Self-Running Networks: A Comprehensive Survey of Foundations, Applications, and Challenges

Shaghayegh Shajarian[1], Sajad Khorsandroo[1], and Mahmoud Abdelsalam[1]

[1]North Carolina Agricultural and Technical State University

July 22, 2025

Abstract

Self-running networks represent a groundbreaking paradigm for achieving fully autonomous network infrastructures capable of self-configuration, self-optimization, self-healing, and self-protection without human intervention. Although numerous isolated studies have explored aspects of self-running networks, an integrated and holistic overview is still lacking. This survey addresses this gap by providing a comprehensive system-level exposition, formalizing the concept of self-running networks, and detailing the architectural components that enable end-to-end autonomy through a unified seven-layer reference model. We also analyze the key self-* functionalities, including their core mechanisms, operational challenges, and representative application domains. Furthermore, we propose a six-level network autonomy maturity model to evaluate the evolution of network intelligence from manual operations to fully autonomous systems. We further synthesize foundational paradigms and practical implementations, providing a consolidated view of current advancements. In addition, we discuss critical challenges in self-running networks, including security, scalability, interoperability, and ethical considerations. Finally, we highlight research directions and open issues to guide future innovations and accelerate the deployment of self-running networks in real-world environments. This work serves as a conceptual and practical reference for researchers, practitioners, and industry stakeholders aiming to design, deploy, and advance next-generation self-running networks.

Self-Running Networks: A Comprehensive Survey of Foundations, Applications, and Challenges

Shaghayegh Shajarian, *Graduate Student Member, IEEE*, Sajad Khorsandroo, and Mahmoud Abdelsalam,
Department of Computer Science, College of Engineering, North Carolina A&T State University, USA

***Abstract*—Self-running networks represent a groundbreaking paradigm for achieving fully autonomous network infrastructures capable of self-configuration, self-optimization, self-healing, and self-protection without human intervention. Although numerous isolated studies have explored aspects of self-running networks, an integrated and holistic overview is still lacking. This survey addresses this gap by providing a comprehensive system-level exposition, formalizing the concept of self-running networks, and detailing the architectural components that enable end-to-end autonomy through a unified seven-layer reference model. We also analyze the key self-* functionalities, including their core mechanisms, operational challenges, and representative application domains. Furthermore, we propose a six-level network autonomy maturity model to evaluate the evolution of network intelligence from manual operations to fully autonomous systems. We further synthesize foundational paradigms and practical implementations, providing a consolidated view of current advancements. In addition, we discuss critical challenges in self-running networks, including security, scalability, interoperability, and ethical considerations. Finally, we highlight research directions and open issues to guide future innovations and accelerate the deployment of self-running networks in real-world environments. This work serves as a conceptual and practical reference for researchers, practitioners, and industry stakeholders aiming to design, deploy, and advance next-generation self-running networks.**

***Index Terms*—Self-Running Networks, Network Management, Autonomous Networks, Self-Configuration, Self-Optimization, Self-Healing, Self-Protection, Network Intelligence.**

I. Introduction

In the early days of computer networking, networks were static, and tasks such as configuration, monitoring, and troubleshooting relied heavily on human intervention. However, as network complexity grew in the late 20th century, traditional, static approaches could no longer deliver the performance, security, and reliability required in dynamic environments. To address these limitations, efforts toward intelligent network management began in 1989 with attempts to apply Artificial Intelligence (AI) for optimizing end-to-end Quality of Service (QoS) and Quality of Experience (QoE) [1], [2]. Despite initial optimism, the computational resources and algorithmic capabilities were insufficient to handle the complexities and dynamic nature of modern networks. A significant leap occurred in 2001 when IBM's eLiza project introduced the concept of autonomic computing. Inspired by the autonomic nervous system, this project's goal was to create networks capable of self-configuration, self-optimization, self-healing, and self-protection, thereby reducing reliance on human intervention [3], [4]. Although eLiza laid the foundation for network automation, technical and implementation challenges hindered its adoption [5].

The late 2000s saw the emergence of Software-Defined Networking (SDN), which decoupled the control and data planes to simplify network management and introduced centralized programmability that enables more flexible and efficient configurations [6]. This shift was reinforced in the early 2010s by advancements in data plane programmability, notably through languages like P4 [7], which enabled dynamic protocol deployment and real-time customization of packet processing within devices. As network softwarization matured [8], [9], and full-stack programmability (aka deep programmability) became practical [10], the focus shifted to enabling networks to make decisions autonomously. In spite of supporting dynamic changes, these platforms lacked the intelligence to interpret real-time conditions and respond without human input [11].

Only with recent advances in AI, Machine Learning (ML), and dedicated compute hardware such as graphics processing units and tensor processing units has it become possible to build networks that can analyze conditions, learn from data, and act in real time [12]. Modern autonomous-network designs organize these capabilities into a closed-loop control pipeline that continuously collects telemetry, analyzes it, evaluates policy, and enforces actions [13], [14]. We refer to this paradigm as a *'self-running network'*, a system that observes its own condition, reasons about it, and applies corrective measures without human intervention [15]. This paradigm forms the foundation of autonomous, next-generation networks. Figure 1 captures milestones in this field at a glance and shows how each step brought us closer to self-running networks.

A. Motivation

The complexity of modern networks requires solutions that move beyond traditional network management practices. Self-running networks have emerged as an advancement, promising fully autonomous functionalities that remove human intervention and enable networks to self-configure, self-optimize, self-heal, and self-protect. Despite this promise, the study of self-running networks is fragmented across multiple domains, lacking a comprehensive and unified perspective that synthesizes these advances and provides a roadmap for the future. Several academic projects have made strides toward self-running networks. For instance, the SOCRATES Project [16] focuses on self-organization for wireless access networks, while the GP4P4 Framework [17] introduces genetic programming for autonomous intent-based network programming, and the Self-

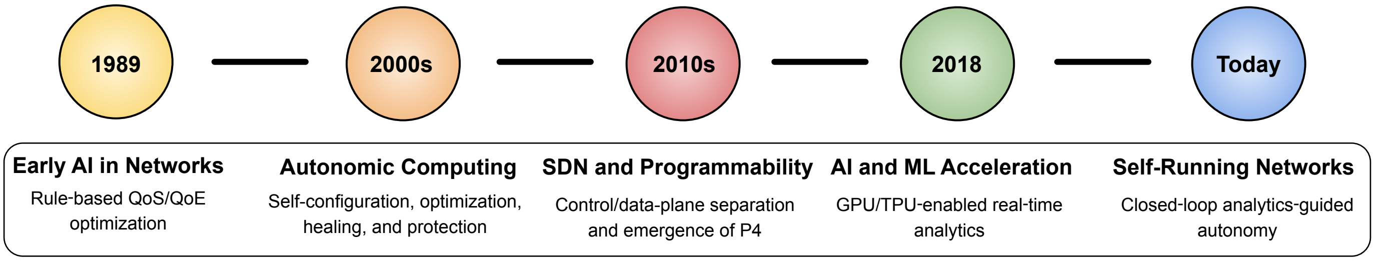
1989
2000s
2010s
2018
Today
Early AI in Networks
Rule-based QoS/QoE optimization
Autonomic Computing
Self-configuration, optimization, healing, and protection
SDN and Programmability
Control/data-plane separation and emergence of P4
AI and ML Acceleration
GPU/TPU-enabled real-time analytics
Self-Running Networks
Closed-loop analytics-guided autonomy

Fig. 1: A chronological overview of major developments in network management

Driving Science Network Project at Lawrence Berkeley National Laboratory [18] explores AI-driven network automation. The SELFNET Project [19] addresses self-management in future internet architectures.

Complementing these academic initiatives, industry has introduced significant advancements. Huawei, at Mobile World Congress 2018, proposed autonomous driving networks [20], leveraging a digital replica to align physical networks with business goals and facilitate the transition from SDN to full autonomy. Juniper Networks introduced Mist AI [21], which integrates AI, ML, and data science to optimize user experiences and simplify operations across wireless access, wired access, and Software-defined Wide Area Networks (SD-WAN) domains. In 2023, Nokia introduced its Digital Operations Center, a modular platform that combines orchestration, assurance, and unified inventory, to create a live network digital twin that drives closed-loop, AI-driven service and slice management, autonomously detecting and resolving issues before they impact service-level agreement [22]. Table I summarizes these flagship initiatives. The fragmentation of these works creates a critical gap in understanding how these innovations can be integrated into a cohesive framework applicable to diverse environments. Accordingly, this survey addresses this gap by presenting the first comprehensive, unified study on self-running networks, their components, functionalities, mechanisms, applications, and challenges. Our work is intended as a resource for network researchers, practitioners, and industry stakeholders aiming to understand, design, and implement self-running networks.

B. Prior Works

To date, there is no comprehensive effort that brings together work related to self-running networks and organizes these contributions into a cohesive, end-to-end vision. There are survey papers that focus on some aspects of these networks. This section reviews key surveys and clarifies how our work advances the state of the art, as shown in Table II, summarizing prior studies. Early foundational works on self-* functionalities (i.e., self-configuration, self-optimization, self-healing, and self-protection) as important aspects of next-generation networks were presented by Dobson et al. [23] and Movahedi et al. [24], who surveyed autonomic communication and proposed paradigms for these functionalities. While these works laid the conceptual groundwork, they predate key developments such as SDN and modern ML, and thus fall short of addressing the requirements for real-time autonomy.

ML as an enabler of network intelligence is addressed in surveys by Boutaba et al. [25] and Luong et al. [26]. The former covers a wide range of ML techniques applied to traffic prediction, QoS optimization, and anomaly detection, while the latter focuses on Deep Reinforcement Learning (DRL) for dynamic decision-making tasks like routing and access control. Although both surveys demonstrate the potential of ML in network intelligence, they focus on algorithmic tools without studies on other components and operations of networks. High-level Paradigms such as Intent-Based Networking (IBN) and Zero-touch network and Service Management (ZSM) have also drawn attention. Surveys by Pang et al. [27] and Leivadeas et al. [28] review IBN architectures, policy translation mechanisms, and lifecycle management strategies. Similarly, Coronado et al. [29] and Liyanage et al. [30] study the ETSI ZSM framework and its relevance for 5G service. However, both IBN and ZSM are treated as isolated paradigms rather than as subsystems within broader networks.

Recent efforts have also explored autonomy in specialized network contexts. Bai et al. [31] survey DRL applications in multi-UAV wireless networks, where dynamic environments demand decentralized, closed-loop coordination. Zuo et al. [32] examine the fusion of blockchain and AI for secure, autonomous decision-making in 6G networks. While these works illustrate advanced capabilities in niche scenarios, they are narrow in scope and do not offer generalized architectural models or benchmarks for broader network autonomy. Hence, the following key limitations are unaddressed:

- **Lack of an Integrated System-Level Model:** Prior surveys address network autonomy in a piecemeal manner, focusing either on isolated algorithmic tasks or abstract architectural paradigms, and they lack a unified conceptual model that captures the full autonomy cycle.
- **Underspecification of Self-* Functionalities:** While self-configuration, self-optimization, self-healing, and self-protection are mentioned, prior works do not systematically define how these operate across the network lifecycle.
- **Fragmented Research Landscape Lacking Synthesis:** Prior literature is highly fragmented, split across disciplines (e.g., AI, telecom, cloud) and problem areas, with little effort to categorize findings or reveal cross-cutting insights.
- **Absence of Autonomy Maturity Framework:** Existing surveys highlight advancements but do not offer a structured way to evaluate or compare autonomy levels. This leaves research and industry without a shared vocabulary

for progress.

C. Contributions of our Survey Paper

Aligned with the motivation and gaps from prior works outlined in Section I-A and I-B, our primary contributions are as follows:

- We propose a conceptual framework that integrates interrelated layers, from monitoring and data collection to feedback and continuous improvements, each with distinct components. This model offers a structured foundation for implementing self-running networks.
- We specify how self-* functionalities (i.e., self-configuration, self-optimization, self-healing, and self-protection) operate throughout the network lifecycle, summarizing their mechanisms, challenges, and domain-specific applications.
- We introduce a six-level autonomy maturity model that delineates the progression from fully manual to fully autonomous networks. This model serves as a roadmap for network operators and designers seeking to evaluate the autonomy of their deployments.
- We consolidate existing research into two main categories: fundamental paradigms and practical applications. This synthesis provides a coherent view of the current state of the art, revealing trends, gaps, and opportunities in the pursuit of self-running networks.
- We conduct a thorough examination of key challenges, including security, interoperability, and scalability, that impede real-world adoption. Additionally, we explore promising directions and outline their potential and limitations to address existing challenges and accelerate the transition toward deployable self-running networks.

D. Organization of the Paper

As depicted in Figure 2, the remainder of the paper is organized as follows. Section II formalizes the concept of self-running networks by presenting a conceptual model and describing each architectural layer together with its constituent components. Section III analyzes the four fundamental functionalities of self-running networks, highlighting their roles, underlying mechanisms, operational challenges, and representative application domains. Section IV surveys current research and development efforts aimed at realizing self-running networks. Section V discusses the open challenges and key considerations that must be addressed before large-scale deployment. Finally, Section VI summarizes the paper's contributions and outlines directions for future work.

II. Self-Running Networks: A Conceptual Model and Components

Self-running (or "autonomous") networks represent a major evolution in network management. These networks integrate multiple components to monitor, analyze, and adapt to conditions in real time without human intervention. In line with IBM's widely adopted MAPE-K loop framework for autonomic systems [4], this section presents a conceptual model for a self-running network environment. As shown in Figure 3, the model comprises several interrelated layers, each with distinct components. The model operates in a cyclical, closed-loop process to enable end-to-end automation and intelligence. To complement this visual representation, Table III provides a detailed summary of each layer's core components and their respective functions within self-running networks. While exact implementations may vary,[*] each layer in this model contributes to a closed-loop process, where continuous monitoring drives automated decisions and actions.

A. Monitoring and Data Collection Layer

By collecting real-time metrics, this layer offers a dynamic view of network conditions, including traffic flows and potential issues. This layer includes two components: *Sensors and Probes*, and *Data Aggregation*.

1) Sensors and Probes: Strategically deployed across network endpoints, switches, routers, and other network elements, these systems capture real-time metrics such as latency, bandwidth usage, packet loss, CPU and memory utilization, and security logs. They operate across multiple layers, including physical, data link, network, transport, and application. They are designed for minimal performance impact on the underlying devices, and they balance the need for detailed monitoring with any possible resource constraints.

2) Data Aggregation: Telemetry collected from various sources (e.g., Sonata [33]) is aggregated into centralized repositories or streaming frameworks such as Kafka [34], Spark [35], or similar big-data platforms. This process includes formatting, time-stamping, filtering, and compressing the data to reduce noise and storage overhead. While this layer ensures high-quality telemetry collection and basic filtering, the responsibility for interpreting this data lies with the subsequent analysis and intelligence layer.

B. Data Analysis and Intelligence Layer

This layer leverages advanced analytics and AI models to convert data into insights, enabling automated or semi-automated actions. The layer consists of three key components: the *Analytics Engine*, *Pattern Recognition*, and *Decision Support*.

1) Analytics Engine: Utilizing statistical methods (e.g., linear regression for capacity planning [36]), along with ML and AI algorithms (e.g., Deep Learning (DL) frameworks for complex network trace processing [37], [38]), this component processes real-time and historical data. It handles large volumes of network metrics, generating key performance indicators, classifying network states, and producing predictive scores to estimate the risk of congestion, failure, or performance degradation.

[*] Different self-running network architectures may structure or name layers and components differently, depending on specific requirements and design philosophies. Besides, terms like "analytics," "intelligence," or "orchestration" may overlap or be subdivided into alternative frameworks. Despite these variations, the principle of a closed-loop approach, where continuous monitoring drives automated decisions and actions, remains consistent across all self-running network models.

