*., 2022; Gallego-Madrid *et al*., 2022). These paradigms are especially relevant to enterprises because application performance, security posture, cost, and user experience often depend on interactions across multiple administrative and technological domains. A path may be healthy at the IP layer while violating application latency, identity, segmentation, or cloud cost objectives. An intelligent automation system therefore needs a richer model of operational context than conventional network monitoring provides.

Recent research offers several enabling mechanisms. In-band telemetry and streaming observability expose fine-grained network state that can support rapid diagnosis and closed-loop control (Tan *et al*., 2021). Network digital twins provide an environment in which alternative configurations or routing policies can be evaluated before production deployment (Wu *et al*., 2021; Almasan *et al*., 2022). Knowledge-defined networking combines programmable control, telemetry, and machine learning to support adaptive decision-making (Ashtari *et al.,* 2022). Reinforcement learning and deep learning provide tools for sequential optimization, although their use in production raises questions about safety, sample efficiency, and non-stationarity (Nguyen *et al*., 2019; Lei *et al*., 2020). Federated learning offers an additional option where operational data cannot be centralized because of privacy, sovereignty, or bandwidth constraints (Yang *et al*., 2022).

Yet greater autonomy also expands operational risk. A model can be accurate on historical data and still fail under concept drift, topology changes, rare incidents, or adversarial manipulation. An automation engine can produce a locally optimal action that violates a global policy or creates instability through interactions with another controller. Explainable AI is consequently relevant not merely for interpretability but for change governance, post-incident analysis, regulatory accountability, and operator trust (Wang *et al*., 2024; Nascita *et al*., 2025). Zero-trust principles also become important because an automation platform itself is a privileged control surface whose identities, data flows, models, and actuation paths require continuous verification (He *et al.,* 2022).

This review addresses the need for an integrated enterprise perspective. Rather than treating prediction, orchestration, telemetry, digital twins, security, and intent translation as separate topics, it asks how they can be combined into a controlled architecture for future-ready enterprise computing. The emphasis is on mechanisms, architectural dependencies, trade-offs, and implementation implications. The resulting framework is not presented as a vendor-specific blueprint; it is a synthesis of design requirements derived from the peer-reviewed literature.

Aim of the Study

The study aims to synthesize contemporary evidence on AI-enabled network automation and to develop an integrated framework that connects intent, telemetry, learning, decision-making, orchestration, validation, security, and assurance for future-ready enterprise computing systems. It also aims to identify the conditions under which higher levels of autonomy can improve operational performance without weakening control, transparency, or resilience.

Research Objectives

The review has five objectives: first, to identify the architectural capabilities required for intelligent closed-loop network automation; second, to compare the roles of supervised, unsupervised, graph-based, federated, reinforcement, and automated machine learning; third, to analyze how intent-based networking, zero-touch management, telemetry, digital twins, and programmable infrastructure interact; fourth, to examine operational risks involving security, explainability, model drift, interoperability, and unsafe actuation; and fifth, to derive an evaluation framework and future research agenda suitable for enterprise deployment.

Research Questions

RQ1: Which technical layers and control functions are necessary for an AI-powered enterprise network automation architecture?

RQ2: How should different AI methods be matched to prediction, anomaly detection, optimization, and policy assurance tasks?

RQ3: What mechanisms can make closed-loop automation safe, explainable, and auditable across

heterogeneous enterprise environments? RQ4: Which unresolved research gaps most constrain progress from assisted operations toward trustworthy autonomy?

REVIEW METHODOLOGY

A structured integrative review design was used because the topic spans networking, distributed systems, machine learning, operations, and security. The method combines systematic search discipline with conceptual synthesis. The uploaded reference paper was used only to inform broad survey organization and academic presentation; its claims, wording, evidence, and references were not reused. The substantive evidence base for this review was independently identified and verified.

Searches were designed around Scopus- and Web of Science-relevant peer-reviewed literature and were cross-checked through publisher records, IEEE Xplore, ACM, Elsevier, Springer, Wiley, institutional repositories, DOAJ, and DOI metadata. Query families combined terms such as "network automation," "zero-touch management," "intent-based networking," "self-driving networks," "machine learning networking," "network digital twin," "in-band telemetry," "programmable data plane," "P4," "network anomaly detection," "reinforcement learning networking," "federated learning 6G," "explainable AI network," and "AutoML." Priority was given to work published from 2020 through 2025, while a small number of earlier papers were retained where they provide foundational treatments of deep reinforcement learning in communications.

Inclusion criteria required a direct contribution to intelligent network management, automation, observability, orchestration, programmable control, learning-based optimization, or governance of AI-enabled network decisions. Sources also had to be peer reviewed and provide verifiable bibliographic metadata and a resolvable DOI. Broad AI papers were included only when their methods materially inform network automation, such as graph neural networks or AutoML. Exclusion criteria removed non-peer-reviewed standards documents, vendor white papers, duplicated preprints when a journal version existed, papers without verifiable DOI metadata, purely physical-layer studies with no management relevance, and application papers whose networking content was incidental.

