# Machine Learning-Based Network Intelligence Framework for Autonomous Data Center Operations


## Abstract:

Data centers increasingly depend on highly dynamic, software-defined networks whose operating conditions change faster than manual procedures can safely accommodate. This review examines how machine learning can be organized as a network intelligence layer for autonomous data center operations, with emphasis on telemetry interpretation, anomaly detection, forecasting, graph-based network modeling, reinforcement learning, intent translation, and closed-loop assurance. An integrative review approach is used to synthesize peer-reviewed research published primarily from 2020 to 2025 across networking, cloud computing, data-center energy management, artificial intelligence, and autonomous systems. The evidence indicates that no single algorithm is sufficient for end-to-end autonomy. Supervised and unsupervised models are effective for classification and anomaly recognition, graph neural networks improve representation of topology-dependent behavior, forecasting models anticipate load and congestion, and reinforcement learning supports sequential control for routing, consolidation, and cooling. However, operational deployment remains constrained by concept drift, incomplete observability, sparse labels, unsafe exploration, weak cross-domain coordination, and limited evidence from production-scale evaluations. The review therefore proposes a layered Machine Learning-Based Network Intelligence Framework that separates observation, representation, intelligence, decision, actuation, and assurance. It argues that dependable autonomy requires probabilistic learning to operate inside deterministic policy boundaries, with confidence estimation, digital-twin validation, rollback, and human escalation for higher-risk actions. This architecture provides a practical research agenda for progressing from isolated AIOps functions toward trustworthy closed-loop data-center operations.



**Keywords:** data center networks; machine learning; autonomous operations; graph neural networks; reinforcement learning; AIOps; intent-based networking; digital twins


## Introduction

Modern data centers are cyber-physical computing environments in which servers, storage systems, switches, virtual machines, containers, accelerators, cooling equipment, and control

platforms interact through tightly coupled feedback loops. Network performance is central to this interaction because east-west traffic, workload migration, distributed storage, microservices, and accelerator fabrics depend on predictable latency, throughput, and loss. Conventional management based on static thresholds, device-by-device configuration, and post-incident troubleshooting is increasingly mismatched to this operating environment. Software-defined networking and programmable telemetry provide richer observability and faster control, but they also increase the volume and dimensionality of operational data that must be interpreted before useful action can be taken. Machine learning therefore becomes valuable not simply as a predictive tool, but as a mechanism for converting high-frequency telemetry into operational knowledge and policy decisions.

Recent work demonstrates several complementary directions. Deep reinforcement learning has been applied to intelligent routing and load balancing in software-defined data-center networks, where an agent can learn adaptive traffic-engineering strategies from changing network state (Liu et al., 2021; Rikhtegar et al., 2021). Online machine-learning classification has also been placed close to programmable forwarding functions so that application traffic can be recognized and routed according to quality-of-service requirements (Xie et al., 2022). These approaches move control from reactive rules toward context-sensitive decisions, but their effectiveness depends on the quality of state representation, reward formulation, operational constraints, and the ability to generalize when workloads or topologies change.

Autonomous operation also extends beyond packet forwarding. Data-center energy and thermal systems are increasingly optimized using machine learning, forecasting, and reinforcement learning. Wang et al. (2022) investigate deep-reinforcement-learning-based energy-aware networking, while Mahbod et al. (2022), Zhang et al. (2023), and Wang et al. (2023) show how learning can support adaptive cooling control. Shaw et al. (2022) apply reinforcement learning to virtual-machine consolidation, demonstrating that compute placement and network conditions cannot always be optimized independently. These studies suggest that a network-intelligence layer should understand workload, energy, and thermal context whenever these variables influence routing, capacity, or risk.

A second challenge is that detection accuracy does not automatically translate into dependable autonomy. Telemetry-driven anomaly detection can exploit fine-grained in-band measurements and graph relationships (Duan et al., 2023), but models can degrade when the meaning of normal behavior changes. Concept-drift methods therefore become operationally important (Jain et al., 2022), as do multivariate time-series anomaly approaches that can represent correlated infrastructure signals (Li & Jung, 2023; Audibert et al., 2022). The central problem is not simply

whether a model can identify an anomaly, but whether an automated system can distinguish uncertainty from urgency, select an appropriate intervention, and verify that the intervention improved service without creating another failure.

This review integrates these fragmented research streams around network intelligence as an operational architecture. Its central argument is that dependable autonomy emerges from coordinated sensing, topology-aware representation, predictive and diagnostic learning, constrained decision-making, programmable actuation, and independent assurance. This systems perspective clarifies where machine learning creates operational value and where deterministic controls or human authorization remain necessary.

**Aim of the Study**

The aim of this review is to develop an evidence-based conceptual framework explaining how machine learning can transform data-center network telemetry into safe, adaptive, and increasingly autonomous operational decisions. The study integrates research on intelligent routing, anomaly detection, topology-aware modeling, resource orchestration, energy optimization, intent-based networking, and digital twins. Rather than proposing a single algorithm, it identifies the functional roles that different learning paradigms can perform within a closed-loop architecture and the assurance mechanisms required before automated actions are applied to production infrastructure.