TABLE I: Flagship Academic and Industrial Initiatives Advancing Self-Running Networks

Category	Institution/Organization	Project	Year	Primary Focus	Key Autonomous Capability
Academic	EU's 7th Framework Programme	SOCRATES [16]	2008	Self-organizing LTE RAN	On-device learning for channel/radio-parameter selection
	Horizon 2020 (H2020) consortium	SELFNET [19]	2017	Autonomic NFV/SDN management stack	Cross-layer self-healing and self-protection
	Delft University of Technology	GP4P4 [17]	2019	Genetic programming to generate P4 data-plane programs from high-level intents	Fully automated intent→P4-rule compilation without human coding
	Lawrence Berkeley Nat. Lab	Self-Driving Science Network [18]	2020	High-throughput scientific WAN transfers	ML-driven closed-loop traffic steering with anomaly handling
Industry	Huawei	Intelligent Driving Networks [20]	2018	End-to-end mobile service assurance via a live digital twin	AI-driven policy optimisation ("Autonomy by Layer")
	Juniper	Mist AI [21]	2019	Campus and SD-WAN operations	Root-cause analysis + proactive anomaly remediation
	Nokia	Digital-Twin Network Ops [22]	2023	Digital-twin network modeling and simulation	AI-powered predictive slice optimization ahead of demand peaks

Legend: Projects are grouped by category (Academic or Industry); citations next to project names indicate the primary reference for each initiative.

TABLE II: Summary of Related Surveys versus our Survey

Year	Survey	Contribution	Research Dimensions			
			System-Level Model	Self-* Functionalities Specification	Research Synthesis Across Domains	Autonomy Maturity Framework
2006	Dobson et al. [23]	Early conceptualization of autonomic communication and self-* functionalities	✓	✓	✗	✗
2011	Movahedi et al. [24]	Architectural overview of autonomic systems pre-ML and pre-SDN	✓	✓	✗	✗
2018	Boutaba et al. [25]	Machine learning approaches applied to traditional network management	✗	✓	✓	✗
2019	Luong et al. [26]	Deep RL-based automation techniques for dynamic decision-making across diverse networking tasks	✗	✓	✓	✗
2020	Pang et al. [27]	Intent-based networking models including IBN lifecycle integration for automated management	✓	✓	✗	✗
2022	Leivadeas et al. [28]	IBN frameworks and automation pipelines in software-defined networks	✓	✓	✗	✗
2022	Coronado et al. [29]	ETSI-aligned ZSM framework for 5G-specific network automation	✓	✓	✗	✗
2022	Liyanage et al. [30]	Taxonomy of ZSM concepts, including automation challenges, and orchestration enablers in 5G and beyond networks	✓	✓	✗	✗
2023	Bai et al. [31]	RL-based closed-loop control for multi-UAV wireless networks with self-optimizing coordination and trajectory adaptation	✗	✗	✓	✗
2023	Zuo et al. [32]	Synergistic use of blockchain and AI for secure, intelligent automation in 6G wireless networks	✗	✗	✓	✗
2025	**Our Survey**	**Proposes a conceptual model for Self-Running Networks that integrates telemetry, analytics, intent, verification, and enforcement across domains, and specifies the networks' key components, functionalities, and associated challenges**	✓	✓	✓	✓

Legend: ✓= addressed; ✗= not addressed.

2) Pattern Recognition: This component identifies trends in network usage, traffic flow, and resource consumption, enabling proactive responses to emerging conditions. It detects anomalies that deviate from historical baselines, such as sudden spikes in latency or unusual changes in bandwidth usage, which may indicate security breaches or impending failures. Additionally, it predicts potential issues, such as link congestion or hardware degradation, before they escalate.

3) Decision Support: This component interprets analytical outputs and provides recommendations for possible network actions, such as rerouting traffic or adjusting QoS settings. It can operate in fully automated mode, where changes are executed without human intervention, or in semi-automated mode, allowing an administrator to review recommendations before implementation. It aligns decisions with high-level business and technical policies, ensuring that performance objectives are balanced with constraints such as cost, energy consumption, and regulatory compliance. The resulting recommendation is expressed at the intent level (e.g., "shift 15 % of video traffic off Link X"); translation into vendor-specific commands is delegated downstream to the next layer.

C. Policy and Orchestration Layer

This layer translates high-level business and service requirements into consistent, enforceable actions across the network infrastructure. It is composed of three components: *Policy Definition*, *Orchestration Engine*, and *Workflow Automation*.

1) Policy Definition: Network administrators define high-level objectives and constraints, such as QoS levels, security rules, and cost limits, using human-readable or domain-specific languages. These policies are dynamically updated as business needs evolve, which enables networks to adapt and remain aligned with organizational strategies.

2) Orchestration Engine: This component interprets policies and translates them into device-specific configurations and commands. It interfaces with various network components,

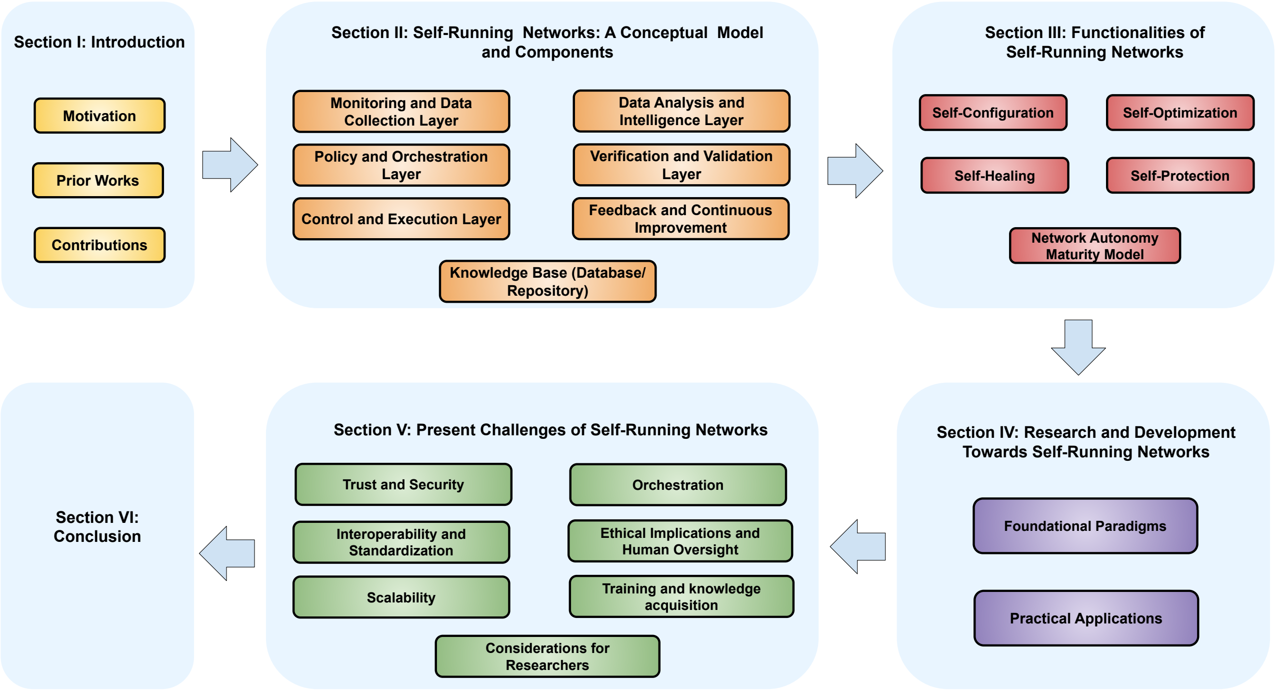
Section I: Introduction
Motivation
Prior Works
Contributions
Section II: Self-Running Networks: A Conceptual Model and Components
Monitoring and Data Collection Layer
Data Analysis and Intelligence Layer
Policy and Orchestration Layer
Verification and Validation Layer
Control and Execution Layer
Feedback and Continuous Improvement
Knowledge Base (Database/ Repository)
Section III: Functionalities of Self-Running Networks
Self-Configuration
Self-Optimization
Self-Healing
Self-Protection
Network Autonomy Maturity Model
Section IV: Research and Development Towards Self-Running Networks
Foundational Paradigms
Practical Applications
Section V: Present Challenges of Self-Running Networks
Trust and Security
Orchestration
Interoperability and Standardization
Ethical Implications and Human Oversight
Scalability
Training and knowledge acquisition
Considerations for Researchers
Section VI: Conclusion

Fig. 2: Overview of the organization of the survey

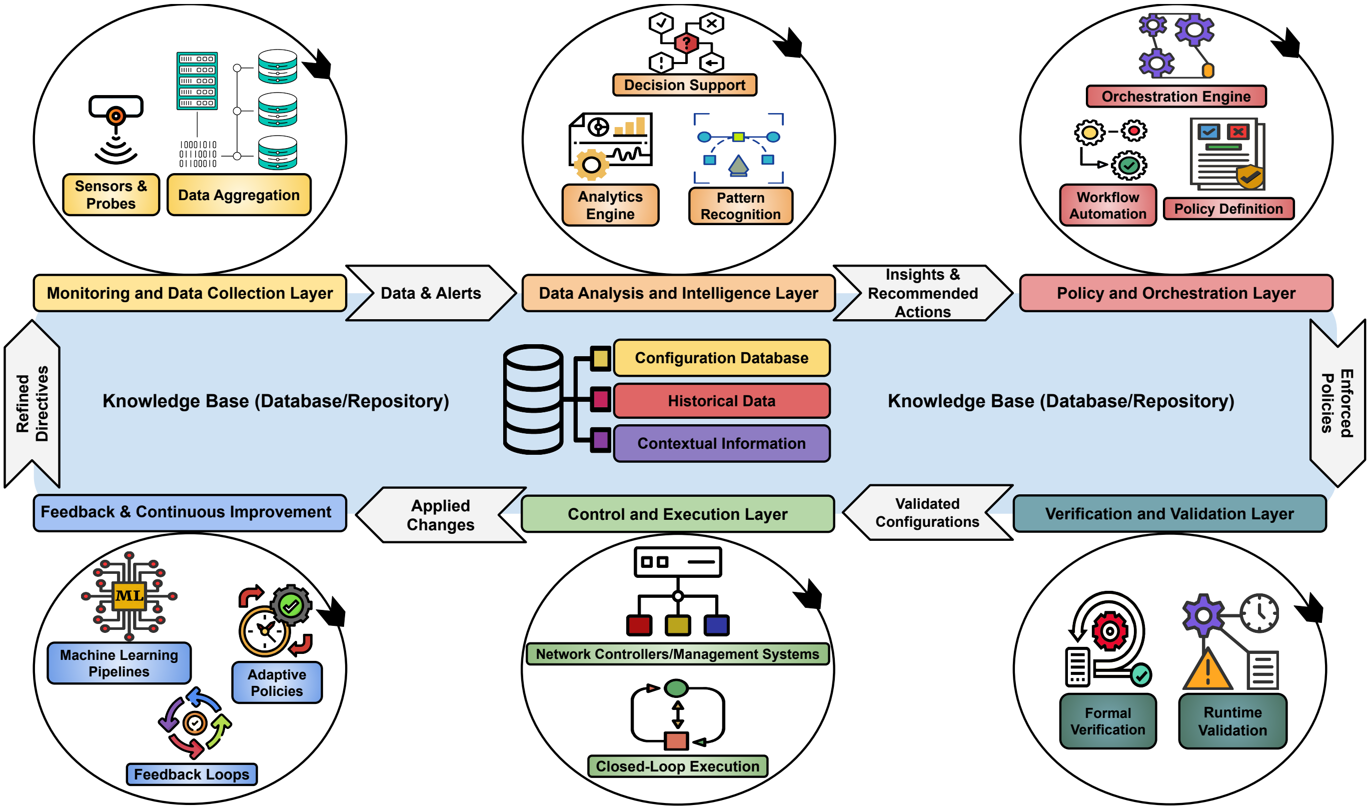

Fig. 3: Proposed self-running network conceptual model and its core components

such as controllers, routers, and virtualized services, to provision or modify resources in real time. To check interoperability across heterogeneous devices and platforms, it often leverages standardized APIs or protocols, such as NETCONF [39], REST [40], and gRPC [41].

TABLE III: Layers, Components, and Key Functions of the Self-Running Network Model

Layer	Components	Key Functions
Monitoring and Data Collection	Sensors and Probes; Data Aggregation	Capture, filter, and preprocess real-time metrics (e.g., traffic, latency, errors) from network elements.
Data Analysis and Intelligence	Analytics Engine; Pattern Recognition; Decision Support	Transform raw telemetry into insights: detect anomalies, predict conditions, and recommend actions.
Policy and Orchestration	Policy Definition; Orchestration Engine; Workflow Automation	Translate high-level intents into device-level configurations and automate multi-step workflows.
Verification and Validation	Formal Verification; Runtime Validation	Guarantee correctness and compliance of planned changes, both pre-deployment and in live operation.
Control and Execution	Network Controllers/Management Systems; Closed-Loop Execution	Programmatically enforce policies and apply configuration updates in real time.
Knowledge Base	Configuration Database; Historical Data; Contextual Information	Store and manage state, past metrics, and contextual metadata to inform analytics and decisions.
Feedback and Continuous Improvement	Feedback Loops; Machine Learning Pipelines; Adaptive Policies	Refine ML models and policies using post-action outcomes to close the automation loop.

3) Workflow Automation: This component coordinates and sequences network configuration tasks across a wide range of components, including physical infrastructure, SDN controllers, virtual machines, containers, and Internet of Things (IoT) endpoints. Automating routine or repetitive processes, such as pushing configuration changes or creating new service chains, reduces the likelihood of human error and accelerates deployment times. Tools like Ansible [42], Chef [43], or custom scripts are often employed to check consistency and traceability in network configurations.

D. Verification and Validation Layer

This layer acts as a safeguard to keep network functionality error-free and secure [44], [45]. By addressing the critical need for trust in autonomous systems, this layer confirms that the network operates dynamically while maintaining compliance with policies and performance objectives. The Verification and Validation Layer consists of two key components: *Formal Verification* and *Runtime Validation*.

1) Formal Verification: This component employs mathematical and logical methods to validate critical properties of the network. Its primary objectives include verifying reachability to make sure all endpoints can communicate without loops or black holes [46], validating that configurations align with security rules [47], service-level agreements, and regulatory constraints, and maintaining safety and liveness to guarantee stable and responsive network behavior under all conditions, including failures. Tools such as model checkers (e.g., SPIN [48], [49], NuSMV [50], [51]) and theorem provers (e.g., Z3 [52]) can be used to analyze configurations and planned changes in a controlled, pre-deployment environment. Model checking [53] explores the state space of configurations to check specific properties hold true, while theorem proving [54], [55] validates compliance with rules and constraints. In addition, symbolic execution [56]–[58] can complement these methods by examining execution paths and parameters to identify potential vulnerabilities and weaknesses in network configurations, logic, or design.

2) Runtime Validation: While formal verification validates configurations before deployment, runtime validation checks that real-time network operations adhere to predefined policies, service-level agreements, and high-level intents. It focuses on changes in configurations or operations, such as traffic rerouting, QoS updates, or resource reallocations, verifying compliance with policies and preventing unintended disruptions. For example, when a routing algorithm modifies traffic flows to alleviate congestion, runtime validation confirms adherence to security rules and QoS objectives, avoiding performance degradation or side effects. Also, runtime validation leverages IBN principles [59] and tools like Intent Verifier [60] to translate high-level intents (e.g., prioritizing traffic during critical events) into actionable configurations.

E. Control and Execution Layer

This layer is responsible for applying decisions made by analytics and policy systems to the physical and virtual infrastructure. It executes changes programmatically, allowing the network to respond in real time to dynamic conditions and high-level directives. It includes two key components: *Network Controllers/Management Systems* and *Closed-Loop Execution*.

1) Network Controllers/Management Systems: This component provides centralized programmatic control over network devices and services. Examples include SDN controllers such as OpenDaylight [61] and ONOS [62], which offer unified interfaces to manage switches, routers, and services like firewalls and load balancers. These systems use standardized protocols (e.g., OpenFlow [63]) and RESTful Application Programming Interfaces (APIs) to confirm interoperability across multiple vendors. Controllers expose northbound APIs to communicate with orchestration and policy layers, and southbound APIs to apply configuration updates to the underlying infrastructure.

2) Closed-Loop Execution: This component enables the real-time implementation of actions recommended by higher layers. It applies updates and feedback such as traffic rerouting, firewall rule adjustments, or QoS tuning without human intervention. Feedback mechanisms are integrated to monitor the impact of these changes, collecting metrics like latency and error rates, and reporting them back to the monitoring and analytics layers.

F. Knowledge Base (Database/Repository)

This layer serves as the central repository of information that supports all other components of the self-running network. It includes three key components: the *Configuration Database*, *Historical Data*, and *Contextual Information*.