Screening was qualitative rather than represented through fabricated article counts. Titles and abstracts were assessed for scope, after which full bibliographic records and available article descriptions were checked for methodological relevance. Quality assessment considered venue reputation, clarity of research problem, transparency of method, relevance to automation architecture, and whether limitations were acknowledged. Surveys were used to establish mature taxonomies; focused empirical or architectural studies were used to clarify mechanisms. Because the purpose is conceptual integration rather than meta-analysis, heterogeneous performance measures were not pooled into a synthetic effect size.

The synthesis followed a framework method. Evidence was coded into seven interacting categories: intent and policy; observability and telemetry; state representation and digital twins; predictive and diagnostic intelligence; decision and optimization; programmable actuation; and assurance, security, and governance. Cross-cutting themes included model drift, explainability, privacy, interoperability, resource efficiency, and human oversight. Relationships among categories were then used to derive the proposed architecture shown in Figure 1 and the lifecycle shown in Figure 2. Table 1 maps automation capabilities to suitable AI techniques, while Table 2 defines evaluation dimensions for enterprise adoption.

Methodological limitations arise from rapid technological change, inconsistent terminology across telecommunications and enterprise networking, and the tendency of some studies to evaluate algorithms in simulations or narrow testbeds. Publication bias may favor successful automation demonstrations. In addition, evidence from 5G/6G zero-touch research does not transfer automatically to enterprise environments; this review therefore uses such work for architectural principles and explicitly interprets those principles through enterprise requirements.

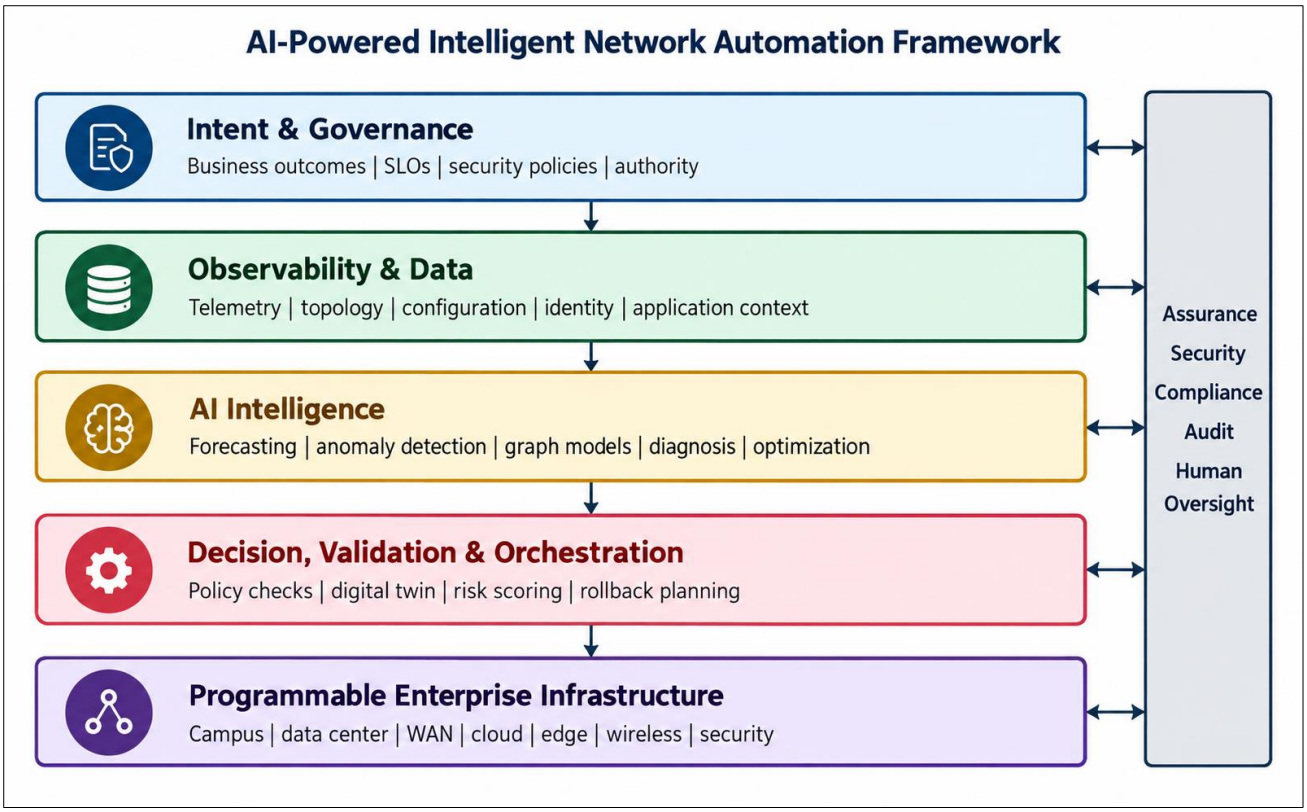

Figure 1: Proposed layered architecture for AI-powered intelligent network automation

Thematic Literature Review and Critical Analysis

The literature converges on the view that autonomous networking is a system problem rather than a model-selection problem. Machine learning contributes prediction and adaptation, but useful autonomy appears only when learning is connected to reliable observation, constrained decision logic, and dependable actuation. Ridwan *et al*., (2021) and Kanakis *et al*., (2023) show that networking tasks span classification, regression, clustering, forecasting, and control, implying that no single learning paradigm is sufficient. AutoML can reduce manual model selection and tuning, but automation of the learning pipeline does not remove the need for domain constraints, data understanding, or operational validation (Santu *et al*., 2022). For enterprise systems, this means the intelligence layer should be modular and task-specific rather than organized around one general model.