**Research Objectives**

The review pursues five objectives. First, it synthesizes how supervised learning, unsupervised learning, deep learning, graph neural networks, and reinforcement learning are currently used for network and infrastructure operations. Second, it compares their operational strengths, data requirements, and failure modes across detection, prediction, optimization, and control tasks. Third, it examines the role of programmable telemetry, intent-based control, and digital twins in connecting learned models with closed-loop operations. Fourth, it identifies technical and governance barriers that prevent experimental models from becoming dependable autonomous services. Fifth, it develops an integrated Machine Learning-Based Network Intelligence Framework that separates observation, intelligence, decision, actuation, and assurance while enabling feedback between them.

**Research Questions**

Four research questions guide the synthesis: RQ1: Which machine-learning capabilities are most suitable for sensing, prediction, diagnosis, and control in autonomous data-center

operations? RQ2: How can topology, time-series behavior, logs, and policy intent be combined into a unified network-intelligence pipeline? RQ3: What technical risks limit safe closed-loop deployment, particularly under concept drift, topology change, uncertain predictions, and rare failures? RQ4: Which architectural controls can preserve reliability, explainability, and human accountability while increasing the speed and scope of automated operations?

**Review Methodology**

A structured integrative review design was adopted because the topic spans computer networking, cloud infrastructure, machine learning, energy management, and autonomous control. The review was designed to compare mechanisms and operational implications rather than to calculate a pooled statistical effect. Searches were framed for Scopus, Web of Science, IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and Wiley Online Library. Core search expressions combined terms such as “data center network,” “autonomous operations,” “machine learning,” “deep reinforcement learning,” “graph neural network,” “network telemetry,” “anomaly detection,” “traffic engineering,” “intent-based networking,” “digital twin network,” “resource allocation,” “virtual machine consolidation,” and “data center cooling.” Boolean variants were used to capture combinations of network control and infrastructure optimization.

The temporal emphasis was 2020-2025 to reflect current software-defined, telemetry-rich, and learning-enabled environments. Earlier work was retained only where it remained foundational to a method represented in the recent literature. Included studies had to be peer reviewed, directly relevant to network or data-center operational intelligence, and sufficiently detailed to assess the learning method, data or environment, operational objective, and evaluation approach. Studies were excluded when they focused only on generic machine-learning accuracy without an infrastructure-management connection, lacked an identifiable peer-reviewed source, or provided insufficient methodological detail.

Screening proceeded by relevance of title and abstract, followed by full-text assessment against the inclusion criteria. No numerical screening totals are reported because this review did not maintain an auditable PRISMA-style counting workflow, and inventing retrospective counts would create false precision. Quality assessment considered clarity of the operational problem, appropriateness of the data and experimental setting, comparability of baselines, treatment of constraints, evidence of robustness or generalization, and transparency about limitations. Greater interpretive weight was given to studies using realistic traces, programmable-network testbeds, production-derived data, or explicit validation constraints.

The synthesis used four stages. First, studies were coded by operational function: observation, detection, prediction, topology modeling, optimization, actuation, or assurance. Second, methods were grouped by learning family. Third, evidence was compared across network, compute, energy, and cooling domains to identify transferable design principles and domain-specific risks. Fourth, recurring gaps were translated into architectural requirements for the proposed framework. Because datasets, topologies, metrics, and control horizons differ substantially, results are interpreted comparatively rather than numerically pooled. This methodological choice supports cross-domain synthesis while avoiding misleading performance rankings.

For data extraction, each retained source was examined for the operational problem, learning paradigm, data source, network or infrastructure scope, control horizon, evaluation setting, reported benefits, and stated limitations. Bibliographic details and DOI identifiers were cross-checked against publisher pages or Crossref-linked records before inclusion in the final reference set. The synthesis deliberately avoids converting heterogeneous experiments into a league table. Instead, reported findings are used to identify recurring architectural relationships, such as the dependence of control quality on observability, the importance of topology-aware state, and the need for assurance before actuation.

**Thematic Literature Review and Critical Analysis**

Telemetry is the foundation of network intelligence because learning models cannot compensate for missing or poorly synchronized observations. Traditional counters provide useful aggregate state but may miss transient events such as microbursts. In-band network telemetry can expose path-level and device-level measurements at much finer granularity, creating data suitable for spatio-temporal learning. Duan et al. (2023) combine telemetry with graph-based feature fusion to identify subtle anomalies, illustrating the benefit of representing both temporal change and network structure. Online traffic classification offers a complementary capability: Xie et al. (2022) separate offline model training from online inference so that flow classes can influence routing without placing the full training burden on the forwarding path. These approaches show that autonomous operation depends on a carefully engineered observation layer, not merely a powerful model.

Anomaly detection research also reveals the limits of static learning assumptions. Network and infrastructure telemetry are multivariate time series in which normal behavior changes with workload mix, software releases, topology modifications, and seasonal demand. Li and Jung (2023) categorize deep approaches around different anomaly forms, while Zamanzadeh Darban et al. (2024) emphasize the diversity of forecasting, reconstruction, representation-learning, and hybrid strategies for time-series anomaly detection. Audibert et al. (2022) provide a counterpoint:

deep neural approaches do not uniformly outperform conventional methods, particularly where training data are limited or anomalies have simpler statistical structure. For autonomous operations, this implies that model selection should be evidence driven and workload specific rather than based on architectural novelty.