1) Configuration Database: This component tracks the current state of physical and virtual network elements and how they are connected. It supports consistent and reliable operation by providing real-time access to configuration details. Version control allows changes to be reviewed, rolled back, or audited.

2) Historical Data: This component stores past records of traffic patterns, resource usage, and performance indicators. It helps the analytics layer detect trends, predict future capacity needs, and identify issues that develop over time.

3) Contextual Information: This component includes metadata that describes the business, user, or application context, such as service-level agreements, usage policies, or location-specific constraints. By using this information, the network can make decisions that reflect business goals and regulatory requirements. It enables the system to prioritize actions based on user needs and organizational rules.

G. Feedback and Continuous Improvement

This layer enables a self-running network to evolve based on new data, changing conditions, and continuous insights, rather than remaining static. Furthermore, it bridges the gap between what the network is configured to do and what it should do to achieve optimal performance, while remaining efficient and reliable over time. This layer includes *Feedback Loops*, *Machine Learning Pipelines*, and *Adaptive Policies*.

1) Feedback Loops: This component evaluates the outcomes of automated actions, comparing actual performance against expected results, such as latency levels following a route change. When discrepancies or suboptimal trends are identified, this information is fed back into the analytics or decision-making layers, enabling the refinement of strategies.

2) Machine Learning Pipelines: This component incorporates periodic model retraining to update AI/ML models with newly gathered data, ensuring they adapt to changes in traffic patterns, device performance, and user demands. To manage high volumes of network metrics, logs, and historical data efficiently, the system leverages scalable infrastructure, often utilizing distributed or cloud-based ML pipelines.

3) Adaptive Policies: This component enables iterative refinement, allowing administrators or automated processes to adjust policies, such as security rules or QoS priorities, in response to evolving business needs or regulatory conditions. By leveraging real-time analytics and historical insights, it drives performance-based adjustments to optimize cost savings, throughput, and compliance.

III. Functionalities of Self-Running Networks

Based on the conceptual model outlined in Section II, we now analyze the behaviors that distinguish self-running networks from traditional automated systems. These behaviors are defined by four interrelated *self-** functionalities: *self-configuration*, *self-optimization*, *self-healing*, and *self-protection*. These functionalities manifest across the network lifecycle, enabling intelligent and autonomous operation. To provide a high-level view of these, Figure 4 illustrates the four self-* functionalities alongside their primary objectives in self-running networks. Table IV summarizes their mechanisms, operational challenges, and domain-specific applications. Moreover, in Subsection III-E, we present the network autonomy maturity model, which helps evaluate the levels of autonomy and intelligence in network environments.

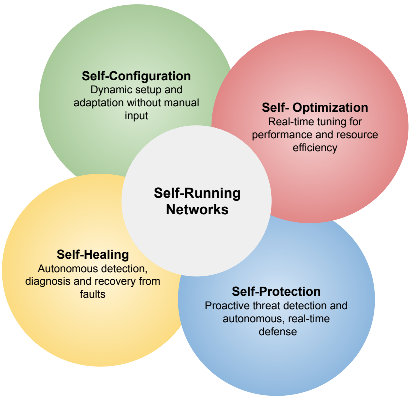
Self-Configuration
Dynamic setup and adaptation without manual input
Self- Optimization
Real-time tuning for performance and resource efficiency
Self-Running Networks
Self-Healing
Autonomous detection, diagnosis and recovery from faults
Self-Protection
Proactive threat detection and autonomous, real-time defense

Fig. 4: Self-* functionalities in self-running networks and their primary objectives

A. Self-Configuration

Self-configuration enables networks to set up and adapt their parameters and resources in response to real-time conditions, operational goals, and environmental changes [64], [65]. This functionality addresses the challenges of manual configuration in large-scale, heterogeneous, and evolving networks.

1) Key Principles and Mechanisms: This functionality is supported by three primary mechanisms: Intent-Based Configuration [66], Zero-Touch Provisioning (ZTP) [67], [68], and Policy-Driven Adaptation [69], [70]. Intent-based configuration enables networks to derive low-level configurations from high-level intents. By abstracting away manual complexity, intent-based configuration enables operators to define objectives in declarative terms, such as ”maximize throughput while maintaining low latency for critical applications.” This intent is translated into optimized routing policies, traffic engineering adjustments, and QoS configurations. The implementation of intent-based configuration follows a structured process. First, AI-driven intent engines parse and infer user-defined goals, converting them into actionable network requirements [71]. Next, constraint-based optimization confirms that the generated configurations adhere to network constraints, such as bandwidth, latency, and resource availability [72]. Finally, continuous validation mechanisms adjust configurations in response to deviations caused by workload fluctuations, congestion, or security threats [73].

ZTP enhances self-configuration by automating the deployment and configuration of network devices [74]. With ZTP, newly deployed devices, such as routers, switches, and virtual appliances, register upon connection, retrieve the necessary firmware and configurations from a centralized controller, and undergo authentication checks to verify compliance with security policies [76]. By eliminating the need for manual setup, ZTP accelerates large-scale network rollouts, enhances scalability, and reduces the risk of misconfigurations that could

TABLE IV: Overview of the Four Self-* Functionalities in Self-Running Networks: Key Mechanisms, Operational Challenges, and Domain-Specific Applications

Functionality	Key Mechanisms	Operational Challenges	Domain-Specific Applications
Self-Configuration	• Intent-Based Configuration [66], [71]–[73] • ZTP [67], [68], [74]–[76] • Policy-Driven Adaptation [69], [70], [77]–[80]	• Scalability [81], [82] • Reliability [83] • Human oversight vs. autonomy [84]	• Network slicing in 5G [64], [85], [86] • Dynamic resource allocation and intelligent load balancing in data center networks [87], [88] • Network parameters adjustments in IoT [89]–[91]
Self-Optimization	• Intent-Based Optimization [92]–[94] • Policy-Driven Optimization [95] • Autonomic Resource Orchestration [96]–[99]	• Stability vs. responsiveness [100] • Computational overhead and real-time decision-making constraints [101] • Cross-domain interoperability [102]	• Power control and handover optimization in 5G [103], [104] • Auto-scaling and load balancing in cloud computing [105] • Routing and resource tuning in IoT/edge computing [106] • Traffic prediction in transport [107] • Energy savings in green networking [108]
Self-Healing	• Anomaly detection and diagnosis [109]–[116] • Automated recovery and remediation [117], [118] • Proactive failure prevention [119]–[124] • Redundancy and failover management [125]–[133] • Root cause analysis and adaptive optimization [134]–[137]	• Accurate fault detection [138] • False positives and negatives [139], [140] • Root cause localization [141] • Performance overhead [142] • Excessive or unstable recovery actions [143] • Cascading failures [144]	• Automated network slicing recovery and base station failover in 5G [145], [146] • Server and switch failover in data centers [147], [148] • Device recovery and rerouting in IoT/edge [117], [149] • Link failure detection in SD-WAN [150] • Fault mitigation in optical networks [151], [152]
Self-Protection	• Proactive threat detection [153], [154] • Dynamic Policy Enforcement/Autonomous Decision-Making [155]–[157] • Intelligent Isolation and Containment [158]–[160] • Behavioral Analysis/UEBA [161], [162] • Deception Techniques [163]	• Accuracy and Reliability of AI Models [164] • Aggressiveness of Automated Actions [165] • Computational Overhead and Scalability [166] • Resource Constraints in IoT and Edge [166] • Legal and Regulatory Compliance [157]	• Threat detection and isolation in 5G [154] • Resilience in MEC via fault isolation and workload rerouting [101] • Secure infrastructure defense in autonomous transport systems [167] • Cyber-physical protection in industrial control and critical infrastructure [157]

lead to performance degradation or security vulnerabilities [75]. Policy-driven adaptation complements intent-based configuration and ZTP [77]. This mechanism guarantees that these configurations and deployments remain aligned with organizational requirements, security mandates, and performance objectives as network conditions evolve [80]. Policies are defined by either administrators or inferred autonomously by the system using AI/ML models [78], [79], specifying high-level outcomes rather than granular implementation details.

2) Operational Challenges and Domain-Specific Applications: Scalability is one of the primary concerns, as the intricacy of managing configurations grows with network size, particularly in heterogeneous environments that combine legacy systems with next-generation networks [81], [82]. Reliability is also vital, as incorrect or suboptimal configurations can lead to significant service disruptions, degraded performance, or heightened security vulnerabilities [83]. Another challenge is balancing autonomy with human oversight. While the goal is to minimize manual intervention, edge cases and unforeseen scenarios may necessitate mechanisms for human-in-the-loop supervision to prevent cascading failures [84]. Despite these challenges, self-configuration has demonstrated potential through successful applications across different domains. In 5G, it enables the autonomous management of network slicing and guarantees that each slice meets the requirements of specific applications, such as ultra-low latency for autonomous vehicles or high reliability for industrial systems [64], [85], [86]. In data center networks, self-configuration supports dynamic resource allocation and intelligent load balancing, optimizing performance, energy consumption, and operational costs [87], [88]. Similarly, in IoT ecosystems, self-configuration plays a critical role in edge computing by dynamically adjusting network parameters to accommodate device heterogeneity, mobility, and varying bandwidth requirements [89]–[91]. As research continues to address limitations, the role of self-configuration is expected to expand, laying the groundwork for self-running networks.

B. Self-Optimization

Self-optimization enables networks to tune performance by adjusting operational parameters and resource allocations in response to changing workloads, traffic patterns, environmental conditions, and user demands. Through this functionality, the network makes informed decisions to balance two competing concerns: resource overprovisioning, such as allocating excess CPU cores or memory, or unnecessarily scaling cloud instances during stable workloads, and service request violations, such as underprovisioning bandwidth, insufficient memory, or inadequate CPU allocation. For example, a SDN controller can leverage real-time traffic analytics and ML to optimize bandwidth allocation across multiple network paths [107]. During high demand, the controller redistributes traffic to underutilized links, reducing congestion without overprovisioning. During low demand, it merges flows to save energy by disabling idle links, thereby improving overall efficiency. This functionality keeps optimal QoS while preventing both excessive resource allocation (e.g., unused bandwidth) and service-level violations (e.g., packet loss and increased latency due to under-provisioning).

1) Key Principles and Mechanisms: This functionality is supported by three primary mechanisms: intent-based optimization, policy-driven optimization, and autonomic resource orchestration. Similar to intent-based configuration, intent-based optimization allows operators to define high-level performance goals, which are interpreted by AI-driven engines and translated into real-time network adjustments such as dynamic load balancing, resource scaling, and traffic redistribution to maintain optimal network performance [92]. The effectiveness of intent-based optimization is further enhanced by closed-loop control systems, which enable the network to react to traffic surges, hardware failures, or changing application demands [93], [94]. In addition to intent-based optimization, policy-driven optimization defines optimization policies that guide network adaptations within predefined operational constraints. Policies can be either static or adaptive. Static policies remain fixed, such as predefined bandwidth caps, processing power limits, or quality of service requirements. In contrast, adaptive policies leverage AI and ML to adjust rules based on historical data, real-time network conditions, and predictive analytics [95].

Also, autonomic resource orchestration checks that resources such as processing power, storage, and network bandwidth are allocated across multiple network layers to optimize overall efficiency. This mechanism includes resource-aware scheduling [96], network slicing [97], and traffic engineering [98] to help distribute resources based on latency requirements, energy constraints, and real-time demand fluctuations. For example, network slicing allows networks to create dedicated, logically isolated segments optimized for specific services, such as low-latency communication for autonomous systems or high-throughput connectivity for data-intensive applications.

2) Operational Challenges and Domain-Specific Applications: One of the primary challenges is balancing performance with stability. While self-optimization mechanisms adjust network parameters to enhance efficiency, excessive optimization can lead to system oscillations, instability, and unintended performance degradation [100]. Another challenge is computational overhead and real-time decision-making constraints. Self-optimization relies on AI-driven analytics, predictive modeling, and real-time monitoring, all of which require substantial computational resources. Optimization mechanisms must make decisions with minimal latency in time-sensitive, resource-constrained environments such as 5G and edge computing, ensuring that network adaptations occur fast enough to maintain service quality [101]. Also, interoperability across network architectures is another concern. Self-optimization must function across SDN, network function virtualization Network Functions Virtualization (NFV), and traditional hardware-based infrastructure, ensuring compatibility without sacrificing efficiency [102]. Despite these challenges, self-optimization has demonstrated real-world impact across multiple domains. Self-optimization in 5G networks can adjust transmission power levels, reallocate spectrum resources, and optimize handover mechanisms [103], [104]. In cloud computing, self-optimization supports intelligent workload distribution, dynamic auto-scaling, and power-efficient resource allocation [105]. By adjusting transmission power, routing paths, and computational workloads, self-optimization in IoT networks improves device longevity and responsiveness [106]. Self-optimization facilitates real-time traffic prediction, congestion-aware rerouting, and adaptive vehicle-to-vehicle communication in autonomous vehicles and intelligent transportation systems. In energy-efficient networking and green computing, self-optimization is reducing carbon by adjusting resource usage, enabling sleep modes for underutilized network devices, and optimizing power distribution [108].

C. Self-Healing

This functionality allows networks to detect, diagnose, and recover from faults or performance issues by analyzing the issue and triggering predefined or adaptive remediation strategies [168], [169]. For instance, a failed network link prompts automatic traffic rerouting to maintain service continuity [170]. Similarly, if a device malfunctions, the network isolates the faulty component and redistributes traffic across available nodes to sustain operations.

1) Key Principles and Mechanisms: This functionality is supported by several key mechanisms: anomaly detection and diagnosis, automated recovery and remediation, proactive failure prevention, redundancy and failover management, and root cause analysis and adaptive optimization. Anomaly detection and diagnosis is achieved through real-time monitoring systems that collect and analyze telemetry, such as packet loss, latency, and throughput. Statistical methods, such as Exponential Moving Averages and Cumulative Sum Control Charts [109]–[111], help detect deviations from baseline performance metrics by monitoring historical trends for significant shifts. Behavior-based techniques, such as Maximum Entropy Estimation [113], analyze deviations in network traffic to identify potential anomalies. Other approaches, such as the Static Baseline Algorithm [114], enhance anomaly detection by setting predefined thresholds and flagging unexpected deviations. ML-based approaches, including Random Forests, Support Vector Machines, clustering, and autoencoders, identify patterns indicative of faults [115], [116]. Moreover, DL models are effective for time-series analysis, detecting complex, multi-dimensional anomalies [112].

Once a fault is detected, automated recovery mechanisms restore normal operations via orchestration platforms and policy-based automation [117], [118]. For example, SDN reroutes traffic through alternate paths using OpenFlow, and orchestration tools like Kubernetes or OpenStack restart failed Virtualized Network Functions or containers. Control-theory feedback loops and Reinforcement Learning (RL) also refine these actions in real time [171], [172]. Low-level automation, such as Python scripts and Ansible playbooks, can handle fine-grained tasks such as log analysis, connection resets, and firmware updates. To further enhance network resilience, proactive failure prevention anticipates and mitigates potential issues before they occur [119].

This is achieved through predictive analytics and adaptive resource management. Predictive maintenance models, such as Random Forests and Gradient Boosting Machines, analyze historical data to predict failures in critical network components, such as routers or switches, based on performance metrics like CPU usage, memory utilization, and temperature. For example, by monitoring these metrics, the system can identify patterns indicative of an impending router failure, enabling the network to reroute traffic, balance loads, or schedule maintenance before a disruption occurs [120]. Moreover, time-series forecasting models, such as Prophet, are employed to forecast network load and resource utilization. These forecasts allow for preemptive scaling or reallocation of resources, ensuring optimal performance even under varying demand [121], [122]. Digital twins (i.e., virtual replicas of the network) can also be utilized because they allow the network to test and implement solutions in a virtual environment before deployment [123], [124].

While proactive measures and automated recovery mechanisms improve network resilience, ensuring uninterrupted service also requires robust redundancy and failover mechanisms [125]–[127]. High availability architectures deploy redundant hardware or software components to guarantee seamless failover, such as hot standby router protocol [128] for router redundancy or bidirectional forwarding detection [129] for rapid link failure detection in network environments. Load balancers like NGINX [130] and HAProxy [131] distribute traffic across multiple nodes, checking that failures in one component do not disrupt the entire network. Also, stateful failover mechanisms, such as Cisco stateful switchover [132] and Kubernetes StatefulSets [133], preserve session data during failover, enabling uninterrupted service even when individual components fail [173].