Intent and policy form the top layer of the proposed framework. Intent-based networking moves operators from specifying commands toward specifying outcomes, such as reachability, latency bounds, segmentation, resilience, or cost constraints. Leivadeas and Falkner (2023) identify intent profiling, translation, resolution, activation, and assurance as core functions, while Ouyang *et al*., (2021) emphasize refinement from declarative intent to machine-readable policy. The critical difficulty is semantic: natural business statements are incomplete, sometimes conflicting, and rarely encode acceptable risk. An enterprise automation platform therefore requires an intent compiler that normalizes objectives, detects conflicts, assigns priorities, and produces verifiable policies before any network change is authorized. This is also where human governance should remain strongest.

Zero-touch management provides the closed-loop operational pattern. Gallego-Madrid *et al*., (2022), Liyanage *et al*., (2022), and Coronado *et al*., (2022) describe architectures in which monitoring, analytics, orchestration, and assurance cooperate to reduce manual intervention. El Rajab *et al*., (2024) extend this direction by connecting zero-touch operation with AutoML and adaptation to changing data. The enterprise implication is not that humans disappear, but that routine detection, diagnosis, optimization, and rollback can be delegated within explicit confidence and policy boundaries. A useful distinction is therefore between autonomous execution and autonomous authority: systems may execute approved classes of actions automatically while escalation rules preserve human authority for ambiguous, high-impact, or novel situations.

Observability is the prerequisite for both learning and assurance. Conventional polling and aggregated counters can hide transient queueing, microbursts, path changes, and service-chain effects.

In-band network telemetry provides finer-grained path-level state and can support rapid diagnosis, traffic engineering, and closed-loop control (Tan *et al*., 2021). Programmable data planes further allow measurement and selected logic to move closer to packet processing. Kfoury *et al.*, (2021), Kaur *et al*., (2021), and Han *et al*., (2020) show how P4 and related programmability increase flexibility, while also introducing portability, verification, and resource constraints. For enterprise automation, telemetry design must therefore balance resolution against overhead. More data is not automatically better; useful telemetry is aligned to service objectives, synchronized across layers, and enriched with topology, configuration, identity, application, and change-event context.

State representation is the bridge between raw telemetry and intelligent control. Network digital twins offer a promising mechanism because they provide a continuously updated computational representation of network behavior. Wu *et al*., (2021) describe broader digital twin network concepts, while Almasan *et al*., (2022) focus on machine-learning-enabled network twins for performance evaluation and routing optimization. Digital twins are particularly important for safe automation because a candidate action can be tested against predicted consequences before production execution. However, twin fidelity is itself a risk variable. An inaccurate model can create false confidence. Enterprises should therefore treat digital-twin outputs as evidence with uncertainty, maintain calibration against observed outcomes, and define which change classes are safe to simulate versus those requiring staged canary deployment.

Topology-aware learning is another component of state representation. Networks are naturally graphs, and graph neural networks can encode relationships that tabular models often lose. Zhou *et al*., (2020) and Wu *et al*., (2021) explain the expressive value of message passing over graph structures, while Abadal *et al*., (2021) highlights the computational implications of graph learning. In enterprise automation, graph models can support fault localization, path-performance prediction, dependency analysis, and blast-radius estimation. Their advantage is strongest where topology matters directly; their limitation is that rapidly changing graphs, hidden dependencies, and heterogeneous node semantics complicate training and generalization.

Predictive intelligence supports traffic forecasting, capacity planning, and pre-emptive remediation. Jiang *et al*., (2021) show how learning has been applied to congestion control, while Tedjopurnomo *et al*., (2022) illustrates the broader strength of deep models for spatio-temporal prediction. In enterprises, analogous techniques can forecast link utilization, application demand, wireless congestion, or cloud-egress patterns. Forecasting becomes operationally valuable when connected to actions such as pre-scaling, route adjustment, or maintenance scheduling. The main danger is automation based on point predictions without uncertainty. A future-ready framework should carry confidence intervals or risk estimates into the decision layer so that low-confidence forecasts trigger conservative actions.

Anomaly detection and diagnosis address the opposite temporal direction: identifying deviations after or as they occur. Wang *et al*., (2021) documents the breadth of machine-learning approaches for network anomaly detection. Unsupervised and semi-supervised methods are attractive because labeled incidents are scarce, but they often generate alerts that are statistically unusual rather than operationally important. Intelligent automation should therefore combine anomaly scores with service impact, topology, recent change history, and policy context. This reduces alert volume and improves root-cause prioritization. Diagnostic models must also be monitored for data drift because enterprise traffic distributions change with new applications, remote-work patterns, acquisitions, and security controls.