Concept drift is particularly consequential because an automated response policy can amplify an outdated detector. Jain et al. (2022) explicitly address drift in network anomaly detection, showing why retraining triggers and adaptive decision boundaries matter when data distributions change. A production system should therefore monitor both infrastructure state and model state. Prediction error, calibration, feature distribution, class balance, and alert prevalence can become operational signals. Drift does not always require immediate retraining; sometimes abstention, fallback to a deterministic baseline, or human review is safer.

Logs provide another evidence channel for diagnosing distributed-system behavior. QLLog uses reinforcement learning concepts to adapt log anomaly assessment and to incorporate feedback into detection (Duan et al., 2021). Logs differ from numerical telemetry because their semantics depend on software versions, templates, and execution context. A robust network-intelligence system should therefore correlate log events with topology, traffic, configuration changes, and service ownership rather than treat each stream independently. Multimodal correlation can reduce false causality, especially during cascading failures.

Topology-aware learning is one of the strongest arguments for graph neural networks in data-center operations. Standard tabular models treat features as independent columns even though network outcomes emerge from relationships between nodes, links, paths, queues, and flows. RouteNet demonstrates that graph neural networks can learn network performance models that generalize across network structures and routing configurations (Rusek et al., 2020). RouteNet-Fermi extends this direction with more expressive network modeling and improved scalability considerations (Ferriol-Galmés et al., 2023). Broader surveys confirm that graph learning is increasingly used for routing, resource allocation, offloading, and orchestration because message passing matches the relational structure of networks (Tam et al., 2022; Waikhom & Patgiri, 2023; Zhou et al., 2022).

The operational value of graph models is not limited to prediction. They can provide compact state representations for reinforcement-learning agents, reducing the need to hand-engineer per-topology features. This is important because policies trained on fixed vector layouts often fail when device counts or connectivity change. A graph encoder can produce topology-aware embeddings that remain structurally meaningful across variations. However, generalization is not automatic: unseen hardware behaviors, queue disciplines, routing protocols, or traffic regimes can

still create out-of-distribution states. Validation therefore needs structural as well as statistical coverage.

Reinforcement learning is attractive for network control because routing, scheduling, consolidation, and cooling are sequential decisions with delayed consequences. DRL-R demonstrates deep-reinforcement-learning-based intelligent routing in software-defined data-center networks (Liu et al., 2021), while DeepRLB applies deep reinforcement learning to load balancing under heterogeneous flow conditions (Rikhtegar et al., 2021). Wang et al. (2022) extend the control perspective toward energy-aware data-center networking. These studies illustrate a key advantage of reinforcement learning: the agent can optimize cumulative outcomes rather than make isolated predictions. They also expose its main risk: the policy learns the objective encoded in the reward, not the operator's unstated intentions.

Reward design is a major source of hidden risk. Optimizing mean latency can sacrifice tail latency, resilience, fairness, or energy, while energy-oriented rewards can increase thermal or failure risk. Adjacent data-center research illustrates this trade-off: Hu and Sun (2020) study power-resource management; Shaw et al. (2022) examine virtual-machine consolidation; and Mahbod et al. (2022) and Zhang et al. (2023) investigate learning-based cooling. These studies show potential gains but also sensitivity to constraints, workload conditions, reward functions, and implementation assumptions.

Physics-informed and model-assisted learning offers one response to unsafe exploration. Phyllis incorporates physical knowledge into lifelong reinforcement learning for data-center cooling, aiming to constrain learning with domain structure rather than relying solely on trial-and-error (Wang et al., 2023). More generally, safety can be improved by training policies in simulation, using conservative action spaces, defining invariant constraints, and requiring candidate actions to pass a model-based or digital-twin check. This shifts reinforcement learning from unrestricted control toward bounded optimization.

Resource orchestration demonstrates the need to coordinate network and compute state. Deep reinforcement learning has been studied for cloud-edge resource allocation and collaborative scheduling (Cen & Li, 2022; Xu et al., 2022), while Yang, Yu, et al. (2022) use neural prediction and policy-gradient methods for energy reduction. Machine-learning studies of data-center efficiency and evidence-based energy management likewise show that workload, thermal, and energy context can affect infrastructure decisions (Yang, Du, et al., 2022; Khan et al., 2023). A network controller should therefore coordinate with compute orchestration rather than repeatedly react to congestion produced by independent scheduling.

Autonomous control also requires a mechanism for expressing desired outcomes. Intent-based networking shifts configuration from device-level commands toward declarative objectives and assurance loops. Pang et al. (2020) describe intent-driven networks as a route toward self-configuring and self-optimizing behavior, while Leivadeas and Falkner (2023) emphasize intent expression, translation, activation, and assurance as core components of a closed loop. Machine learning can contribute to each stage, but intent must remain bounded by machine-verifiable policy. Natural-language or learned intent interpretation should never be allowed to bypass explicit security, redundancy, change-window, or compliance constraints.