Additionally, the ability to diagnose and address the root cause of faults is critical for achieving true self-healing functionality in self-running networks. Advanced AI/ML-based techniques, such as graph-based algorithms and causal inference models, enable root cause analysis by analyzing system dependencies and identifying the underlying sources of failures [134], [135]. Once the root cause is identified, RL agents iteratively refine recovery strategies, adapting to dynamic network conditions and optimizing remediation actions based on real-time feedback [136]. Also, Explainable AI (XAI) techniques, such as Shapley additive explanations and model-agnostic explanations, provide transparency into AI-driven decision-making [137].

2) Operational Challenges and Domain-Specific Applications: Self-healing offers significant benefits, but it also faces several challenges that must be addressed to realize its full potential. First, accurate fault detection is a key part of effective self-healing, yet it remains a significant challenge [138]. Self-healing systems must precisely differentiate between transient issues (e.g., temporary congestion, packet loss) and persistent failures (e.g., hardware malfunctions, software bugs). However, legitimate network changes, such as updates, reconfigurations, or traffic rerouting, can potentially exhibit behaviors similar to faults, leading to false positives or false negatives. To address this, self-healing systems must incorporate context-aware detection mechanisms that consider the operational state of the network and leverage historical data to improve fault identification accuracy [139], [140]. Beyond fault detection, pinpointing the exact source of a problem, especially in multi-layered, highly distributed networks, is important. Modern networks often consist of interdependent layers, including programmable SDN controllers, Virtual Network Functions (VNF)s, and physical infrastructure, each introducing potential points of failure. For example, performance degradation in a VNF could stem from an underlying hardware issue, a misconfiguration in the SDN controller, or even a cascading effect from another VNF. This complexity is exacerbated in multi-vendor environments, where proprietary systems and a lack of standardized telemetry data hinder cross-layer diagnostics. Advanced root cause techniques and distributed tracing tools (e.g., Jaeger, OpenTelemetry) are essential for mapping dependencies and identifying the true source of faults [141].

However, these techniques often require significant computational resources and may struggle with real-time analysis in large-scale networks. In addition, mechanisms such as redundancy, failover, and traffic rerouting can consume additional bandwidth and computational resources, leading to increased latency or resource contention. Finally, preventing cascading failures is also a challenge, where an automated recovery action inadvertently triggers a chain reaction of additional failures [144]. For example, rerouting traffic to bypass a failed node may overload adjacent nodes, or poorly designed reconfiguration loops, where the network repeatedly attempts and fails to resolve a fault, can worsen disruptions.

Even with these challenges, self-healing has been applied in various network environments. In 5G networks, self-healing enables the autonomous recovery of network slicing and base stations, ensuring uninterrupted service for critical applications such as augmented reality [145], [146]. For example, if a base station fails, self-healing mechanisms can reroute traffic to neighboring stations or activate backup resources, maintaining seamless connectivity. In data center networks, self-healing supports rapid failover and load redistribution during network failures, recovers performance, and minimizes downtime. For instance, if a server or switch fails, the system can reroute traffic and redistribute workloads to healthy nodes [147], [148]. In IoT networks, self-healing facilitates edge computing by adjusting to device failures or connectivity issues [117], [149].

For example, if an edge device fails in a smart factory, self-healing mechanisms can reroute data processing tasks to other devices or the cloud. Self-healing is also applied in enterprise networks, particularly in SD-WAN. Here, self-healing mechanisms can detect and resolve issues such as link failures or congestion [150]. In cloud-native environments, self-healing is integral to container orchestration platforms like Kubernetes, where it automatically restarts failed containers, replaces unresponsive pods, and guarantees high availability of microservices [174]. Additionally, in telecommunications networks, self-healing is used to manage optical networks, where it can detect fiber cuts or equipment failures and reroute traffic to minimize service disruptions [151], [152].

D. Self-Protection

Unlike traditional security models that rely on static rules and manual intervention, self-protection functionality leverages intelligent automation to identify vulnerabilities, enforce security controls, and mitigate risks before they escalate [154]. Also, in contrast to self-healing, which focuses on post-incident recovery, self-protection is designed to prevent failures and security breaches through continuous monitoring, analysis, and proactive countermeasures.

1) Key Principles and Mechanisms: Self-protection functionality is supported by five key mechanisms: proactive threat detection, dynamic policy enforcement, intelligent isolation and containment, behavioral analysis, and deception-based defense. Proactive threat detection helps to assess risk, anticipate vulnerabilities, and adapt security postures such as Zero Trust Security (ZTS) [156] [157]. This requires the deployment of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) to monitor network activity and detect malicious behavior. Tools such as Zeek, Snort, and Cisco Secure IPS identify patterns indicative of intrusion attempts, malware propagation, or unauthorized access. In addition, modern IDS IPS solutions integrate ML-driven anomaly detection, enabling networks to recognize deviations from baseline traffic patterns proactively [153]. Behavioral analysis techniques such as User and Entity Behavior Analytics (UEBA) [161] also enhance this functionality by monitoring deviations in user access behaviors, device activity, and network communication flows [162]. Moreover, this functionality integrates automated decision-making frameworks such as SOAR and XDR to modify firewall rules, access control policies, and encryption protocols based on detected anomalies [155].

Intelligent isolation and containment utilize technologies such as quarantine mechanisms [158], micro-segmentation [159], and just-in-time [160], further enhancing system resilience by isolating network segments and restricting access based on real-time risk assessment. Additionally, deception techniques, including honey tokens and decoy systems, mislead adversaries, gathering intelligence while delaying and mitigating attacks [163]. Hence, self-protection functionality works with self-configuration and self-optimization to modify access controls, firewall settings, encryption protocols, and traffic management policies in response to detected anomalies.

2) Operational Challenges and Domain-Specific Applications: Despite the advantages of self-protection in autonomous networks, several operational challenges hinder its deployment and effectiveness. One of the primary challenges is the accuracy and reliability of AI-driven decision-making in security and performance optimization. While ML models enhance anomaly detection and automated response mechanisms, they are susceptible to false positives and false negatives, which can lead to unnecessary disruptions or overlooked threats [164]. An overly aggressive self-protection mechanism may isolate legitimate network activity, degrade system performance, or disrupt essential services. Conversely, insufficient sensitivity in anomaly detection may allow sophisticated attacks to bypass security measures. Confirming that AI models are continuously trained, validated, and refined with high-quality, unbiased datasets is critical to improving their precision and reliability.

In large-scale networks, such as telecommunications and cloud infrastructures, the demand for high-speed processing and decision-making creates computational overhead. Implementing lightweight AI models and edge-based security intelligence can help offload computational workloads to distributed processing nodes. However, balancing real-time self-protection with network efficiency is a fundamental challenge, particularly in resource-constrained environments such as IoT and edge computing ecosystems [166]. Moreover, self-protection functionality needs AI-driven decision-making, which can raise problems in security regarding transparency and compliance with legal frameworks. This functionality should check that actions such as isolating users, blocking access, or migrating workload comply with legal frameworks.

However, self-protection can have profound implications across different types of networks to make networks reliable and resilient. For instance, deploying AI-driven self-adaptive firewalls and autonomous security orchestration allows 5G networks to detect and isolate threats before they impact service availability. Multi-access edge computing environments benefit from self-protection by automatically detecting hardware failures, isolating faulty nodes, and rerouting workloads. Also, beyond traditional environments, self-protection is becoming integral to autonomous transportation networks, confirming the safe and reliable operation of connected vehicles, smart traffic management systems, and automated fleet coordination [167]. In industrial control systems and critical infrastructure, self-protecting networks are essential for defending power grids, water treatment facilities, and manufacturing plants from cyber-physical attacks and operational disruptions.

While each self-* functionality addresses specific needs, they often rely on shared mechanisms and support each other to achieve holistic network autonomy. For instance, anomaly detection serves as a mechanism for self-healing and self-protection; the former uses it to detect faults and performance degradations, while the latter uses it to identify security threats or behavioral deviations. Similarly, intent-based frameworks are applied in self-configuration, self-optimization, and self-protection; the former translates high-level goals into device configurations, while the latter two use intents to drive performance tuning and adaptive security policies. Policy engines also play a shared role, whether by enforcing configuration rules, optimization boundaries, or access control decisions.

Even autonomic orchestration overlaps: self-optimization and self-healing both rely on AI-driven resource orchestration and dynamic adaptation. However, the former focuses on performance efficiency and the latter on service recovery. These overlaps highlight the need for integrated implementation across functionalities rather than isolated implementation. The four mentioned functionalities of self-running networks are the keys to a new era of network management. By removing human intervention and leveraging real-time insights, self-running networks improve overall network performance and enhance user experiences. However, to understand the network intelligence and autonomy, we need a structured model that indicates the network's maturity and evaluates how close it is to becoming a self-running network. This model will be presented in subsection III-E.

E. Network Autonomy Maturity Model

The International Telecommunication Union [175] defines a standardized framework for evaluating intelligence in future networks, particularly focusing on IMT-2020. In the context of self-running networks, knowing about the intelligence level of network management plays a central role in supporting the evolution of these next-generation networks. Their framework outlines six levels of network intelligence, ranging from fully manual, human-driven operations (Level 1) to fully autonomous systems (Level 6). Based on this framework, we propose a network autonomy maturity model that outlines how intelligence evolves across different dimensions of networks. As illustrated in Table V, the model evaluates each level based on five key features and maps them to corresponding levels of intelligence. These features represent essential capabilities that develop as networks become more intelligent.

There are five features to evaluate network maturity. **Network contextual analysis** leverages extensive datasets to gain insights into network events and identify their underlying causes. It provides a deep understanding of the network environment without requiring real-time processing. **Offline predictive analytics** employ AI/ML techniques to forecast potential events or changes in a non-real-time setting, enabling proactive planning and management. **Operational decision-making** refers to the autonomous formulation and implementation of optimization policies. Here, decisions are based on non-real-time data, ensuring strategies are data-driven and carefully validated. **Real-time inference and action** focus on immediate predictions and rapid responses, allowing the network to adapt dynamically and maintain optimal performance. **Anomaly detection and resolution** involve the proactive identification of unexpected events and the autonomous adjustment of network policies to mitigate their impact.

Each level in the model reflects a specific degree of autonomy across these five features, progressively reducing the need for human oversight. **Level 1**, fully human-centric networks, represent systems in which all cognitive and operational functions, ranging from context analysis to decision-making and exception handling, are entirely performed by human operators. These networks lack any form of machine intelligence or real-time predictive capability, relying solely on manual intervention and expertise. **Level 2**, assisted automation networks introduce limited AI assistance specifically for contextual analysis under constant human supervision. Human operators control decision-making, prediction tasks, and exception handling. **Level 3**, advanced automation networks, extend AI autonomy into context analysis, requiring only occasional human oversight, while integrating AI-assisted predictive analytics under human supervision. Decision-making, real-time predictions, and exception handling remain fully controlled by human experts.

Level 4, partially autonomous networks significantly expand AI autonomy, independently performing most context analyses with minimal oversight and moderate autonomy in real-time and predictive analytics. Humans retain primary responsibility for decisions and fully control exception handling, reflecting a balanced human-AI collaborative environment. **Level 5**, advanced autonomous networks, represents near-complete AI autonomy across all analytical and operational tasks, including context analysis, predictive analytics, and operational decision-making, requiring minimal human intervention. Human oversight is limited primarily to exception handling, specifically managing unusual or critical scenarios exceeding AI capabilities. **Level 6**, fully autonomous networks operate entirely without human intervention, independently managing all analytical tasks, predictive functions, operational decisions, and exception handling. These networks run themselves, fix their problems, and adapt to changes without needing humans to step in.

Hence, the Network Autonomy Maturity Model provides a framework for evaluating how intelligence emerges and progresses within self-running networks. The five core features closely align with the four self-* functionalities introduced earlier in this section. For example, self-configuration and self-optimization benefit from enhanced contextual understanding and decision-making, while self-healing and self-protection rely on real-time responses and anomaly resolution. This model serves as a foundational reference to reason about autonomy levels in practical terms. It allows researchers to position a given network environment within a defined maturity spectrum and identify which functionalities are underdeveloped or missing. In this way, the model supports structured thinking and provides a shared vocabulary for discussing, comparing, and designing self-running network behaviors.

IV. Research and Development Towards Self-Running Networks

In this section, we examine the research efforts that have contributed to the emergence of self-running networks. To provide a structured analysis, we categorize these works into two key domains: foundational paradigms and practical applications. The first domain covers studies that propose the frameworks and conceptual models forming the basis of self-running networks. The second domain focuses on practical

Intelligence Levels Key Features	Level 1 Fully Human-Centric Networks	Level 2 Assisted-Automation Networks	Level 3 Advanced-Automation Networks	Level 4 Partially Autonomous Networks	Level 5 Advanced Autonomous Networks	Level 6 Fully Autonomous Networks
Network Contextual Analysis	Manual	AI-assisted with human oversight	Moderate Autonomy	High Autonomy	High Autonomy	Full Autonomy
Offline Predictive Analytics	Manual	Manual	AI-assisted with human oversight	Moderate Autonomy	High Autonomy	Full Autonomy
Operational Decision-Making	Manual	Manual	Manual	AI-assisted with human oversight	High Autonomy	Full Autonomy
Real-Time Inference and Action	None	None	None	Moderate Autonomy	High Autonomy	Full Autonomy
Anomaly Detection and Resolution	Manual	Manual	Manual	AI-assisted with human oversight	Moderate Autonomy	Full Autonomy

TABLE V: Network Autonomy Maturity Model: This figure presents a six-level model describing the evolution of network intelligence from fully human-centric to fully autonomous networks. Each level is assessed across key features, and the color gradient represents increasing autonomy: teal blue for AI-assisted with human oversight, light teal for moderate autonomy, medium teal for high autonomy, and darker teal for full autonomy.

implementations and use cases that demonstrate how these networks have been applied in real-world scenarios. This categorization offers a clearer understanding of both the underlying principles and their tangible outcomes.

A. Foundational Paradigms

Research in foundational paradigms explores the core principles and architectural approaches that enable self-running network behavior. Table VI shows an overview of these studies.

1) Policy-Based Network Management: The framework and architectural components of this paradigm were defined by the Internet Engineering Task Force in 2000 [192] [193]. It is an approach in which predefined rules, known as policies, are employed to configure network components and services. This approach is practical, especially for managing heterogeneous networks that require continuous availability and need to be reconfigured dynamically without downtime. Service providers can develop and implement these policies, and they are responsible for identifying and setting up appropriate policy configurations [193]. In general, these policies can be categorized into two groups: authorization policies, which determine what activities a user has permission to engage in within the system, and obligation policies, which define actions that the system must or must not take in response to specific events [194].

Based on this paradigm, Verma [176] presents a framework for easing the complexities of managing modern IP networks. The framework centralizes the configuration and provisioning of network devices, using a policy management tool that allows administrators to define policies at a business level, which are then translated into technology-specific configurations. The framework reduces the manual effort required for network management by implementing centralization and business-level abstractions. The paper also introduces algorithms for policy validation to ensure consistency, feasibility, and conflict-free operation. However, a significant limitation is the potential latency and performance bottlenecks that may arise from centralizing policy management in dynamic and large-scale networks.

Rana et al. [177] utilize Policy-Based Network Management (PBNM) in their study of Home Area Network (HAN). HAN often involves wired and wireless devices, home appliances, gaming consoles, and cameras. The research aims to simplify administrative management tasks by employing PBNM, reducing operational costs, and minimizing errors. In their HAN testbed, they implement a traffic management platform that enables them to apply policies for traffic prioritization. This approach decreases packet loss to 30% and enhances the performance of Voice over Internet Protocol (VoIP) services.