Decision-making is where predictive intelligence becomes operational risk. Reinforcement learning is appealing because it optimizes sequential actions under feedback, and surveys by Nguyen *et al*., (2019) and Lei *et al*., (2020) show its potential for routing, resource allocation, access control, caching, and autonomous IoT. Yet unconstrained online exploration is unsuitable for many enterprise networks. Safe use requires offline learning, digital-twin training, constrained action spaces, policy shields, rollback mechanisms, and staged deployment. Knowledge-defined networking offers a complementary architecture in which telemetry and machine learning enrich software-defined control (Ashtari *et al*., 2022). The combined lesson is that AI should propose or select actions inside an explicit operational envelope rather than replace engineering constraints.

Distributed enterprise environments create data-governance problems for centralized learning. Branches, subsidiaries, edge sites, and regulated workloads may be unable to export detailed telemetry. Federated learning can train shared models while leaving raw data local, reducing some privacy and transfer constraints (Yang *et al*., 2022). Nevertheless, federated learning introduces challenges involving non-identically distributed data,

communication overhead, poisoned updates, and unequal site resources. It should therefore be viewed as a governance-aware learning architecture, not as a default privacy solution.

Security must be integrated across the automation stack. Zero-trust architecture emphasizes continuous verification of identity, context, and authorization rather than implicit trust based on network location (He *et al*., 2022). In an intelligent automation platform, this principle applies to operators, service accounts, telemetry producers, model registries, controllers, and actuation APIs. Model artifacts and training data also require integrity controls because manipulated observations can induce harmful actions. Security automation should consequently include policy-as-code, least-privilege controller access, signed models, protected audit trails, and separation between recommendation and execution privileges for high-risk changes.

Explainability becomes most important when an automated decision changes production state. Wang *et al*., (2024) argues that explainability is central to human-centric intelligent networks, and Nascita *et al*., (2025) reviews its use in traffic analysis, prediction, and intrusion detection. For enterprise operations, explanation should answer practical questions: what evidence triggered the action, which policy objective it served, what alternatives were considered, how confident the model was, and what rollback condition was defined. Explanations that merely expose feature importance are insufficient if they cannot be mapped to operational evidence and change records.

Finally, edge computing changes where automation logic can execute. Carvalho *et al.,* (2021) shows that edge architectures reduce latency and central bottlenecks but increase heterogeneity and distributed management complexity. Future-ready enterprises will therefore need hierarchical automation: fast local loops for latency-sensitive adaptation, domain controllers for coordinated policy, and an enterprise layer for cross-domain intent and governance. The architecture in Figure 1 reflects this hierarchy while keeping assurance and security cross-cutting rather than relegated to a final stage.

Table 1: Mapping of enterprise automation functions to AI methods and control considerations

Automation function	Suitable methods	Typical horizon	Enterprise value	Primary control risk
Demand and performance forecasting	Time-series models; deep learning; graph models	Minutes to weeks	Proactive capacity and routing	Forecast uncertainty and drift
Anomaly detection and triage	Unsupervised, semi-supervised, ensemble learning	Seconds to minutes	Earlier detection and lower alert load	False positives; weak context
Fault localization and dependency reasoning	Graph neural networks; causal/diagnostic models	Seconds to minutes	Reduced mean time to repair	Hidden or stale dependencies
Policy and intent assurance	Semantic parsing; rules; classification	Change time / continuous	Consistent objectives and compliance	Ambiguous intent translation
Resource and path optimization	Constrained optimization; reinforcement learning	Sub-second to hours	Adaptive performance and efficiency	Unsafe exploration; instability
Distributed model learning	Federated learning	Hours to days	Data locality and shared intelligence	Poisoning; non-IID data
Model selection and adaptation	AutoML; drift detection	Hours to weeks	Lower tuning effort and adaptation	Automation of poor objectives

Proposed AI-Powered Intelligent Network Automation Framework

The framework derived from the synthesis has five operational layers connected by an assurance loop. The first layer is intent and governance. It captures business outcomes, service-level objectives, security requirements, cost limits, maintenance constraints, and change authority. Rather than passing free-form requirements directly to an AI model, the layer compiles them into normalized policies with priorities, conflict rules, measurable indicators, and explicit prohibited actions. This separation is important because automation quality cannot exceed the precision of the objectives that define success.

The second layer is observability and data engineering. It ingests streaming telemetry, flow records, logs, configuration snapshots, topology events, cloud and application metrics, identity

context, and change records. Data is time-aligned, quality-scored, and linked to the assets and services it describes. A semantic state store then exposes both current state and recent history. This layer should distinguish authoritative configuration from inferred state and preserve provenance so that every automated decision can later be reconstructed. In-band telemetry and programmable measurement can be used selectively where conventional monitoring lacks sufficient resolution (Tan *et al*., 2021).

The third layer is intelligence. It contains specialized models rather than a monolithic AI engine. Forecasting models estimate demand and performance; anomaly models identify deviations; graph models reason over topology and dependencies; causal or diagnostic models rank likely root causes; and optimization or reinforcement-learning models evaluate candidate actions. AutoML can assist with model selection and adaptation, but deployment gates should consider stability, calibration, inference cost, and explainability in addition to predictive accuracy (Santu *et al*., 2022). Model registry functions should record training data lineage, validation results, approved use cases, and expiration or retraining conditions.