Digital twins strengthen assurance by providing an updated environment for testing likely consequences before actuation. Wu et al. (2021) describe digital twin networks as virtual representations that co-evolve with physical networks, while Si-Mohammed et al. (2024) integrate simulation with network digital-twin decision support. For data centers, a useful twin can be a calibrated set of models estimating latency, congestion, energy, or failure impact under candidate actions. Graph-based learned surrogates may accelerate such evaluation when packet-level simulation is too slow.

Figure 1 positions these findings as a layered framework. The observation layer collects synchronized network telemetry, logs, configuration events, workload context, and infrastructure signals. A representation layer creates time-series features, event embeddings, and topology graphs. The intelligence layer performs anomaly detection, forecasting, graph-based performance estimation, and causal or probabilistic diagnosis. The decision layer combines candidate actions from optimizers or reinforcement-learning policies with intent, operational policy, and hard constraints. The actuation layer uses programmable network and orchestration interfaces. An independent assurance path estimates confidence, detects out-of-distribution states, tests higher-risk actions, verifies outcomes, and triggers rollback or human escalation.

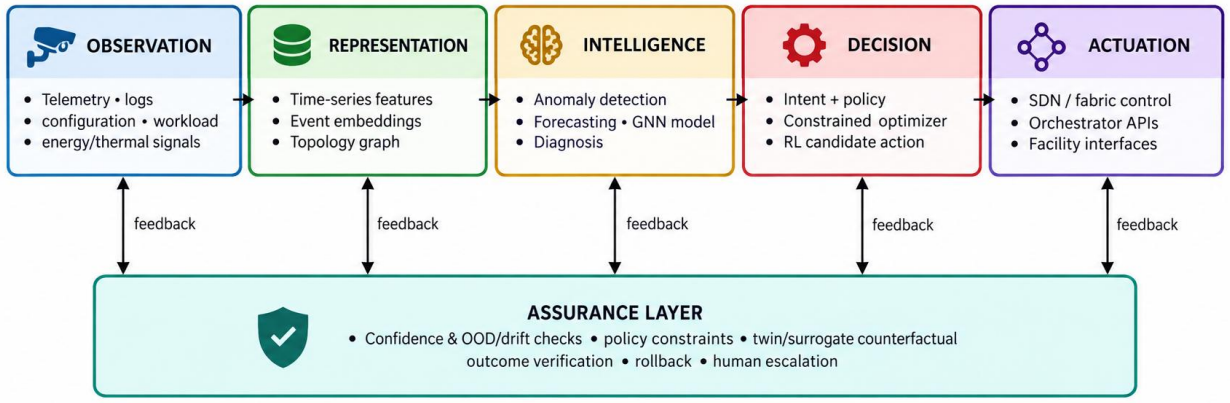


*Figure 1. Machine Learning-Based Network Intelligence Framework for Autonomous Data Center Operations.*

Table 1 compares the principal machine-learning families. Supervised models are strongest when high-quality labels exist and the operational classes are stable, but labels for rare failures are scarce. Unsupervised and self-supervised methods reduce label dependence yet make threshold calibration difficult. Time-series models capture temporal dependencies, graph neural networks preserve relational structure, and reinforcement learning supports adaptive sequential control. Hybrid designs can combine these strengths, but their complexity makes testing, explainability, and lifecycle management harder. The appropriate architecture therefore depends on the operational action, not only predictive accuracy.

| ML family | Primary role | Common inputs | Strength | Main limitation | Best-fit autonomy tier |
|---|---|---|---|---|---|
| Supervised classifiers | Traffic/fault classification | Labeled flows, events, KPIs | Strong discriminative baseline | Label scarcity and drift | Recommendation / bounded automation |
| Unsupervised / self-supervised | Novelty and anomaly detection | Unlabeled telemetry and logs | Reduces label dependence | Threshold calibration | Triage / early warning |
| Time-series | Forecasting / | Multivariat | Captures | Drift and | Predictive pre- |

| | | | | | |
|---|---|---|---|---|---|
| deep models | temporal anomalies | e KPI sequences | temporal dependence | calibration | emption |
| Graph neural networks | Topology-aware performance modeling | Topology and link/node state | Preserves relational structure | OOD topology and scaling | Modeling / routing support |
| Reinforcement learning | Sequential network/infrastructure control | State, actions, rewards | Adaptive multi-step control | Unsafe exploration / reward error | Constrained closed-loop control |
| Hybrid / twin-assisted | Counterfactual decision support | Telemetry, graphs, simulator | Cross-model assurance | Complexity and twin fidelity | Higher-risk automation with guardrails |

*Table 1. Machine-learning families for autonomous data-center operations.*

Security and privacy must be embedded in network intelligence. Tang et al. (2022) show that federated intrusion detection can reduce raw-data centralization, although federation introduces poisoning, heterogeneity, and coordination risks. Because autonomous controllers are high-value targets, production designs require signed model artifacts, protected policy stores, role-separated approvals, feature provenance, and continuous behavior monitoring.

The literature supports graded autonomy. Recommendation, anomaly ranking, and forecasting can use permissive thresholds; bounded traffic rebalancing can execute automatically when rollback is immediate; and changes affecting redundancy, security segmentation, or multi-domain dependencies should require stronger validation or approval. This tiering aligns decision speed with operational consequence and creates a practical path from AIOps toward autonomous control.