The researchers implemented a testbed with several components, including a Policy Builder, Policy Engine, Traffic Conditioner, and Traffic Controller, to experiment with different traffic management scenarios. The results demonstrated that policy-based management could effectively manage network resources, prioritize essential traffic, and improve overall service quality in HAN. However, real-world networks experience varying conditions and user behaviors, which might challenge the static policies used in the study. Solomon et al. [178] propose a policy creation model for managing network bandwidth at the campus network. By employing a structured approach to policy creation and implementation, the research optimizes bandwidth allocation, prioritizing academic and research activities. Key methodologies in this work include network analysis, traffic modeling, and the application of policy-based network management using tools like the Fortigate firewall for simulation. However, the model's static nature limits its responsiveness to real-time network changes, which is a critical limitation in dynamic network environments. Alquhayz et al. [179] present an approach to enhancing security in 5G networks by integrating a policy-based management system with the Y-Comm framework. The system aims to prevent end-user devices from being exploited as attack tools by implementing an intelligent agent to detect and report malicious activities. Their results from the simulation show the system's effectiveness in reducing disconnection rates in scenarios involving IP spoofing and man-in-the-middle attacks. While the system includes mechanisms for detecting malicious behavior, it does not discuss the potential privacy implications

TABLE VI: Existing Research on Foundational Paradigms Toward Self-Running Networks

Category	Paper/Year	Objective	Contribution	Limitation
Policy-Based Network Management	Verma 2002 [176]	Simplifying the management of complex network infrastructures by using PBNM	• Developed a framework that simplifies network administration by centralizing configuration and using business-level abstractions.	Latency and bottlenecks in centralized policy management
	Rana et al. 2009 [177]	Using PBNM for managing HAN to improve QoS and security management while minimizing user complexity	• Set up a practical environment to experiment with and demonstrate the effectiveness of PBNM • Showed significant enhancements in VoIP quality and reduction in packet loss through policy enforcement	Limited adaptability of static policies to dynamic conditions
	Solomon et al. 2017 [178]	Proposing a policy creation model for policy making in organizations to improve bandwidth management and network efficiency.	• Development and implementation of a Policy Creation Model tailored for the specific university network. • Demonstrated improved bandwidth management and network efficiency post-policy implementation.	Lack of dynamic reconfiguration for variable traffic
	Alquhayz et al. 2019 [179]	Developing a policy-based approach to security management systems to prevent end-user devices from being used as attack tools	• Developed a policy-based security management system integrated with Y-Comm architecture to enhance 5G security. • Introduced intelligent agents to detect malicious behavior in end-user devices.	Potential privacy concerns from user activity monitoring
Autonomic Networks Management	Arzo et al. 2021 [180]	Proposing an architecture for ANM using a multi-agent system	• Proposed and evaluated an ANM architecture using a multi-agent system, assessing functionality, reliability, latency, and resource consumption.	Lack of integration details with existing systems
	Tsagkaris et al. 2015 [181]	Enhancing network management by integrating ANM and SDN	• Developed a customizable ANM framework integrating SDN/OpenFlow capabilities.	Lack of real-world validation for scalability and performance
	Jiang et al. 2017 [182]	Designing and implementing an autonomic network management framework for 5G mobile networks	• Implemented an autonomic management framework under the EU H2020 SELFNET project for software-defined and virtualized 5G networks. • Designed network intelligence using ML techniques for self-healing, self-protection, and self-optimization.	Limited testing beyond congestion scenarios
	Stamou et al 2019 [183]	Enhancing device-to-device communication and resource management in cognitive radio networks	• Proposed a framework integrating ANM with SDR, SDN, and NFV. • Tested the framework on real-world testbeds to demonstrate feasibility and effectiveness.	Complex dependencies and new security vulnerabilities
Intent-Based Networks Management	Abbas et al. 2020 [184]	Implementing an IBN slicing system for 5G networks to manage core and RAN resources	• Implemented a GAN-based deep learning model for predictive resource management and enhanced slice assurance. • Automated the life-cycle management of network services, significantly reducing manual effort.	Security risks from open-source reliance and automation
	Abbas et al 2021 [185]	Developing an intent-based slice life-cycle management system for 5G networks to automate creation, configuration, and management using high-level intents.	• Proposed an IBN framework for managing the life-cycle of network slices in 5G networks. • Implemented a prototype system and validated it with performance tests to demonstrate effectiveness and efficiency.	Limited support for user mobility
	Collet et al. 2022 [186]	Developing a forecasting model that autonomously learns the relationship between predictions and network management objectives to optimize complex IBN goals	• Implemented an architecture enabling the model to autonomously learn suitable loss functions for various objectives, reducing manual intervention.	High computational demands from the dual DNN architecture
	Orlandi et al. 2024 [187]	Simplifying service ordering and configuration through user-friendly interfaces and NLP	• Developed a framework for engaging non-expert users, including an NLP-enhanced chatbot for intent translation and vertical automation from service order to network deployment.	Dependence on product catalog accuracy and completeness
Zero-Touch Networks Management	Rezazadeh et al. 2020 [188]	Developing an AI-driven, zero-touch network slicing solution using twin delayed deep deterministic policy gradient	• Adopted and fine-tuned the TD3 method to improve convergence speed and learning stability in continuous DRL tasks. • Developed a comprehensive 5G network slicing environment using OpenAI Gym to enable standardized testing and algorithm comparison.	Challenges in real-world implementation beyond simulation
	Sousa et al. 2021 [189]	Enhancing end-to-end service monitoring	• Introduced the MMG component to generate Service Monitoring Models using service deployment models and standard information models.	Need for scalability and performance evaluation in large-scale networks
	Angui et al. 2022 [190]	Automating the deployment of Cloud-RAN in 6G networks, focusing on latency and resource management	• Introduced and validated a ZTC model for Cloud-RAN that automates resource discovery, deployment, and configuration of network elements. • Developed a protocol using Elliptic Curve Cryptography for session key establishment and Proof-of-Authority for block verification.	Potential challenges in real-world deployments
	Kumar et al. 2022 [191]	Enhancing security in IoT-enabled Zero Touch Networks	• Created a novel intrusion detection system combining Variational AutoEncoder and attention-based Gated Recurrent Units for automatic feature extraction and intrusion detection. • Developed a protocol using Elliptic Curve Cryptography for session key establishment and Proof-of-Authority for block verification.	Limited scalability beyond small-scale test setups

of monitoring and analyzing user activities.

2) Autonomic Networks Management: Autonomic Networks Management (ANM) addresses the ability of networks to be aware of themselves and their environment [4]. ANM can perceive current network conditions, plan, decide, act on those conditions, learn from the consequences of these actions, and follow their goals. This feedback loop implements a learning model in which past interactions with the environment guide current and future interactions and result in intelligence enhancements [180]. The goal is to create self-managed networks to overcome the rapidly growing complexity of networks. ANM shares motivation and has confluent goals with other emerging technologies, such as SDN and NFV, as all three concepts seek to increase the flexibility, reliability, and efficiency of operations and optimize network management and control [183]. The autonomic network contains multiple autonomic elements capable of regulating their internal operations and interactions with other autonomic elements.

Each autonomic element consists of one or more managed elements and an autonomic manager. The role of the autonomic manager is to oversee and control each managed element, such as a CPU, printer, database, or directory service. This approach minimizes the requirement for human intervention in managing these elements, leading to enhanced efficiency and reduced manual oversight [4].

Several papers focus on the architecture of autonomic networks [180], [195]–[200]. For example, the study by Arzo et al. [180] introduces an architecture designed to manage complex networks by using multiple interacting agents. Each agent performs specific network functions autonomously, such as network slicing, path computation, and QoS monitoring, and the architecture can be replaced with traditional monolithic network management systems. Multiple papers focus on integrating autonomic network management with emerging network technologies to address the increasing complexity of network systems (including [181], [182], [201], among others). For instance, Tsagkaris et al. [181] work on enhancing network management by integrating ANM and SDN to develop a customizable management framework. The authors develop a prototype and conduct various experiments to demonstrate the potential gains in efficiency and manageability. Key use cases in this work included policy-based traffic engineering, life-cycle management of autonomic control loops, and coordination of multiple control loops, showing reduced power consumption and improved traffic management.

Jiang et al. [182] focus on ANM in 5G systems. It leverages technologies such as SDN and NFV to propose a management framework. The presented framework, developed under the EU H2020 SELFNET project, emphasizes self-healing, self-protection, and self-optimization functionalities for 5G networks, similar to the themes of integrating advanced network management techniques with modern network architectures found in the other papers. Stamou et al. [183] develop a new framework that integrates ANM with SDN, Software-Defined Radio (SDR), and NFV to enhance device-to-device communication and resource management in cognitive radio networks. Through experimental validation on two real-world testbeds, the framework demonstrated its capability to adapt to different network conditions, efficiently allocate resources, and minimize collisions. While the framework shows promise in improving spectrum utilization and QoS, the integration of these advanced technologies also introduces potential security vulnerabilities that need to be addressed. Despite the aim of autonomic networks to achieve self-management, they are unable to eliminate the necessity for operator or external system intervention because autonomic networks require an operator or outside system to define guidance and information regarding their purposes and service instances [59].

3) Intent-Based Networks: The intent is the high-level expression that can be translated and deployed in networks, and an IBN is a network that is operated and managed based on the intent. The intent is to evolve the term policy. Hence, IBN is the evolution after PBNM, and as opposed to the policy, the intent defines a high-level operational goal without specifying how it should be achieved. Additionally, intents are independent of any hardware to ensure they can be defined across different technologies [202]. Since Intent-Based Network Management has the ability to manage networks holistically at a higher level of abstraction, operators can concentrate more on their desired outcomes without being concerned about the low-level device configuration required to achieve or implement them [59]. Multiple studies demonstrate how IBN principles are implemented in different contexts and for various purposes within 5G networks [184]–[187], [203]–[211] among others.

For example, Abbas et al. [184] leverage an IBN-based approach for performing end-to-end network slicing to design, control, manage, and monitor network slice resources. In this work, network operators can provide the network slice using the IBN tool to take autonomous actions for both domains. Following this work, the authors focus on an IBN approach for managing the life-cycle of network slices in 5G networks. The proposed system automates the creation, configuration, and management of network slices using high-level intent expressions [185].

Collet et al. [186] introduce LossLeaP, a DL-based model for IBN that autonomously learns and aligns its predictions with complex network management objectives using a predictor and a loss-learning block. While LossLeaP outperforms existing models in forecasting tasks, its practical deployment faces challenges due to high computational overhead, dependency on comprehensive training data, and integration complexity. Meanwhile, multiple works propose an architecture based on IBN [187], [212], [213]. For instance, in a recent study, Orlandi et al. [187] implement an architecture of IBN, particularly to simplify user interactions and service configurations through Natural Language Processing (NLP) and user-friendly interfaces. Hence, IBN management is more about aligning network operations with business intents and ensuring that these intents are met. However, there are still many challenges in IBN, from declaring intents and transforming users' business or operational intents to ensuring the intent works as users' intentions in ever-changing services and applications. Also, IBN is not completely decentralized since some functions need to be centralized. The need for a global view due to the volume of data cannot be possible in IBN.

4) Zero-Touch Network: It was established in December 2017 with the primary goal of fully automating networks and moving away from inflexible management systems towards more adaptable services [214], [215]. In other words, this management framework supports and executes operational tasks such as planning and design, delivery, deployment, provisioning, monitoring, and optimization. The architecture of ZSM contains multiple Management Domains (MDs), which have a responsibility to orchestrate, control, and assure resources and services within its scope [30]. This architecture should be modular, extensible, scalable, and resilient to failure. Multiple research papers were published regarding ZSM.

Rezazadeh et al. [188] present a novel zero-touch network slicing solution leveraging the TD3 algorithm for continuous multi-objective resource allocation in 5G networks. By developing an OpenAI Gym environment for standardized testing, the study shows significant improvements in network performance metrics, which highlights the potential of advanced DRL methods in achieving efficient and automated network management. Sousa et al. [189] present a methodology for end-to-end service monitoring in zero-touch networks by introducing the monitoring model generator component. This component uses service deployment models and standard information models to create high-level monitoring templates based on an ontology-based schema. However, this approach is validated through a proof-of-concept implementation.

A study by Angui et al. [190] suggests that zero-touch cloud Radio Access Network (RAN) management automates the cloud-RAN deployment to have end-to-end services. It suggests a model named zero-touch commissioning to automate processes and tasks such as resource discovery and life cycle management of RAN units. Automated frameworks like ZSM are vulnerable, and attacks such as data modification, man-in-the-middle, replay, and session key disclosure attacks can affect the networks. However, this network management solution does not offer security. So, by deploying networks in ZSM without any security measures and communicating through an insecure channel, the networks are vulnerable to any kind of attack, such as unauthorized access, data integrity violations, and Distributed Denial of Service (DDoS). To enable secure data sharing in ZSM, Kumar et al. [191] propose a blockchain-based framework with DL algorithms, including an IDS that combines a variational autoencoder and an attention-based gated recurrent unit.

B. Practical Applications

In this section, we focus on state-of-the-art research on the three main applications of self-running networks. Table VII shows an overview of these studies.

1) Resource Management: As the traditional methods for resource management do not work in today's complex networks, the researchers focus on how resource management can be autonomous. Zhang et al. [216] propose an ML-based framework to enhance radio resource management, which is useful in heterogeneous networks with diverse service demands. Their approach focuses on maximizing the energy efficiency of the system while meeting constraints such as QoS, interference limitations, and power limitations. The framework involves solving the user association problem using the Lagrange dual decomposition method, while subchannel allocation and power control are addressed through semi-supervised learning and Deep Neural Networks (DNNs). The simulation results show that the proposed scheme enhances energy efficiency compared to traditional methods and has a lower computational complexity. However, the effect of the DL algorithm is significantly dependent on the labeled samples.

The paper by Yu et al. [217] proposes an intelligent ultra-dense edge computing framework to address the challenges of resource management in multi-access edge computing within 5G networks. The framework combines blockchain and AI to optimize computation offloading, resource allocation, and service caching placement. To implement this framework, a two-timescale DRL approach is introduced, which consists of fast-timescale learning for delay-sensitive decisions and slow-timescale learning for delay-insensitive decisions. Additionally, Federated Learning (FL) is utilized to train the model in a distributed manner to ensure data privacy and reduce training overhead. This framework reduces task execution time and network resource usage. Specifically, the proposed DRL algorithm can reduce task execution time by up to 31.87% compared to other benchmark strategies. However, the framework assumes homogeneous resource availability and does not fully address the complexity introduced by highly heterogeneous resources. Also, it only assumes a static environment for the caching decisions.

Mason et al. [218] work on a solution for dynamic resource allocation in network slicing scenarios using the DRL approach. They develop a distributed architecture where multiple agents cooperate to manage network resources. The system is designed to address the diverse requirements of different network slices, such as enhanced mobile broadband and ultra-reliable low-latency communication. The learning agents are trained to allocate network resources dynamically, adapt to changing conditions, and optimize performance across various network topologies. The proposed DRL-based strategy outperforms traditional static and empirical resource allocation methods. Moreover, it demonstrates the system's adaptability through transfer learning, which leads to the results of how policies learned in one network topology can be efficiently adapted to new scenarios. However, they do not compare their work with other dynamic network resource allocations.

In a part of their study, Allahham et al. [219] focus on resource allocation within the RAN, utilizing deep multi-agent RL. Specifically, they examine mobile health networks with the objective of satisfying the diverse QoS requirements presented by various applications in this domain. To assess the effectiveness of their model, they conduct evaluations of their framework and benchmark it against two other studies. However, this research lacks clear future technological guidance.