The fourth layer is decision, simulation, and orchestration. Candidate actions are generated from model outputs and filtered through deterministic policy checks. High-impact actions are evaluated in a network digital twin, emulator, or canary environment before execution. A decision record should include expected benefit, uncertainty, blast radius, dependencies, rollback plan, and the evidence that satisfied authorization conditions. Actions are then translated into controller APIs, infrastructure-as-code transactions, service-orchestrator requests, or programmable data-plane updates. Idempotency and transaction semantics are desirable because partial execution can be more damaging than no execution.

The fifth layer is infrastructure and service execution across campus, data center, WAN, cloud, edge, wireless, and security domains. Local controllers retain domain-specific safety rules while exposing standardized capabilities upward. This supports hierarchical autonomy: a wireless controller may optimize channels rapidly, a WAN controller may reroute traffic at a slower horizon, and an enterprise controller may coordinate cross-domain objectives without micromanaging device state. Such hierarchy also limits blast radius because failures in one decision domain need not propagate automatically across the entire enterprise.

Assurance closes the loop across all layers. After actuation, the platform verifies whether the intended service outcome occurred, whether side effects appeared, and whether the original model assumptions remained valid. Verification results update the operational state, generate learning feedback, and determine whether to retain, reverse, or escalate the change. Figure 2 represents this lifecycle as observe, diagnose, predict, decide, validate, act, verify, and learn. Importantly, learning does not imply automatic retraining and redeployment; model updates should pass independent validation before they can influence production control.

This architecture also supports progressive autonomy. At an initial maturity level, AI produces recommendations while humans execute changes. At an intermediate level, low-risk actions execute automatically after policy validation, with operators supervising exceptions. At advanced maturity, multiple domain loops coordinate under shared enterprise intent, but high-impact decisions still require stronger evidence or approval. The progression creates measurable checkpoints for governance and avoids treating full autonomy as a binary destination. It also aligns technical capability with organizational trust, ensuring that increased automation authority follows demonstrated reliability rather than vendor claims or model novelty.

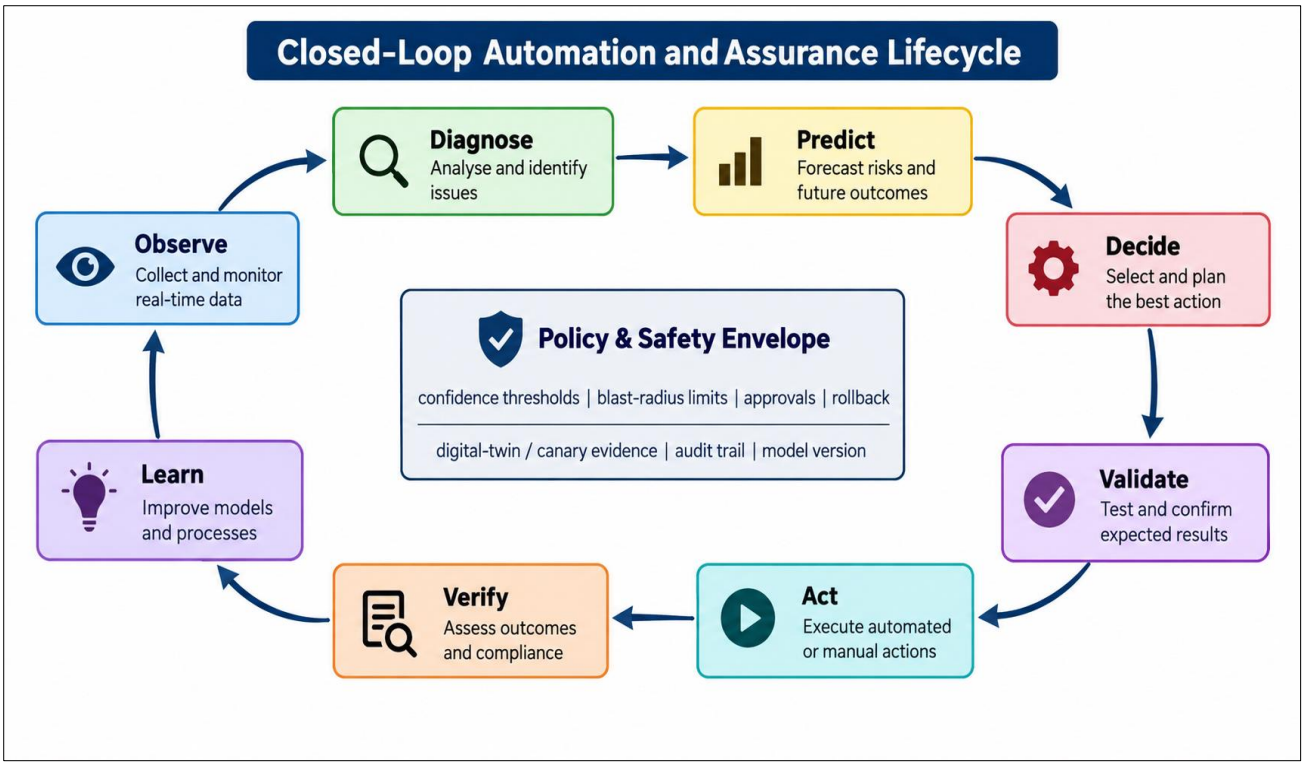

Figure 2: Governed closed-loop lifecycle for production network automation and assurance

DISCUSSION

The synthesis indicates that the central design problem is coordination across timescales and authority boundaries. Telemetry may update in milliseconds, anomaly models in seconds, traffic forecasts in minutes, capacity plans in weeks, and business policies over months. Treating all automation as one loop creates instability and unnecessary coupling. A layered design should therefore separate fast local control from slower cross-domain optimization while sharing a consistent intent and assurance model.