Model lifecycle engineering is therefore as important as initial training. An autonomous data-center controller should maintain explicit links between telemetry schema versions, feature transformations, model artifacts, policies, and the infrastructure configuration against which validation was performed. When any of these dependencies changes, previous evidence may no longer justify automated authority. Continuous evaluation should include shadow deployment, canary actions, counterfactual replay, and post-action verification. These practices also help separate model error from data-pipeline or actuator error, which is essential because the same service symptom can originate from prediction failure, stale state, incorrect policy translation, or

unsuccessful execution. Treating the learning pipeline as part of the production control plane makes provenance and reproducibility operational requirements rather than documentation tasks.

## Discussion

The synthesis indicates that the most important design choice is not the selection of a single "best" model, but the separation of responsibilities across a closed-loop system. Detection, prediction, diagnosis, optimization, and authorization are distinct functions. Conflating them creates brittle automation because a model trained to predict congestion does not automatically know whether rerouting is permitted, whether a maintenance window is active, or whether the apparent congestion is caused by a failing service dependency. A dependable architecture therefore couples learned intelligence with explicit operational policy.

Three findings are significant. First, topology-aware representations are central because data-center networks are graphs; RouteNet-style models preserve structure when reasoning about latency, loss, and routing (Rusek et al., 2020; Ferriol-Galmés et al., 2023). Second, sequential control needs safe boundaries: routing, consolidation, and cooling studies show that reward design and environment mismatch can produce unsafe behavior (Shaw et al., 2022; Zhang et al., 2023). Third, autonomy depends on assurance latency. Hierarchical assurance can allow pre-approved low-risk actions to execute quickly while exceptional or high-impact actions escalate.

The proposed framework therefore treats intelligence as probabilistic and policy as authoritative. Models produce predictions, anomaly scores, recommended actions, or value estimates; policy determines whether those outputs can become changes. This distinction also supports explainability. Operators do not always need a complete mathematical explanation of a deep model, but they do need to know which signals changed, which service objective is threatened, why an action is authorized, which constraints were checked, what outcome is expected, and how the system will revert if verification fails.

The framework also addresses a measurement gap. Algorithmic studies often report accuracy, reward, latency, energy, or throughput, while operators prioritize service-level violations, change failure rate, recovery time, rollback success, false-action rate, and operational workload. Evaluations should therefore measure both model and control quality. Slightly lower predictive accuracy may be preferable when a model is stable under drift, calibrated, fast, interpretable, and easy to constrain.

Figure 2 illustrates the resulting assurance-oriented decision cycle. Telemetry is transformed into state estimates; multiple models generate predictions and candidate actions; a policy gate checks authorization and risk; and a twin or surrogate model evaluates likely consequences when

required. Approved actions are executed, outcomes are verified, and evidence returns to the observation layer. Uncertain or high-risk decisions can abstain, fall back to a deterministic baseline, or escalate to human review.

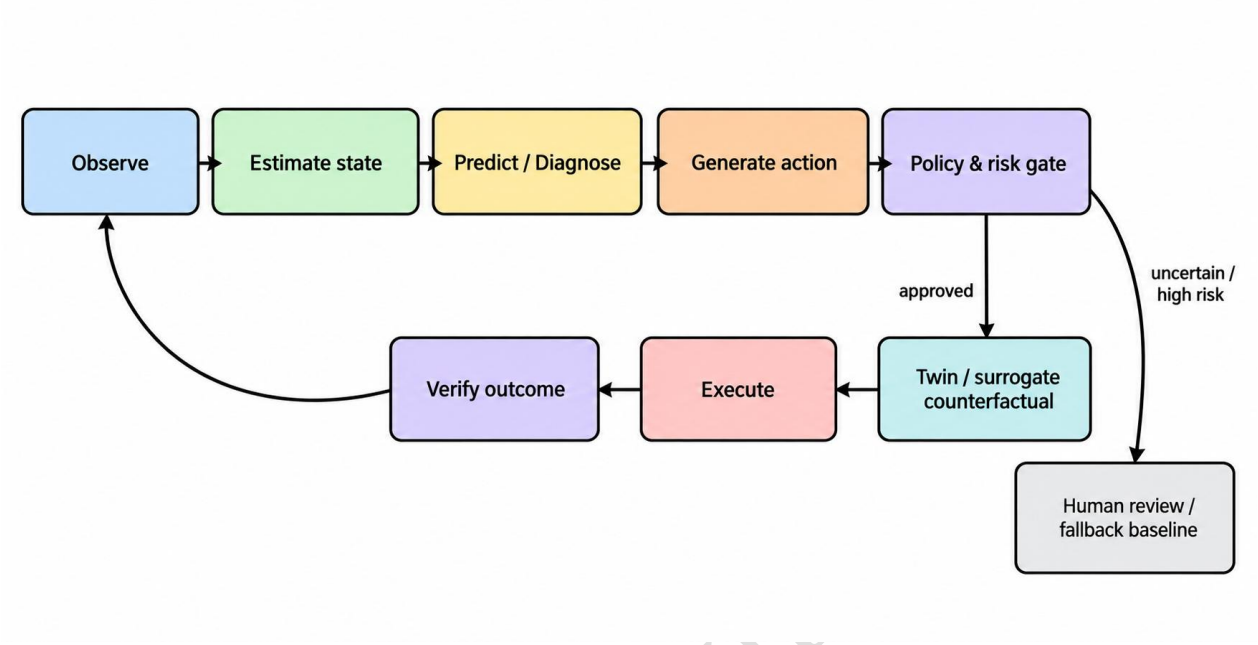


*Figure 2. Assurance-oriented closed-loop decision cycle for autonomous data-center operations.*