2) Network Traffic Analysis and Prediction: In the face of rapid traffic, having autonomous networks to enhance traffic management, analysis, and prediction is so crucial to maintaining high levels of performance and ensuring QoS. Efficient traffic analysis helps identify congestion points, predict potential bottlenecks, and adjust routes that enhance the user

TABLE VII: Existing Research on Practical Applications Toward Self-Running Networks

Paper/Year	Objective	Contribution	Limitation	Methods
Resource Management				
Zhang et al. 2020 [216]	Maximizing energy efficiency in heterogeneous networks while satisfying QoS and interference constraints	• Proposed an ML-based framework combining Lagrange dual decomposition and semi-supervised DNNs for resource allocation, improving energy efficiency, and reducing complexity	Dependent on labeled samples	Lagrange Dual Decomposition, Semi-Supervised Learning and DNNs
Yu et al. 2020 [217]	Achieving real-time, low-overhead computation offloading and resource allocation in ultra-dense 5G networks	• Proposed a DRL approach for offloading, resource allocation, and caching. • Leveraged FL to train the DRL model to ensure data privacy for edge devices.	Assumes homogeneous resources and static environment	DRL, FL, Blockchain Integration
Mason et al. 2022 [218]	Addressing dynamic resource allocation in network slicing scenarios	• Developed a DRL-based resource allocation method • Introduced a distributed multi-agent architecture with transfer learning to enhance performance	Limited observability and no comparison with other dynamic methods	DRL, Transfer Learning
Allahham et al. [219]	Focusing on resource allocation in RAN for mobile health networks to satisfy diverse QoS requirements	• Used deep multi-agent RL for resource allocation	Lacks future guidance	Deep multi-agent RL
Traffic Analysis				
Xavier et al. 2022 [220]	Deploying ML-based traffic classification in programmable network devices	• Introduced MAP4 for in-device ML-based traffic classification • Validated in real-world scenarios, showing high accuracy and low performance impact	Limited to edge deployments due to hardware constraints	Decision Trees
Shahraki et al. 2021 [221]	Exploring Active Learning techniques for network traffic classification	• Achieved high accuracy with fewer labeled samples • Adapted effectively to dynamic traffic environments	Adds retraining overhead	Uncertainty Sampling, Query-By-Committee, and Active Learning
Hardegen et al. 2020 [222]	Predicting network flow characteristics using real-world traffic data	• Built a real-time flow pipeline for DL-based prediction • Collected and analyzed campus network flow data for training	Limited generalizability and model diversity	DNNs
Routing				
Chen et al. 2020 [223]	Improving routing efficiency in SDNs using DRL	• Proposed RL-Routing leveraging trust and throughput features • Demonstrated higher throughput and lower latency across various topologies	Lacks multi-agent coordination	DRL, SDN
Xu et al. 2024 [224]	Addressing QoS-aware routing under dynamic conditions in SDNs	• Proposed a centralized DRL framework with GNN-based state representation and causal inference • Enhanced decisions via feature embedding and causal impact estimation	Adds complexity for real-time use	Graph Neural Networks (GNNs), Causal Inference, DRL

experience by reducing latency and packet loss and improving the reliability and efficiency of network operations. Moreover, by continuously monitoring traffic patterns and behavior, it becomes possible to quickly identify and respond to potential security threats such as intrusions, DDoS attacks, and unauthorized access [225]. By fully harnessing the capabilities of self-running networks, it can detect subtle anomalies and variations in traffic that traditional methods might miss.

In this field, Xavier et al. [220] introduce a framework, MAP4, with the aim of deploying ML models directly within programmable network devices using the P4 language. By leveraging the capabilities of P4, the authors implement decision tree models to classify network traffic. Moreover, the framework addresses the constraints of P4, such as the lack of floating-point operations, by utilizing decision trees that can be expressed through if-else chains. For evaluating this framework, MAP4 is validated through two primary scenarios, IDS and IoT device classification. The results indicate that MAP4 can accurately classify network flows with minimal latency, even under high transfer rates. The per-packet and per-flow models deployed on Netronome SmartNICs showed that most traffic could be classified correctly.

The study by Shahraki et al. [221] research on the application of active learning techniques to network traffic classification. The primary goal is to enhance the efficiency and accuracy of traffic classification systems while minimizing the amount of labeled data required. The paper explores several active learning strategies, including uncertainty sampling, query-by-committee, and evaluates their effectiveness in different network scenarios. The study finds that active learning can reduce the labeling effort by selecting only the most informative instances for labeling, thus maintaining high classification accuracy with fewer labeled samples. However, the computational overhead associated with active learning can be a significant limitation.

Hardegen et al. [222] enhance network traffic engineering by developing a DL-based model that predicts network flow characteristics. By collecting and analyzing real-world network traffic data from a university campus, the authors created a flow data stream pipeline that trains and deploys DNNs. These models predict flow characteristics such as bit rate, duration, and packet count that enable proactive traffic routing. The practical application of these predictions can lead to optimized flow routing, preventing congestion and ensuring balanced load distribution across network paths. Furthermore, they propose a hybrid approach combining centralized and distributed network management architectures, leveraging the predictive capabilities of their model to improve overall network performance.

However, the model was trained and validated on data from a university campus network, which may have specific traffic patterns and behaviors. This dependency on specific network characteristics raises concerns about the model's effectiveness in different network environments that have varying traffic profiles. Additionally, the paper primarily focuses on DNNs without exploring other potentially more suitable models, such as Recurrent Neural Networks (RNNs) and GNNs. These alternative models might better capture sequential or structured data and offer improved performance and accuracy in predicting network flow characteristics.

3) Routing: Self-running networks, specifically in the context of routing, leverage advanced algorithms and ML to optimize the paths that data packets take across the network. By monitoring network conditions such as traffic volume, latency, and congestion, these networks can make real-time adjustments to routing tables. Moreover, the adaptive nature of self-running networks allows them to respond instantly to changes in network topology, such as the addition or failure of nodes, maintaining optimal routing paths without human intervention. This is more valuable in large-scale and highly dynamic environments like mobile networks and cloud data centers, where traditional static routing protocols would struggle to keep up with the rapid changes. Self-running networks also enhance routing reliability through predictive analytics. By analyzing historical data and current network states, these networks can forecast potential issues and reroute traffic preemptively. This predictive routing minimizes the risk of packet loss and ensures consistent data delivery, even under fluctuating network conditions.

Chen et al. [223] present RL-Routing, an innovative algorithm designed to optimize routing in SDN using DRL. The main challenge it addresses is the inefficiency of traditional routing algorithms like open shortest path first and least loaded, which rely on static network states and cannot predict future network changes. This approach uses comprehensive network state information, including link trust levels and switch throughput rates, to optimize routing decisions. Simulation results on different network topologies, such as Fattree, NSFNet, and ARPANet, demonstrate that this approach outperforms traditional methods in terms of throughput and communication delay.

Also, scalability is improved by using a single agent per switch, but multi-agent coordination scenarios are not addressed in this work. A recent study by Xu et al, [224] addresses the challenge of QoS-aware routing in SDN by introducing a centralized DRL framework augmented with causal inference and GNNs. The RL agent quantifies the causal impact of its actions on network state (using causal inference techniques to guide exploration) and employs a GNNs-based state representation to embed node and link features, leading to more effective path selection under dynamic traffic conditions. In simulation tests on real network topologies, this approach outperformed baseline algorithms, reducing packet loss and latency while increasing throughput. However, causal analysis adds computational complexity and may require optimization for real-time deployment.

V. Present Challenges in Self-Running Networks

As we discussed in previous sections, self-running networks promise to remove human intervention by autonomously configuring, optimizing, healing, and protecting themselves. However, there is a complex set of challenges that must be addressed before these networks can be reliably and safely deployed. Issues related to trust, security, orchestration, interoperability, ethical oversight, and scalability are at the forefront. While these obstacles highlight gaps in current technologies and frameworks, they also present opportunities for innovation. Addressing these challenges, since there are tight interrelations, is essential for realizing the full potential of self-running networks in future communication systems. To provide a concise overview of these challenges, Table VIII summarizes the key insights discussed across the subsections that follow.

A. *Trust and Security*

Several studies show that ML-based models have the potential to be the targets of security attacks, such as poisoning training data or adversarial attacks that can damage networks [226]. For instance, when using DL-based IDS, various unknown data sources in networks can lead to misleading and incorrect results [191]. These attacks can manipulate the model by altering the training data, leading to misclassifying dangerous incidents as safe. Recent work by Rifà-Pous et al. [227] reinforces this concern by providing an analysis of how AI integration into future 6G networks introduces new threat vectors. Their study classifies vulnerabilities specific to ML-driven architectures, including model inversion, evasion, and poisoning attacks, and highlights the need for new, adaptive security frameworks. Moreover, Adeke et al. [228] demonstrate that ML-based traffic classifiers can be easily fooled by maliciously crafted inputs, causing misclassification of flows and potential network breaches. While various studies have outlined the types and consequences of adversarial and poisoning attacks on ML-based network security systems, recent findings also emphasize that adversarial robustness itself is insufficiently addressed [229].

In addition to concerns about security and attacks, the black-box nature of ML techniques, characterized by limited transparency regarding how decisions are derived, raises critical concerns about the trustworthiness of these systems. To mitigate these concerns, the integration of XAI has been proposed. However, XAI is a developing field within the broader ML/AI landscape, and its effective application in networking scenarios is still emerging. Jacobs et al. [230], for instance, propose the TRUSTEE framework to enhance trust in ML models. This framework transforms a black-box model, alongside its training dataset, into a white-box model,

TABLE VIII: Overview of Challenges in Self-Running Networks

Challenge Area	Overview	Key References
Trust and Security	• ML-driven networks are vulnerable to poisoning, evasion, and adversarial attacks. • Black-box nature of ML models complicates trust and diagnosis. • XAI techniques are emerging but face scalability and real-time accuracy trade-offs.	[191], [226]–[235]
Orchestration	• Need to orchestrate self-healing, self-optimization, self-protection, and self-configuration. • Trade-off between decentralized agent autonomy and centralized oversight. • Solutions range from multi-agent systems to hierarchical Large Language Models (LLMs) + RL frameworks.	[236]–[241]
Interoperability and Standardization	• Harmonizing heterogeneous hardware, software, and protocols. • Structural frameworks enable cross-domain coordination but risk rigidity. • Standards catalogs provide guidance, yet must evolve to avoid obsolescence.	[242]–[244]
Ethical Implications and Human Oversight	• Autonomous decisions blur accountability and legal liability. • Privacy, transparency, and user consent concerns in critical domains. • Emerging approaches embed explainability and human-in-the-loop checkpoints.	[245]–[249]
Scalability and Complexity	• Scaling requires real-time data analytics and digital twin simulations. • Big data pipelines offer reactive adaptability; digital twins enable proactive planning. • Data synchronization, model accuracy, and privacy preservation remain challenges.	[250]–[252]
Training and Knowledge Acquisition	• Volatile network conditions and limited labeled data complicate model training. • Approaches include stream-based learning, federated paradigms, and hybrid models. • Privacy-preserving anonymization frameworks support data sharing but face fidelity issues.	[253]–[261]

providing a decision tree-based explanation of the model's reasoning. In another study, Fiandrino et al. [231] propose an XAI framework for DRL based RAN controllers named EXPLORA. It links a DRL agent's actions to an attributed graph of network states, enabling real-time visualization of why certain radio resource decisions were made. Another research introduces a framework [232] that contains an explainable edge-security system for OpenRAN architectures, integrating XAI and LLMs into anomaly detection to provide human-interpretable justifications for security decisions. Also, Nazari et al. [233] introduce localized sub-specifications that decompose high-level intents into device-level requirements, making synthesized configurations easier to validate across heterogeneous systems, which makes the system more interpretable.

Despite these advancements, techniques like XAI also struggle with scalability, real-time explanations, and accuracy trade-offs, highlighting an open research direction for future development [234], [235]. Taken together, the research reflects a maturing understanding of the multifaceted risks in ML-driven networks. However, the path forward will require tightly integrated approaches that address both the transparency and resilience of self-running networks. In particular, critical open questions remain regarding the development of universal defense mechanisms against evolving adversarial tactics, the design of operationally useful and scalable explainability techniques for high-speed networks, and the integration of robust and explainable solutions in heterogeneous and dynamic environments. Table IX synthesizes key trust and security approaches, highlighting their respective advantages, known limitations, and broader open questions in secure autonomous networking.

B. *Orchestration*

Self-running networks require an intelligent orchestration system to coordinate autonomous functions such as self-healing, self-optimization, self-protection, and self-configuration. For example, in the event of a major link failure, the system must reroute traffic, optimize new paths, assess potential security threats, and deploy fallback configurations in real time. The challenge lies in designing orchestration that is both responsive to immediate disruptions and adaptive to longer-term network needs, while guaranteeing stability across heterogeneous systems. Current approaches diverge in how orchestration is structured and delegated. Multi-agent paradigms view the network as a collection of interacting AI-driven subsystems, each responsible for a specific control domain. Xiao et al. [236] propose agentic AI networking, in which agents collaborate through shared generative models that encode a global knowledge base. This enables distributed coordination and dynamic task allocation. Their follow-up work [237] extends the concept to cross-layer orchestration, where agents jointly negotiate policies for routing, slicing, and congestion control using shared state representations. These approaches prioritize decentralization and adaptability, but they assume a reliable substrate for agent interaction, which may be difficult to guarantee in dynamic or resource-constrained environments.

In contrast, Benzaid and Taleb [238] emphasize architectural coordination over agent autonomy. Their work highlights the inherent difficulty of aligning multiple self-managing functions in real time and proposes layered orchestration principles that support predictability and control. This view assumes tighter central oversight and favors a more structured orchestration hierarchy, potentially sacrificing some of the flexibility offered by distributed learning agents. A hybrid model is proposed by Shokrnezhad et al. [239], who introduce a hierarchical orches-

TABLE IX: Trust and Security Approaches: Pros, Cons, and Open Questions

Approach	Advantages	Disadvantages	Key Open Questions
Adversarial Defenses	• Protect models from malicious manipulation	• Often tailored to specific threat types and may reduce efficiency	• What universal defense strategies can adapt to evolving adversarial tactics?
XAI Integration	• Improves trust and transparency in model decisions	• May introduce computational overhead and explanation ambiguity	• How can explainability be made operationally useful in high-speed decision contexts?
Hybrid Robust and XAI	• Enables both interpretability and resilience	• Complex to implement and validate in production systems	• How can we co-design robust and explainable systems that scale across diverse network environments?

tration framework combining LLMs for high-level planning with RL agents for localized, low-level decisions. This stratified approach aligns abstract network intents with actionable policies. However, it introduces its own complexities; LLMs are often opaque, prone to hallucinations, and lack the domain-specific grounding required for safe deployment in production networks [240]. Industry frameworks such as Cloudify [241] focus on orchestration across multi-vendor environments and service layers. These efforts prioritize interoperability and operational pragmatism but tend to rely on rule-based automation, limiting adaptability in novel scenarios.

Taken together, these approaches represent competing trade-offs, distributed agent systems offer flexibility but depend on coordination protocols, hierarchical models provide clarity but risk bottlenecks, and LLMs-driven solutions promise generalization but suffer from explainability and safety gaps. A common thread is the need to define when and how control should transition between centralized and decentralized mechanisms, ensure global coherence in dynamic conditions, and establish safe and interpretable multi-layer orchestration architectures. Table X compares the principal orchestration approaches, outlining their strengths, limitations, and open research questions that guide future architectural design.

C. Interoperability and Standardization

The challenge of interoperability and standardization in self-running networks is complex since it is related to various elements such as hardware compatibility, software integration, protocol alignment, and security frameworks. These networks combine different technologies, and their successful deployment relates to achieving high interoperability among disparate systems [242]. There are different approaches on the scope and pace of Interoperability and standardization. Some emphasize structural solutions that enable systems to coordinate across domains despite differences in implementation. Xu et al. [243], for example, propose a multi-domain architecture that enables interoperability by enforcing standard interaction rules between independently managed network segments. While this supports collaboration without centralized control, it also relies heavily on agreement at design time, so it can limit flexibility as network conditions evolve.

Others focus on codifying existing practices into broadly applicable standards. The NIST big data interoperability framework [244] exemplifies this approach by cataloging over 130 relevant standards and mapping them to stages of autonomous data pipelines. This effort provides a foundation for systematic alignment across systems but also reveals critical gaps, particularly in areas such as policy explainability, federated telemetry, and model provenance. This study shows that many autonomy-specific needs are unsupported by current standards. At the same time, standards that offer cross-vendor compatibility may become obsolete as new protocols and abstractions emerge.

This dynamic undermines efforts to impose universal formats and suggests that future frameworks must be adaptable and modular by design. Moving forward, self-running networks will require interoperability and standardization frameworks that not only connect components but also support dynamic behavior and adaptation across heterogeneous environments. Key open challenges include ensuring that interoperability frameworks remain flexible as underlying technologies evolve, enabling standards to keep pace with rapid innovation and emerging protocols, and designing modular abstractions that strike a balance between detailed granularity and broad generality. Table XI outlines the major approaches to interoperability and standardization, highlighting their respective strengths, limitations, and specific open research questions guiding future cross-domain compatibility efforts.