The evidence also suggests that enterprises should measure autonomy by governed capability rather than by the amount of human involvement removed. A mature system knows which actions it may take, what evidence is required, how uncertainty changes its permissions, and when to escalate. This reframes human-in-the-loop design from a temporary limitation into a control mechanism. Zero-touch concepts remain valuable, but production autonomy should be risk-tiered.

A further finding is that data architecture and model operations are as important as network orchestration. Telemetry quality, semantic normalization, lineage, feature consistency, model versioning, drift detection, and rollback are prerequisites for trustworthy decisions. Without them, sophisticated algorithms can amplify poor data or stale assumptions. The proposed framework therefore treats the AI lifecycle as part of network operations rather than as a separate data-science activity.

A practical consequence is that evaluation must capture trade-offs among service performance, operational efficiency, and control risk. An automation policy that reduces latency but increases configuration churn, energy use, or rollback frequency may not improve the enterprise system overall. Similarly, an accurate anomaly model can be operationally weak if it produces explanations too slowly or depends on telemetry unavailable during failure. Evaluation should combine technical metrics with process metrics, including decision latency, intervention rate, policy compliance, recovery time, model calibration, and change success. Longitudinal testing is important because short laboratory experiments rarely expose concept drift, seasonal workload changes, controller interactions, or policy conflicts. This broader measurement approach turns automation from a demonstration of algorithmic capability into an accountable operating discipline.

Table 1 summarizes the relationship between operational functions and AI methods. Table 2 complements it by defining evaluation dimensions that prevent narrow optimization of throughput or latency from obscuring resilience, safety, and governance. Together they support a balanced interpretation of "intelligent" automation: measurable performance improvement under bounded operational risk.

Table 2: Evaluation dimensions for trustworthy enterprise network automation

Dimension	Example indicators	Measurement logic	Common pitfall
Service performance	Latency, loss, throughput, availability, SLA attainment	Compare pre/post change and control periods	Optimizing one metric while degrading another
Operational efficiency	MTTD, MTTR, operator interventions, change lead time	Track automation effect on workflow and recovery	Counting automated actions as value
Model quality	Precision/recall, calibration, drift, uncertainty	Evaluate by task and operating regime	Using aggregate accuracy only
Automation safety	Rollback rate, failed changes, blast radius, oscillation	Measure harmful or unstable actions	Ignoring near misses
Policy and security	Policy violations, authorization failures, audit completeness	Verify every decision against controls	Treating security as post-processing
Resource sustainability	Compute cost, telemetry overhead, energy use	Measure full control-loop cost	Ignoring AI and telemetry overhead
Human governance	Escalation quality, explanation usefulness, approval burden	Assess decision support and accountability	Assuming less human work is always better

Research Gaps and Future Research

Several gaps remain. First, cross-domain state representation is immature. Enterprise services traverse identity, network, cloud, application, and security layers, yet most models optimize within one layer. Research is needed on shared semantic graphs and causal representations that preserve domain ownership while supporting end-to-end reasoning. Second, safe reinforcement learning requires stronger methods for hard policy constraints, uncertainty-aware exploration, and verifiable rollback.

Third, intent translation remains vulnerable to ambiguity. Future systems should combine formal policy languages with language models or semantic parsers, but every generated intent should be validated against deterministic constraints before execution. Fourth, digital twins need measurable fidelity indicators so operators can know when simulated outcomes are trustworthy. Fifth, explainability research should move from generic feature attribution toward operational narratives linked to topology, policy, and change history.

Additional work is needed on concept drift, adversarial manipulation of telemetry and models, federated learning under non-uniform enterprise data, energy-aware optimization, and interoperability across vendors. Research should also define autonomy benchmarks that include time-to-detect, time-to-recover, policy violations, rollback frequency, false automation rate, operator workload, and business-service impact rather than algorithmic accuracy alone.

Practical, Managerial, and Policy Implications

For practitioners, implementation should begin with observability and policy normalization before autonomous control. High-quality telemetry, configuration inventories, topology models, service dependency maps, and identity context create the foundation on which learning can operate. Initial automation should target repetitive, reversible, high-volume tasks such as anomaly triage, capacity forecasting, configuration compliance, and bounded remediation. Digital-twin or canary validation should precede broader closed-loop actuation.