Table 2 summarizes deployment barriers and corresponding controls. Concept drift requires continuous performance monitoring and retraining triggers. Sparse labels require self-supervised learning, weak supervision, or active expert feedback. Topology changes require graph-aware validation. Reinforcement-learning risk requires constrained action spaces and safe baselines. Delayed telemetry requires freshness checks. Security requires model integrity and provenance controls. These mechanisms convert model lifecycle management into an operational discipline rather than a periodic data-science activity.

| **Barrier** | **Operational failure mode** | **Required control** | **Evidence / indicator** |
|---|---|---|---|
| Concept drift | Stale thresholds or policies | Drift monitoring and retraining triggers | Calibration error, feature shift |
| Sparse labels | Missed rare failures | Self-supervision and expert feedback | Rare-event coverage / review yield |
| Topology / OOD change | Invalid performance estimates | Graph-aware OOD checks and staged | Topology novelty / residual error |

| | | | |
|---|---|---|---|
| | | validation | |
| Unsafe RL behavior | Reward exploitation / risky action | Constrained action space, baseline, rollback | Constraint violations / false actions |
| Delayed/noisy telemetry | Action on stale state | Freshness checks and reconciliation | Timestamp skew / missing-data rate |
| Weak explainability | Slow or incorrect operator response | Action rationale and impact explanation | Review time / override rate |
| Cross-domain coupling | Local optimization harms global service | Hierarchical coordination / precedence | SLO-energy-resilience outcomes |
| Model/security compromise | Manipulated decisions | Signed artifacts, provenance, access control | Integrity / unauthorized changes |

*Table 2. Deployment barriers, failure modes, and assurance controls.*

## Research Gaps and Future Research

The largest research gap is production-grade evidence. Many studies use simulation, public traces, or bounded testbeds, which are essential for repeatability but do not reproduce the full uncertainty of production networks. Future research should publish longitudinal evaluations showing how models behave after software upgrades, topology changes, traffic regime shifts, and hardware failures. Benchmarking should include drift, missing telemetry, delayed measurements, and adversarial or corrupted inputs rather than only clean steady-state data.

A second gap concerns multi-objective and multi-domain control. Routing, compute placement, energy, cooling, and resilience are often optimized separately despite strong coupling. Research should investigate hierarchical or multi-agent architectures with explicit coordination contracts so that local optimization does not degrade global service objectives. The challenge is not merely to combine rewards, but to define precedence among hard safety constraints, service guarantees, sustainability targets, and cost objectives.

Third, uncertainty calibration and abstention require more attention. Autonomous models should expose when they are outside their training support and should degrade gracefully to conservative behavior. Future studies should compare uncertainty estimators, conformal

approaches, out-of-distribution detectors, and risk-sensitive decision rules using operational failure costs rather than only statistical calibration.

Fourth, digital twins need lifecycle research. Their value depends on synchronization, fidelity, validation, and computational cost. A stale or overconfident twin can become another source of automation error. Research should establish measurable fidelity thresholds and determine when learned surrogates, discrete-event simulation, or hybrid physics-data models are appropriate.

Finally, explainability must be connected to operator action. Generic saliency maps are rarely enough. Explanations should identify affected services, causal hypotheses, policy constraints, expected impact, and rollback criteria. Human factors research is needed to determine which explanation formats improve decision quality during incidents without creating alert overload.

A further gap concerns causal diagnosis. Most learning systems identify correlation or optimize behavior without proving why an incident occurred. Future work should combine structural network knowledge, change-event history, service dependencies, and interventions to distinguish causes from coincident symptoms. Causal evidence could reduce unnecessary remediation and improve confidence in autonomous rollback, especially during cascading failures where several metrics deteriorate simultaneously.

**Practical, Managerial and Policy Implications**

For practitioners, implementation should begin with observability quality rather than model sophistication. Telemetry schemas, timestamps, topology identifiers, configuration versions, and service ownership data must be consistent before cross-domain learning can be reliable. Organizations can then introduce autonomy progressively: first through recommendations, then bounded automated remediation, and finally wider closed-loop control for action classes with proven safety histories.

For managers, model governance should be integrated into change management. Each production model should have an owner, training-data lineage, validation record, approved action scope, performance thresholds, retraining criteria, and rollback procedure. Operational success should be measured through reliability and service outcomes rather than model accuracy alone. Investments in digital twins, policy engines, and reproducible test environments may deliver more dependable autonomy than repeatedly replacing models with marginally more accurate architectures.