D. Ethical Implications and Human Oversight

As networks become more autonomous, leveraging AI and ML for network management without human intervention can raise several ethical considerations [245], [246]. One key concern is the degree of autonomy granted to such systems, especially when deployed in critical domains such as healthcare or transportation. Cheong [247] underscores the need for explicit ethical boundaries, calling for human-in-the-loop mechanisms that ensure systems remain intelligible and behaviorally aligned with social values. This perspective frames ethical oversight primarily as a governance challenge, where regulatory guardrails are meant to contain otherwise opaque and potentially unsafe system behavior. Accountability is another factor in this challenge. As decision-making is delegated to autonomous systems, traditional notions of responsibility become blurred. When adverse outcomes occur, such as misrouted traffic or denied service, pinpointing whether the fault lies with the algorithm, the designer, or the system integrator is often ambiguous. Legal and ethical frameworks must evolve to address this gap, but current models offer little consensus on how to assign blame or enforce redress in machine-led operations. Efforts to operationalize oversight

TABLE X: Orchestration Approaches: Pros, Cons, and Open Questions

Model	Advantages	Disadvantages	Key Open Questions
Decentralized Agents	• High adaptability and local autonomy	• Requires robust coordination and communication layers	• How can agent-based systems ensure global coherence under dynamic conditions?
Hierarchical Control	• Structured oversight and policy consistency	• Risk of central bottlenecks or slow reaction to local events	• When and how should control shift between centralized and distributed layers?
Hybrid (LLMs + RL)	• Combines general reasoning with task-specific learning	• Opaqueness of LLMs and integration complexity	• What architectural principles ensure safe and interpretable orchestration across multiple intelligence layers?

TABLE XI: Interoperability and Standardization Approaches: Pros, Cons, and Open Questions

Approach	Advantages	Disadvantages	Key Open Questions
Structural Frameworks	• Enable cross-domain coordination without central control	• Rigid design-time agreements may age poorly	• How can interoperability frameworks remain adaptable across evolving technologies?
Standards Catalogs	• Map existing specifications and ease compliance	• May lag behind innovation; lack semantic depth	• What strategies ensure standards evolve in sync with emerging protocols?
Modular Abstractions	• Promote flexibility through plug-and-play components	• Complex versioning and integration overhead	• How should modules be designed to balance granularity with generality?

have moved toward technical solutions that enhance explainability and operator control.

Patel et al. [248] propose concept-based explanations that translate internal model logic into human-understandable categories. This approach improves traceability and supports intervention, but is most effective when operators already have sufficient domain expertise. It remains unclear whether these explanations are actionable in high-speed or large-scale deployments. Sterz et al. [249] push the discussion further by arguing that “human-in-the-loop” alone is not meaningful unless accompanied by concrete capabilities. They define six measurable conditions, such as epistemic access and causal power, that determine whether a human can truly understand, influence, and take moral responsibility for the system’s decisions. Their framework provides a normative structure but introduces new implementation challenges: how to design systems that grant such access without compromising speed, automation, or privacy. In parallel, ethical concerns about data collection remain.

Self-running networks must analyze large volumes of traffic and user behavior, raising questions about surveillance and informed consent. While technical mechanisms for privacy preservation are developing, ethical use of data is often more about context, policy, and user rights than technical compliance. In comparing these perspectives, there is a clear divide between normative approaches that emphasize governance and responsibility and technical strategies that embed transparency and control into system design. Both directions are necessary but insufficient on their own. Critical open challenges include ensuring that human oversight remains both meaningful and scalable, defining what constitutes adequate explanations for accountability, and mapping technical actions reliably to legal and ethical standards across diverse environments. Hence, the central challenge lies in designing autonomous networks that uphold social values while maintaining operational viability, particularly in time-sensitive scenarios where continuous human intervention is impractical. Table XII presents a comparative view of mechanisms for ethical oversight, clarifying their contributions, limitations, and the specific ethical questions that must guide future developments.

E. Scalability

The integration of diverse vendors, heterogeneous devices, and multifaceted services requires distinct commands to scale networks on demand. Advanced network management approaches can provide significant assistance in addressing these complexities. One approach focuses on data-centric scalability. Dai et al. [250] propose the use of big data analytics combined with distributed computing to process large volumes of heterogeneous network data in real time. Their method leverages ML to extract actionable insights from massive datasets, supporting scalability by improving decision-making throughput. However, this approach is inherently dependent on data availability, high-throughput processing infrastructure, and accurate labeling, which can be difficult to maintain consistently in volatile environments. Alternatively, Hong and Zhou [251] propose NetGraph, a digital twin platform that models data center networks through virtual replicas of the physical infrastructure. By separating configuration from state data, NetGraph enables more controlled and modular scalability, allowing the network to simulate changes before they are deployed. This abstraction offers advantages in planning, fault simulation, and proactive management. Nevertheless, its reliance on accurate synchronization between the digital and physical layers introduces complexity and raises concerns about model drift in real-time environments.

TABLE XII: Ethical Implications and Human Oversight Approaches: Pros, Cons, and Open Questions

Mechanism	Advantages	Disadvantages	Key Open Questions
Human-in-the-Loop	• Adds accountability and introduces human judgment	• May slow down automation and create bottlenecks	• How can human oversight be made both meaningful and scalable in autonomous systems?
Concept Explanations	• Improve transparency by translating model behavior into human terms	• Effectiveness depends on user expertise and domain context	• What defines a sufficient explanation for decision accountability?
Accountability Frameworks	• Clarify moral and legal responsibility for autonomous decisions	• Difficult to implement across jurisdictions and system boundaries	• How can technical actions be mapped reliably to legal and ethical standards?

The potential of digital twin architectures is further discussed by Wu et al. [252], who acknowledge its promise but point to unresolved implementation issues. These include the absence of standardization frameworks, privacy risks in modeling real-time traffic, and dependencies on complementary technologies like IoT and edge computing. These limitations make full-scale deployment of digital twins in dynamic network environments difficult in practice. In comparing these approaches, big data analytics provides reactive scalability by optimizing ongoing operations across distributed nodes, while digital twin systems offer proactive scalability by forecasting outcomes before deployment.

Each method involves trade-offs: the former requires robust data pipelines and analytics maturity, while the latter depends on accurate, continuously updated models and interoperable system components. The complexity of integrating these solutions grows with network size and heterogeneity, highlighting the need for cohesive and adaptable management frameworks. Key open challenges include achieving scalability without sacrificing responsiveness or accuracy, ensuring reliable synchronization and validity of digital twin models at scale, and integrating proactive planning with real-time execution into a unified, operationally efficient system. Table XIII summarizes these scalability strategies, outlining their benefits, limitations, and the specific open questions that will shape future approaches in this regard.

F. Training and Knowledge Acquisition

Training ML models in self-running networks is an important challenge due to volatile environments, limited access to real-world data, and high resource demands. Approaches to this problem diverge in how they address adaptability, distribution, generalization, and data availability. To address the instability of network conditions, several studies prioritize adaptability over batch learning assumptions. Stream-based models, such as those proposed by Wassermann et al. [253], are designed to process data incrementally, maintaining responsiveness in environments where data cannot be stored or revisited. This contrasts with federated approaches, such as those by Yu et al. [254] and Lim et al. [255], which emphasize data privacy and decentralized training by distributing learning across edge nodes. While FL preserves data locality, it introduces new issues such as non-Independent and Identically Distributed (non-IID) data distributions, communication overhead, and uneven resource allocation. Lee et al. [256] attempt to mitigate these by introducing hierarchical learning architectures, though the complexity of coordination across agents raises scalability concerns. A different line of work explores the integration of domain knowledge into the learning process.

Zappone et al. [257] argue that combining analytical models with ML can reduce training data requirements and improve robustness in the face of noisy or incomplete information. While conceptually attractive, these hybrid systems often face practical limitations in dynamic, heterogeneous network environments where model assumptions can become invalid over time. Resource constraints also limit real-world deployment. Liu et al. [258] highlight how training models at scale can incur unacceptable latency and cost, particularly in bandwidth-constrained scenarios. This motivates a shift toward more lightweight or reusable architectures. Recent advances in LLMs reflect this shift by emphasizing transferability and general-purpose reasoning. Wu et al. [259] demonstrate that pretrained LLMs can be adapted to diverse networking tasks with minimal task-specific tuning, offering a potential reduction in training cost and engineering effort.

Wang et al. [262] explore how LLMs can automate network configuration tasks, demonstrating their potential to reduce manual effort and support operator decision-making. However, these models are not without drawbacks and concerns around explainability, domain specificity, and hallucinated outputs remain open. To tackle these challenges, Shajarian et al. [260] propose an AI-driven framework leveraging retrieval-augmented techniques that integrate LLMs with explicit knowledge bases, working toward improving domain correctness and reducing hallucinated outputs while supporting human operators. Across all methods, the lack of accessible, high-quality training data continues to be a shared limitation. Privacy concerns restrict the availability of real-world traffic traces and configuration files, impeding reproducibility and benchmarking.

To mitigate these issues, Wang et al. [261] propose ConfMask, a framework for anonymizing network configurations without removing semantic structure, enabling safer data sharing for training and evaluation. While effective in preserving privacy, anonymization introduces trade-offs in terms of data fidelity and downstream utility for learning systems. While no single method resolves all challenges, the emerging consensus supports a layered strategy combining decentralized training, lightweight adaptation, hybrid modeling, and privacy-aware data sharing. Key open questions include how to achieve continual learning without instability in highly dynamic en-

TABLE XIII: Scalability Strategies: Pros, Cons, and Open Questions

Strategy	Advantages	Disadvantages	Key Open Questions
Big-Data Analytics	• Reactive optimization using live telemetry	• Requires high-throughput pipelines	• How can scalability be achieved without compromising responsiveness or accuracy?
Digital Twins	• Proactive "what-if" planning	• Risk of model drift, privacy exposure	• What frameworks ensure reliable synchronization and model validity at scale?
Hybrid Reactive+Proactive	• Combines fast adaptation with foresight	• Coordination overhead, toolchain complexity	• How can planning and real-time execution be integrated into a cohesive system?

TABLE XIV: Training and Knowledge Acquisition Strategies: Pros, Cons, and Open Questions

Paradigm	Advantages	Disadvantages	Key Open Questions
Stream-based Learning	• Adapts on-the-fly to non-stationary data	• Forgetting past contexts	• What mechanisms support continual learning in dynamic environments while avoiding instability?
Federated Learning	• Keeps raw data local, preserves privacy	• Non-IID Data, communications overhead, uneven resources	• How can collaborative models be both efficient and fair across diverse subsystems?
Pretrained/Transfer LLMs	• Broad coverage	• Hallucinations, compute-heavy	• What design principles ensure reliable and domain-grounded generalization?

vironments, how to ensure fairness and resource efficiency in federated or collaborative training across diverse nodes, and how to design general-purpose models such as LLMs that provide reliable, domain-grounded knowledge without sacrificing interpretability or safety. Table XIV reviews these dominant training and knowledge acquisition approaches, highlighting their respective benefits, limitations, and specific open research directions.

G. Considerations for Researchers

Addressing the outlined challenges requires researchers to strategically focus their efforts on several key areas. Below are essential considerations to guide future work in self-running networks:

Holistic Security and Trust Frameworks: Researchers should prioritize developing robust, adversarial-resistant security mechanisms, integrating XAI techniques to enhance transparency and trust. The goal is to mitigate both known vulnerabilities and emerging threats inherent to ML-driven architectures.

Balanced Orchestration Models: Efforts should target orchestration architectures that effectively combine decentralized adaptability with hierarchical stability. Research must explore mechanisms that enable agents and central controllers to dynamically negotiate tasks and responsibilities while preserving operational integrity, particularly under resource constraints.

Flexible and Modular Standardization: Researchers must advocate for interoperability frameworks that balance rigid standards with adaptive modularity. Proposals should encourage standards that evolve alongside technology innovations, enabling seamless integration of heterogeneous systems without stifling innovation or creating obsolescence.

Ethical Oversight Integration: It is essential for researchers to embed ethical considerations directly within autonomous decision-making frameworks. Future studies should explore methods that offer measurable accountability, ensure human oversight, and uphold privacy rights, especially in critical or sensitive deployment contexts.

Proactive Scalability Approaches: Research should address scalability challenges by integrating data-centric and digital twin methodologies, proactively managing complexity through simulation and predictive analytics. Researchers need to emphasize creating resilient systems capable of adapting rapidly to varying operational scales and conditions without sacrificing accuracy or performance.

Sustainable Training and Knowledge Acquisition: Efforts should aim for sustainable ML training practices that balance resource consumption, data privacy, and performance. Future research should investigate lightweight, transferable models and methods such as federated learning, hybrid analytical models, and retrieval-augmented systems to address training data limitations and operational constraints.

By targeting these considerations, researchers can address current challenges, accelerating progress towards reliable, transparent, and efficient self-running networks.

VI. Conclusion

Self-running networks have evolved from an aspirational vision to an operational necessity as modern infrastructures grow in complexity and scale. Fueled by advances in 5G/6G, massive-scale IoT, and the cloud–edge continuum, traditional, manually configured systems are no longer sufficient. Our survey revealed that while many existing studies focus on individual autonomy-enabling technologies, they often lack an integrated perspective. This underscores the need for a cohesive framework that organizes the fragmented landscape and defines how self-running network architectures should be designed, evaluated, and deployed.

To this end, we introduced a seven-layer conceptual model encompassing telemetry, analytics, orchestration, execution, and feedback. We elaborated on the core *self-** functionalities (self-configuration, self-optimization, self-healing, and self-protection) alongside a six-level autonomy maturity model that helps assess the progression of network intelligence. By synthesizing foundational paradigms with practical applications and by critically examining the challenges around trust, scalability, interoperability, and ethical governance, this survey lays the foundation for the next phase of autonomous networking. Despite significant strides, fully autonomous networks remain a work in progress. The next phase of autonomous networking aims to move beyond reactive automation toward intelligent, explainable, and self-evolving systems that operate collaboratively across domains. Realizing this vision will require advances in five key areas:

Architectural Standardization. Unified abstractions and interfaces are needed to harmonize telemetry, analytics, orchestration, and feedback across heterogeneous platforms. Collaboration among standards bodies (e.g., ETSI ZSM, ITU-T FG-AN) and industry consortia can accelerate this effort.

Scalable and Explainable Intelligence. Deploying ML/DL at scale needs balancing inference latency, model complexity, and transparency. Future systems should emphasize interpretable AI and decentralized, privacy-aware learning paradigms.

Autonomy Assurance and Trustworthiness. As networks gain autonomy, the need for formal verification, runtime validation, and ethical governance becomes critical. Research into verifiable AI pipelines and policy-compliant autonomy is essential to establish trust in such networks.

Cross-Domain Interoperability. Real-world networks span cloud, edge, IoT, and 5G/6G domains. Self-running networks must bridge these environments through domain-agnostic intent models, standardized interfaces, and shared policy enforcement frameworks.

Open Testbeds and Benchmarks. Progress requires community-driven platforms, shared testbeds, and standardized reproducible benchmarks. Emulating environments such as LBNL's self-driving science networks or Nokia's digital twin systems in open testbeds can also serve as blueprints for experimentation.

Ultimately, self-running networks offer a blueprint for intelligent infrastructure that is resilient to failure, adaptive to context, and optimized for performance with minimal human intervention. This survey lays the groundwork for deeper exploration and interdisciplinary innovation in the pursuit of trustworthy, scalable, and fully autonomous networking systems.

Acknowledgment

This work is partially supported by NSF grants 2113945, 2200538, 2416992, and 2230610 at NC A&T SU. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agency.

Appendix A
List of Abbreviations

SDN	Software-Defined Networking
NFV	Network Functions Virtualization
SDR	Software-Defined Radio
IBN	Intent-Based Networking
SD-WAN	Software-defined Wide Area Networks
IoT	Internet of Things
ML	Machine Learning
NLP	Natural Language Processing
RL	Reinforcement Learning
QoS	Quality of Service
QoE	Quality of Experience
AI	Artificial Intelligence
API	Application Programming Interface
MDs	Management Domains
ZSM	Zero-touch network and Service Management
ZTP	Zero-Touch Provisioning
PBNM	Policy-Based Network Management
DDoS	Distributed Denial of Service
HAN	Home Area Network
VoIP	Voice over Internet Protocol
XAI	Explainable AI
ANM	Autonomic Networks Management
RAN	Radio Access Network
FL	Federated Learning
DRL	Deep Reinforcement Learning
GNNs	Graph Neural Networks
DL	Deep Learning
DNNs	Deep Neural Networks
RNNs	Recurrent Neural Networks
IDS	Intrusion Detection Systems
IPS	Intrusion Prevention Systems
ZTS	Zero Trust Security
VNF	Virtual Network Functions
LLMs	Large Language Models

References

[1] L. Zhang and S. Thomopoulos, "Neural network implementation of the shortest path algorithm for traffic routing in communication networks," in *International 1989 Joint Conference on Neural Networks*. IEEE, 1989, pp. 591–vol.