For managers, the main governance decision is defining automation authority. Change classes can be tiered by blast radius, reversibility, confidence, and regulatory impact. Low-risk actions may execute automatically; medium-risk actions may require approval; high-risk or novel actions should remain advisory. Ownership should be explicit across network engineering, platform operations, cybersecurity, data science, and application teams because autonomous outcomes cross organizational boundaries.

Policy makers and enterprise governance functions should require traceability for AI-driven changes, including source telemetry, model version, decision rationale, policy checks, executed commands, verification results, and rollback status. Procurement criteria should emphasize open APIs, portable telemetry, model exportability, and evidence of interoperability. Such requirements reduce lock-in and make auditability an architectural property rather than an afterthought.

Limitations

This review is conceptual and does not perform a quantitative meta-analysis because the included literature uses heterogeneous tasks, datasets, metrics, and test environments. Several influential zero-touch and 6G studies originate in telecommunications and must be adapted cautiously to enterprise networks. Rapid advances in generative AI and autonomous agents may also outpace the reviewed evidence. Finally, DOI verification confirms bibliographic validity but does not guarantee that

every proposed method has been validated at production enterprise scale.

CONCLUSION

AI-powered network automation is best understood as a governed closed-loop architecture rather than as the deployment of isolated machine-learning models. Future-ready enterprise computing requires the coordinated translation of intent, continuous observability, contextual state representation, predictive and diagnostic intelligence, constrained decision-making, programmable actuation, and outcome assurance. Research on intent-based networking, zero-touch management, telemetry, digital twins, programmable data planes, graph learning, reinforcement learning, federated learning, and explainable AI provides complementary components for this architecture.

The review shows that trustworthy autonomy depends on explicit policy boundaries, reliable data, uncertainty awareness, pre-deployment validation, secure control paths, and auditable explanations. Enterprises should therefore progress incrementally from decision support to bounded autonomy, expanding automated authority only when evidence demonstrates stable performance and safe rollback. The proposed framework and evaluation dimensions provide a basis for this progression. The most important future advances are likely to come from cross-domain semantic state models, safe learning under hard constraints, fidelity-aware digital twins, operationally useful explanations, and standardized measures of automation risk. These developments can make enterprise networks more adaptive and resilient while preserving the accountability required for critical computing infrastructure.

REFERENCES

- Abadal, S., Jain, A., Guirado, R., Lopez-Alonso, J., & Alarcon, E. (2022). Computing graph neural networks: A survey from algorithms to accelerators. ACM Computing Surveys, 54(9), 1-38. https://doi.org/10.1145/3477141
- Almasan, P., Ferriol-Galmes, M., Paillisse, J., Suarez-Varela, J., Perino, D., Lopez, D. R., Pastor Perales, A. A., Harvey, P., Ciavaglia, L., Wong, L., Ram, V., Xiao, S., Shi, X., Cheng, X., Cabellos-Aparicio, A., & Barlet-Ros, P. (2022). Network digital twin: Context, enabling technologies, and opportunities. IEEE Communications Magazine, 60(11), 22-27. https://doi.org/10.1109/MCOM.001.2200012
- Ashtari, S., Zhou, I., Abolhasan, M., Shariati, N., Lipman, J., & Ni, W. (2022). Knowledge-defined networking: Applications, challenges and future work. Array, 14, 100136. https://doi.org/10.1016/j.array.2022.100136
- Carvalho, G., Cabral, B., Pereira, V., & Bernardino, J. (2021). Edge computing: Current trends, research challenges and future directions. Computing, 103, 993-1023. https://doi.org/10.1007/s00607-020-00896-5
- Coronado, E., Behravesh, R., Subramanya, T., Fernandez-Fernandez, A., Siddiqui, M. S., Costa-Perez, X., & Riggio, R. (2022). Zero touch management: A survey of network automation solutions for 5G and 6G networks. IEEE Communications Surveys & Tutorials, 24(4), 2535-2578. https://doi.org/10.1109/COMST.2022.3212586
- El Rajab, M., Yang, L., & Shami, A. (2024). Zero-touch networks: Towards next-generation network automation. Computer Networks, 243, 110294. https://doi.org/10.1016/j.comnet.2024.110294
- Gallego-Madrid, J., Sanchez-Iborra, R., Ruiz, P. M., & Skarmeta, A. F. (2022). Machine learning-based zero-touch network and service management: A survey. Digital Communications and Networks, 8(2), 105-123. https://doi.org/10.1016/j.dcan.2021.09.001
- Han, S., Jang, S., Choi, H., Lee, H., & Pack, S. (2020). Virtualization in programmable data plane: A survey and open challenges. IEEE Open Journal of the Communications Society, 1, 527-534. https://doi.org/10.1109/OJCOMS.2020.2990182
- He, Y., Huang, D., Chen, L., Ni, Y., & Ma, X. (2022). A survey on zero trust architecture: Challenges and future trends. Wireless Communications and Mobile Computing, 2022, 6476274. https://doi.org/10.1155/2022/6476274
- Hussain, M., Shah, N., Amin, R., Alshamrani, S. S., Alotaibi, A., & Raza, S. M. (2022). Software-defined networking: Categories, analysis, and future directions. Sensors, 22(15), 5551. https://doi.org/10.3390/s22155551
- Jiang, H., Li, Q., Jiang, Y., Shen, G., Sinnott, R., Tian, C., & Xu, M. (2021). When machine learning meets congestion control: A survey and comparison. Computer Networks, 192, 108033. https://doi.org/10.1016/j.comnet.2021.108033
- Kanakis, M. E., Khalili, R., & Wang, L. (2023). Machine learning for computer systems and networking: A survey. ACM Computing Surveys, 55(4), Article 71, 1-36. https://doi.org/10.1145/3523057
- Kaur, S., Kumar, K., & Aggarwal, N. (2021). A review on P4-programmable data planes: Architecture, research efforts, and future directions. Computer Communications, 170, 109-129. https://doi.org/10.1016/j.comcom.2021.01.027