For policy and risk functions, accountability must remain explicit. Autonomous infrastructure should produce tamper-evident records of observed state, model version, proposed action, applicable policy, approving authority, execution result, and rollback. High-impact actions should

be subject to segregation of duties and predetermined escalation. These requirements preserve auditability without preventing low-risk automation from operating at machine speed.

Procurement and architecture decisions should also avoid locking autonomy to a single vendor model. Interfaces for telemetry, policy evaluation, model serving, simulation, and rollback should remain modular so that components can be independently validated or replaced. This supports auditability and reduces the risk that an opaque end-to-end platform becomes impossible to govern as infrastructure evolves. This modularity also enables independent assurance teams to test models against standardized failure scenarios before expanded automation privileges are granted in production.

## Limitations

This review is integrative rather than meta-analytic, and the included literature uses heterogeneous datasets, metrics, topologies, and simulation assumptions. Direct numerical comparison between studies would therefore be misleading. The review emphasizes publications from 2020-2025, so some foundational earlier work is not discussed in detail. Rapid development in programmable networks, accelerator fabrics, and large-model-based operations may also change the technology landscape after the review period. Finally, the proposed framework is conceptual; its value should be tested through implementation across heterogeneous production environments and compared against established automation baselines.

## Conclusion

Machine learning can support autonomous data-center operations most effectively when it is embedded in a disciplined closed-loop architecture rather than deployed as an isolated predictor or controller. The literature shows credible capabilities in anomaly detection, traffic classification, topology-aware performance modeling, forecasting, routing, resource allocation, energy optimization, and adaptive control. At the same time, the evidence repeatedly exposes limitations arising from concept drift, uncertain transferability, incomplete labels, unsafe exploration, cross-domain coupling, and the difference between simulation success and production assurance.

The Machine Learning-Based Network Intelligence Framework proposed in this review organizes these capabilities into observation, representation, intelligence, decision, actuation, and assurance layers. Graph neural networks provide a natural mechanism for representing network relationships; time-series and anomaly models convert telemetry into predictive and diagnostic signals; reinforcement learning supports sequential optimization; intent and policy mechanisms

constrain allowable outcomes; and digital twins provide a means to test higher-risk changes before execution. These components are complementary rather than interchangeable.

The principal implication is that autonomy should be defined by controlled behavior, not by the presence of artificial intelligence. A system becomes operationally autonomous only when it can observe relevant state, recognize uncertainty, select an authorized action, execute within safe boundaries, verify the result, and recover when assumptions fail. Future research should therefore emphasize production evidence, cross-domain coordination, calibrated uncertainty, assurance latency, and operator-centered explanations. The most credible path to future-ready data centers is progressive autonomy: expand automated authority only as evidence demonstrates that learning, policy, simulation, rollback, and governance operate coherently.

Accordingly, autonomy should be treated as an engineering property of the complete operational control system, demonstrated through evidence of safe behavior under both routine and exceptional conditions.

## References


1. Liu, W.-X., Cai, J., Chen, Q. C., & Wang, Y. (2021). DRL-R: Deep reinforcement learning approach for intelligent routing in software-defined data-center networks. Journal of Network and Computer Applications, 177, 102865. https://doi.org/10.1016/j.jnca.2020.102865
2. Rikhtegar, N., Bushehrian, O., & Keshtgari, M. (2021). DeepRLB: A deep reinforcement learning-based load balancing in data center networks. International Journal of Communication Systems, 34(15), e4912. https://doi.org/10.1002/dac.4912
3. Xie, S., Hu, G., Wang, X., Xing, C., & Liu, Y. (2022). A decision tree-based online traffic classification method for QoS routing in data center networks. Security and Communication Networks, 2022, 9419676. https://doi.org/10.1155/2022/9419676
4. Wang, Y., Li, Y., Wang, T., & Liu, G. (2022). Towards an energy-efficient data center network based on deep reinforcement learning. Computer Networks, 210, 108939. https://doi.org/10.1016/j.comnet.2022.108939
5. Yang, Z., Du, J., Lin, Y., Du, Z., Xia, L., Zhao, Q., & Guan, X. (2022). Increasing the energy efficiency of a data center based on machine learning. Journal of Industrial Ecology. https://doi.org/10.1111/jiec.13155
6. Khan, W., De Chiara, D., Kor, A.-L., & Chinnici, M. (2023). Advanced data analytics modeling for evidence-based data center energy management. Physica A: Statistical Mechanics and its Applications, 624, 128966. https://doi.org/10.1016/j.physa.2023.128966
7. Mahbod, M. H. B., Chng, C. B., Lee, P. S., & Chui, C. K. (2022). Energy saving evaluation of an energy efficient data center using a model-free reinforcement learning approach. Applied Energy, 322, 119392. https://doi.org/10.1016/j.apenergy.2022.119392
8. Shaw, R., Howley, E., & Barrett, E. (2022). Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers. Information Systems, 107, 101722. https://doi.org/10.1016/j.is.2021.101722
9. Hu, X., & Sun, Y. (2020). A deep reinforcement learning-based power resource management for fuel cell powered data centers. Electronics, 9(12), 2054. https://doi.org/10.3390/electronics9122054
10. Zhang, Q., Zeng, W., Lin, Q., Chng, C.-B., Chui, C.-K., & Lee, P.-S. (2023). Deep reinforcement learning towards real-world dynamic thermal management of data centers. Applied Energy, 333, 120561. https://doi.org/10.1016/j.apenergy.2022.120561
11. Wang, R., Cao, Z., Zhou, X., Wen, Y., & Tan, R. (2023). Phyllis: Physics-informed lifelong reinforcement learning for data center cooling control. Proceedings of the 14th

ACM International Conference on Future Energy Systems, 114-126. https://doi.org/10.1145/3575813.3595189

12. Cen, J., & Li, Y. (2022). Resource allocation strategy using deep reinforcement learning in cloud-edge collaborative computing environment. Mobile Information Systems, 2022, 9597429. https://doi.org/10.1155/2022/9597429
13. Xu, J., Xu, Z., & Shi, B. (2022). Deep reinforcement learning based resource allocation strategy in cloud-edge computing system. Frontiers in Bioengineering and Biotechnology, 10, 908056. https://doi.org/10.3389/fbioe.2022.908056
14. Yang, D., Yu, J., Du, X., He, Z., & Li, P. (2022). Energy saving strategy of cloud data computing based on convolutional neural network and policy gradient algorithm. PLOS ONE, 17(12), e0279649. https://doi.org/10.1371/journal.pone.0279649
15. Duan, Y., Li, C., Bai, G., Chen, G., Zhou, F., Chen, J., Gao, Z., Sun, H., & Zhang, C. (2023). MFGAD-INT: In-band network telemetry data-driven anomaly detection using multi-feature fusion graph deep learning. Journal of Cloud Computing, 12, 126. https://doi.org/10.1186/s13677-023-00492-w
16. Jain, M., Kaur, G., & Saxena, V. (2022). A K-Means clustering and SVM based hybrid concept drift detection technique for network anomaly detection. Expert Systems with Applications, 193, 116510. https://doi.org/10.1016/j.eswa.2022.116510
17. Tang, Z., Hu, H., & Xu, C. (2022). A federated learning method for network intrusion detection. Concurrency and Computation: Practice and Experience, 34(10), e6812. https://doi.org/10.1002/cpe.6812
18. Duan, X., Ying, S., Yuan, W., Cheng, H., & Yin, X. (2021). QLLog: A log anomaly detection method based on Q-learning algorithm. Information Processing & Management, 58(3), 102540. https://doi.org/10.1016/j.ipm.2021.102540
19. Li, G., & Jung, J. J. (2023). Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. Information Fusion, 91, 93-102. https://doi.org/10.1016/j.inffus.2022.10.008
20. Audibert, J., Michiardi, P., Guyard, F., Marti, S., & Zuluaga, M. A. (2022). Do deep neural networks contribute to multivariate time series anomaly detection? Pattern Recognition, 132, 108945. https://doi.org/10.1016/j.patcog.2022.108945
21. Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C. C., & Salehi, M. (2024). Deep learning for time series anomaly detection: A survey. ACM Computing Surveys, 57(1), Article 15. https://doi.org/10.1145/3691338
22. Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P., & Cabellos-Aparicio, A. (2020). RouteNet: Leveraging graph neural networks for network modeling and

optimization in SDN. IEEE Journal on Selected Areas in Communications, 38(10), 2260-2270. https://doi.org/10.1109/JSAC.2020.3000405

23. Ferriol-Galmés, M., Paillissé, J., Suárez-Varela, J., Rusek, K., Xiao, S., Shi, X., Cheng, X., Barlet-Ros, P., & Cabellos-Aparicio, A. (2023). RouteNet-Fermi: Network modeling with graph neural networks. IEEE/ACM Transactions on Networking, 31(6), 3080-3095. https://doi.org/10.1109/TNET.2023.3269983
24. Tam, P., Song, I., Kang, S., Ros, S., & Kim, S. (2022). Graph neural networks for intelligent modelling in network management and orchestration: A survey on communications. Electronics, 11(20), 3371. https://doi.org/10.3390/electronics11203371
25. Waikhom, L., & Patgiri, R. (2023). A survey of graph neural networks in various learning paradigms: Methods, applications, and challenges. Artificial Intelligence Review, 56, 6295-6364. https://doi.org/10.1007/s10462-022-10321-2
26. Zhou, Y., Zheng, H., Huang, X., Hao, S., Li, D., & Zhao, J. (2022). Graph neural networks: Taxonomy, advances, and trends. ACM Transactions on Intelligent Systems and Technology, 13(1), Article 15, 1-54. https://doi.org/10.1145/3495161
27. Pang, L., Yang, C., Chen, D., Song, Y., & Guizani, M. (2020). A survey on intent-driven networks. IEEE Access, 8, 22862-22873. https://doi.org/10.1109/ACCESS.2020.2969208
28. 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
29. 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
30. Si-Mohammed, S., Bardou, A., Begin, T., Guérin Lassous, I., & Vicat-Blanc, P. (2024). NS+NDT: Smart integration of network simulation in network digital twin, application to IoT networks. Future Generation Computer Systems, 157, 124-144. https://doi.org/10.1016/j.future.2024.03.038