- Kfoury, E. F., Crichigno, J., & Bou-Harb, E. (2021). An exhaustive survey on P4 programmable data plane switches: Taxonomy, applications, challenges, and future trends. IEEE Access, 9, 87094-87155. https://doi.org/10.1109/ACCESS.2021.3086704
- Lei, L., Tan, Y., Zheng, K., Liu, S., Zhang, K., & Shen, X. (2020). Deep reinforcement learning for autonomous Internet of Things: Model, applications and challenges. IEEE Communications Surveys & Tutorials, 22(3), 1722-1760. https://doi.org/10.1109/COMST.2020.2988367
- Leivadeas, A., & Falkner, M. (2023). A survey on intent-based networking. IEEE Communications Surveys & Tutorials, 25(1), 625-655. https://doi.org/10.1109/COMST.2022.3215919
- Liyanage, M., Pham, Q.-V., Dev, K., Bhattacharya, S., Maddikunta, P. K. R., Gadekallu, T. R., & Yenduri, G. (2022). A survey on zero touch network and service management (ZSM) for 5G and beyond networks. Journal of Network and Computer Applications, 203, 103362. https://doi.org/10.1016/j.jnca.2022.103362
- Nascita, A., Aceto, G., Ciuonzo, D., Montieri, A., Persico, V., & Pescape, A. (2025). A survey on explainable artificial intelligence for Internet traffic classification and prediction, and intrusion detection. IEEE Communications Surveys & Tutorials, 27(5), 3165-3198. https://doi.org/10.1109/COMST.2024.3504955
- Nguyen, N. C. L., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y.-C., & Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking: A survey. IEEE Communications Surveys & Tutorials, 21(4), 3133-3174. https://doi.org/10.1109/COMST.2019.2916583
- Ouyang, Y., Yang, C., Song, Y., Mi, X., & Guo, L. (2021). A brief survey and implementation on refinement for intent-driven networking. IEEE Network, 35(6), 75-83. https://doi.org/10.1109/MNET.001.2100194
- Ridwan, M. A., Radzi, N. A. M., Abdullah, F., & Jalil, Y. E. (2021). Applications of machine learning in networking: A survey of current issues and future challenges. IEEE Access, 9, 52523-52556. https://doi.org/10.1109/ACCESS.2021.3069210
- Santu, S. K. K., Hassan, M. M., Smith, M. J., Xu, L., Zhai, C., & Veeramachaneni, K. (2022). AutoML to date and beyond: Challenges and opportunities. ACM Computing Surveys, 54(8), Article 175. https://doi.org/10.1145/3470918
- Tan, L., Su, W., Zhang, W., Lv, J., Zhang, Z., Miao, J., Liu, X., & Li, N. (2021). In-band network telemetry: A survey. Computer Networks, 186, 107763. https://doi.org/10.1016/j.comnet.2020.107763
- Tedjopurnomo, D. A., Bao, Z., Zheng, B., Choudhury, F. M., & Qin, A. K. (2022). A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. IEEE Transactions on Knowledge and Data Engineering, 34(4), 1544-1561. https://doi.org/10.1109/TKDE.2020.3001195
- Wang, S., Balarezo, J. F., Kandeepan, S., Al-Hourani, A., Chavez, K. G., & Rubinstein, B. I. P. (2021). Machine learning in network anomaly detection: A survey. IEEE Access, 9, 152379-152396. https://doi.org/10.1109/ACCESS.2021.3126834
- Wang, S., Qureshi, M. A., Miralles-Pechuan, L., Huynh-The, T., Gadekallu, T. R., & Liyanage, M. (2024). Explainable AI for 6G use cases: Technical aspects and research challenges. IEEE Open Journal of the Communications Society, 5, 2490-2540. https://doi.org/10.1109/OJCOMS.2024.3386872
- Wu, Y., Zhang, K., & Zhang, Y. (2021). Digital twin networks: A survey. IEEE Internet of Things Journal, 8(18), 13789-13804. https://doi.org/10.1109/JIOT.2021.3079510
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4-24. https://doi.org/10.1109/TNNLS.2020.2978386
- Yang, Z., Chen, M., Wong, K.-K., Poor, H. V., & Cui, S. (2022). Federated learning for 6G: Applications, challenges, and opportunities. Engineering, 8, 33-41. https://doi.org/10.1016/j.eng.2021.12.002
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. AI Open, 1, 57-81. https://doi.org/10.1016/j.aiopen.2021.01.001