[2] M. K. M. Ali and F. Kamoun, "Neural networks for shortest path computation and routing in computer networks," *IEEE transactions on neural networks*, vol. 4, no. 6, pp. 941–954, 1993.

[3] A. G. Ganek and T. A. Corbi, "The dawning of the autonomic computing era," *IBM systems Journal*, vol. 42, no. 1, pp. 5–18, 2003.

[4] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[5] L. Fallon, J. Keeney, and R. K. Verma, "Autonomic closed control loops for management, an idea whose time has come?" in *2019 15th International Conference on Network and Service Management (CNSM)*. IEEE, 2019, pp. 1–5.

[6] N. Feamster, J. Rexford, and E. Zegura, "The road to sdn: an intellectual history of programmable networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 87–98, 2014.

[7] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese *et al.*, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 87–95, 2014.

[8] F.-Y. Wang, L. Yang, X. Cheng, S. Han, and J. Yang, "Network softwarization and parallel networks: beyond software-defined networks," *IEEE network*, vol. 30, no. 4, pp. 60–65, 2016.

[9] D. Sur, B. Pfaff, L. Ryzhyk, and M. Budiu, "Full-stack sdn," in *Proceedings of the 21st ACM Workshop on Hot Topics in Networks*, 2022, pp. 130–137.

[10] N. Foster, N. McKeown, J. Rexford, G. Parulkar, L. Peterson, and O. Sunay, "Using deep programmability to put network owners in control," pp. 82–88, 2020.

[11] A. Sivaraman, T. Mason, A. Panda, R. Netravali, and S. A. Kondaveeti, "Network architecture in the age of programmability," pp. 38–44, 2020.

[12] A. Gupta, C. Mac-Stoker, and W. Willinger, "An effort to democratize networking research in the era of ai/ml," in *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, 2019, pp. 93–100.

[13] N. Battula, *PERRY: Flexible and Scalable Data Preprocessing System for" ML for Networks" Pipelines*. University of California, Santa Barbara, 2023.

[14] T. Swamy, A. Zulfiqar, L. Nardi, M. Shahbaz, and K. Olukotun, "Homunculus: Auto-generating efficient data-plane ml pipelines for datacenter networks," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, 2023, pp. 329–342.

[15] P. Kalmbach, J. Zerwas, P. Babarczi, A. Blenk, W. Kellerer, and S. Schmid, "Empowering self-driving networks," in *Proceedings of the afternoon workshop on self-driving networks*, 2018, pp. 8–14.

[16] L. Schmelz, J. Van Den Berg, R. Litjens, K. Zetterberg, M. Amirijoo, K. Spaey, I. Balan, N. Scully, and S. Stefanski, "Self-organisation in wireless networks use cases and their interrelation," in *Wireless World Res. Forum Meeting*, vol. 22. Citeseer, 2009, pp. 1–5.

[17] M. Riftadi, J. Oostenbrink, and F. Kuipers, "Gp4p4: Enabling self-programming networks," *arXiv preprint arXiv:1910.00967*, 2019.

[18] D. of Energy (DoE) Lawrence Berkeley National Laboratory, "Building and deploying self-driving science networks exploring wan, wireless and beyond," https://selfdrivingnetwork.lbl.gov/our-research, 2024, accessed: November 7, 2024.

[19] E. Commission, "Selfnet - framework for self-organized network management in virtualized and software defined networks," https://cordis.europa.eu/project/id/671672, 2018, accessed: September 20, 2024.

[20] Huawei, "Moving towards autonomous driving networks," https://www.huawei.com/en/huaweitech/publication/87/moving-towards-autonomous-driving-networks, Last accessed: July 2023.

[21] Juniper Networks, "Transform IT with AI-driven operations and support," https://www.juniper.net/us/en/products/mist-ai.html, Last accessed: July 2023.

[22] Nokia, "The twin-first revolution: Reimagining network operations with digital twins," 2023, accessed: 2025-05-28. [Online]. Available: https://www.nokia.com/blog/the-twin-first-revolution-reimagining-network-operations-with-digital-twins/

[23] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli, "A survey of autonomic communications," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 1, no. 2, pp. 223–259, 2006.

[24] Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle, "A survey of autonomic network architectures and evaluation criteria," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 464–490, 2011.

[25] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, pp. 1–99, 2018.

[26] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE communications surveys & tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[27] L. Pang, C. Yang, D. Chen, Y. Song, and M. Guizani, "A survey on intent-driven networks," *IEEE Access*, vol. 8, pp. 22 862–22 873, 2020.

[28] A. Leivadeas and M. Falkner, "A survey on intent based networking," *IEEE Communications Surveys & Tutorials*, 2022.

[29] E. Coronado, R. Behravesh, T. Subramanya, A. Fernández-Fernández, S. Siddiqui, X. Costa-Pérez, and R. Riggio, "Zero touch management: A survey of network automation solutions for 5g and 6g networks," *IEEE Communications Surveys & Tutorials*, 2022.

[30] M. Liyanage, Q.-V. Pham, K. Dev, S. Bhattacharya, P. K. R. Maddikunta, T. R. Gadekallu, and G. Yenduri, "A survey on zero touch network and service management (zsm) for 5g and beyond networks," *Journal of Network and Computer Applications*, vol. 203, p. 103362, 2022.

[31] Y. Bai, H. Zhao, X. Zhang, Z. Chang, R. Jäntti, and K. Yang, "Toward autonomous multi-uav wireless network: A survey of reinforcement learning-based approaches," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 3038–3067, 2023.

[32] Y. Zuo, J. Guo, N. Gao, Y. Zhu, S. Jin, and X. Li, "A survey of blockchain and artificial intelligence for 6g wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2494–2528, 2023.

[33] A. Gupta, R. Harrison, M. Canini, N. Feamster, J. Rexford, and W. Willinger, "Sonata: Query-driven streaming network telemetry," in *Proceedings of the 2018 conference of the ACM special interest group on data communication*, 2018, pp. 357–371.

[34] J. Kreps, N. Narkhede, J. Rao *et al.*, "Kafka: A distributed messaging system for log processing," in *Proceedings of the NetDB*, vol. 11, no. 2011. Athens, Greece, 2011, pp. 1–7.

[35] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *2nd USENIX workshop on hot topics in cloud computing (HotCloud 10)*, 2010.

[36] C. Pehlivan, V. Augusto, and X. Xie, "Dynamic capacity planning and location of hierarchical service networks under service level constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 863–880, 2014.

[37] O. Aouedi, V. A. Le, K. Piamrat, and Y. Ji, "Deep learning on network traffic prediction: Recent advances, analysis, and future directions," *ACM Computing Surveys*, 2024.

[38] F. A. Demmese, S. Shajarian, and S. Khorsandroo, "Transfer learning with resnet50 for malicious domains classification using image visualization," *Discover Artificial Intelligence*, vol. 4, no. 1, p. 52, 2024.

[39] R. Enns, "Network configuration protocol (netconf)," *Internet Engineering Task Force, RFC6241*, 2011.

[40] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.

[41] grpc, "grpc: A high performance open-source universal rpc framework." 2020.

[42] I. Red Hat, "Ansible: Automation for everyone," https://github.com/ansible/ansible, 2012, accessed: July 2024.

[43] I. Chef Software, "Chef: Infrastructure automation," https://www.chef.io, 2009, accessed: July 2024.

[44] X. Xu, Y. Yuan, Z. Kincaid, A. Krishnamurthy, R. Mahajan, D. Walker, and E. Zhai, "Relational network verification," in *Proceedings of the ACM SIGCOMM 2024 Conference*, 2024, pp. 213–227.

[45] F. Li, M. Li, Y. Pu, Y. Zhang, X. Wang, and J. Cao, "Xnv: Explainable network verification," *IEEE/ACM Transactions on Networking*, 2024.

[46] K.-T. Förster, R. Mahajan, and R. Wattenhofer, "Consistent updates in software defined networks: On dependencies, loop freedom, and blackholes," in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE, 2016, pp. 1–9.

[47] F. A. Wolf and P. Müller, "Verifiable security policies for distributed systems," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 4–18.

[48] G. J. Holzmann, "The model checker spin," *IEEE Transactions on software engineering*, vol. 23, no. 5, pp. 279–295, 1997.

[49] M. del Mar Gallardo, J. Martínez, and P. Merino, "Model checking active networks with spin," *Computer communications*, vol. 28, no. 6, pp. 609–622, 2005.

[50] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri, "Nusmv: a new symbolic model checker," *International journal on software tools for technology transfer*, vol. 2, pp. 410–425, 2000.

[51] Y. Ge, X. Feng, and F. Tang, "Verification and analysis for ethernet protocols with nusmv," in *Computer, Informatics, Cybernetics and Applications: Proceedings of the CICA 2011*. Springer, 2012, pp. 311–320.

[52] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.

[53] E. M. Clarke, "Model checking," in *Foundations of Software Technology and Theoretical Computer Science: 17th Conference Kharagpur, India, December 18–20, 1997 Proceedings 17*. Springer, 1997, pp. 54–56.

[54] M. Davis, G. Logemann, and D. Loveland, "A machine program for theorem-proving," *Communications of the ACM*, vol. 5, no. 7, pp. 394–397, 1962.

[55] X. Liu, H. Liu, X. Yi, and J. Wang, "Llm-enhanced theorem proving with term explanation and tactic parameter repair," in *Proceedings of the 15th Asia-Pacific Symposium on Internetware*, 2024, pp. 21–30.

[56] R. Stoenescu, M. Popovici, L. Negreanu, and C. Raiciu, "Symnet: Scalable symbolic execution for modern networks," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 314–327.

[57] J. Song, C. Cadar, and P. Pietzuch, "Symbexnet: Testing network protocol implementations with symbolic execution and rule-based specifications," *IEEE Transactions on Software Engineering*, vol. 40, no. 7, pp. 695–709, 2014.

[58] R. Baldoni, E. Coppa, D. C. D'elia, C. Demetrescu, and I. Finocchi, "A survey of symbolic execution techniques," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–39, 2018.

[59] A. Clemm, L. Ciavaglia, L. Granville, and J. Tantsura, "Rfc 9315: Intent-based networking-concepts and definitions," 2022.

[60] Y. Yao, Z. Cui, L. Tian, M. Li, F. Pan, and Y. Hu, "Scaver: A scalable verification system for programmable network," in *Proceedings of the 2024 SIGCOMM Workshop on Formal Methods Aided Network Operation*, 2024, pp. 14–19.

[61] T. L. Foundation, "Opendaylight: Open source sdn platform," https://www.opendaylight.org, 2013, accessed: May 2024.

[62] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow *et al.*, "Onos: towards an open, distributed sdn os," in *Proceedings of the third workshop on Hot topics in software defined networking*, 2014, pp. 1–6.

[63] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.

[64] S. Rani, H. Babbar, M. Krichen, K. Yu, and F. H. Memon, "Network slicing for zero-touch networks: A top-notch technology," *IEEE Network*, 2023.

[65] H. H. Liu, X. Wu, W. Zhou, W. Chen, T. Wang, H. Xu, L. Zhou, Q. Ma, and M. Zhang, "Automatic life cycle management of network configurations," in *Proceedings of the Afternoon Workshop on Self-Driving Networks*, 2018, pp. 29–35.

[66] N. Van Tu, J.-H. Yoo, and J. W.-K. Hong, "Towards intent-based configuration for network function virtualization using in-context learning in large language models," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*. IEEE, 2024, pp. 1–8.

[67] A. Arulappan, G. Raja, A. K. Bashir, A. Mahanti, and M. Omar, "Ztmp: Zero touch management provisioning algorithm for the onboarding of cloud-native virtual network functions," *Mobile Networks and Applications*, pp. 1–13, 2023.

[68] D. Giannopoulos, G. Katsikas, K. Trantzas, D. Klonidis, C. Tranoris, S. Denazis, L. Gifre, R. Vilalta, P. Alemany, R. Muñoz *et al.*, "Across: Automated zero-touch cross-layer provisioning framework for 5g and beyond vertical services," in *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 2023, pp. 735–740.

[69] A. Hassan, S. Aggarwal, M. Ibrahim, P. Sharma, and F. Qian, "Wixor: Dynamic tdd policy adaptation for 5g/xg networks," *Proceedings of the ACM on Networking*, vol. 2, no. CoNEXT4, pp. 1–24, 2024.

[70] J. Keeney and V. Cahill, "Chisel: A policy-driven, context-aware, dynamic adaptation framework," in *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*. IEEE, 2003, pp. 3–14.

[71] T. A. Khan, A. Muhammad, K. Abbas, and W.-C. Song, "Intent-based networking platform: An automated approach for policy and configuration of next-generation networks," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021, pp. 1921–1930.

[72] J.-R. Jiang, H.-W. Huang, J.-H. Liao, and S.-Y. Chen, "Extending dijkstra's shortest path algorithm for software defined networking," in *The 16th Asia-Pacific Network Operations and Management Symposium*. IEEE, 2014, pp. 1–4.

[73] C. Monsanto, J. Reich, N. Foster, J. Rexford, and D. Walker, "Composing software defined networks," in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 1–13.

[74] A. Hazra, A. Morichetta, I. Murturi, L. Lovén, C. K. Dehury, V. C. Pujol, P. K. Donta, and S. Dustdar, "Distributed ai in zero-touch provisioning for edge networks: challenges and research directions," *Computer*, vol. 57, no. 3, pp. 69–78, 2024.

[75] M. El Rajab, L. Yang, and A. Shami, "Zero-touch networks: Towards next-generation network automation," *Computer Networks*, vol. 243, p. 110294, 2024.

[76] I. Šimunić and I. Grgurević, "Automation of network device configuration using zero-touch provisioning-a case study," in *International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures*. Springer, 2021, pp. 105–119.

[77] S. K. Fayazbakhsh, V. Sekar, M. Yu, and J. C. Mogul, "Flowtags: Enforcing network-wide policies in the presence of dynamic middlebox actions," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, 2013, pp. 19–24.

[78] T. S. Salem, G. Castellano, G. Neglia, F. Pianese, and A. Araldo, "Toward inference delivery networks: Distributing machine learning with optimality guarantees," *IEEE/ACM Transactions on Networking*, 2023.

[79] A. Mercian, F. Ahmed, P. Sharma, S. Wackerly, and C. Clark, "Mind the semantic gap: Policy intent inference from network metadata," in *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*. IEEE, 2021, pp. 312–320.

[80] S. Mitropoulos and C. Douligeris, "A policy-driven methodology for managing telecommunication networks," *Annual Review of Communications*, vol. 59, pp. 615–624, 2006.

[81] D. I. Wolinsky, Y. Liu, P. S. Juste, G. Venkatasubramanian, and R. Figueiredo, "On the design of scalable, self-configuring virtual networks," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, 2009, pp. 1–12.

[82] R. M. da Silva Bezerra and J. S. B. Martins, "Scalability issues in network self-management: A partitioning approach towards scalable autonomic management computation," *International Journal of Innovative Computing, Information and Control*, vol. 10, no. 6, pp. 2143–2156, 2014.

[83] H. Chen, *Self-configuration framework for networked systems and applications*. The University of Arizona, 2008.

[84] D. S. Nunes, P. Zhang, and J. S. Silva, "A survey on human-in-the-loop applications towards an internet of all," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 944–965, 2015.

[85] A. Seng, U. Trick, A. Lehmann, and B. Ghita, "Path determination for network slicing in wireless mesh disaster networks," in *Mobile Communication-Technologies and Applications; 25th ITG-Symposium*. VDE, 2021, pp. 1–6.

[86] M. Schacherbauer and A. Banerjee, "Using self-organizing networks in 5g," *Network*, vol. 97, no. 4, pp. 1–4, 2020.

[87] B. Stephens, A. Cox, W. Felter, C. Dixon, and J. Carter, "Past: Scalable ethernet for data centers," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012, pp. 49–60.

[88] J. Mudigonda, P. Yalagandula, M. Al-Fares, and J. C. Mogul, "Spain: Cots data-center ethernet for multipathing over arbitrary topologies." in *NSDI*, vol. 10, 2010, pp. 18–18.

[89] N. S. Bülbül, D. Ergenç, and M. Fischer, "Sdn-based self-configuration for time-sensitive iot networks," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. IEEE, 2021, pp. 73–80.

[90] E. Mota, J. Barbosa, G. B. Figueiredo, M. Peixoto, and C. Prazeres, "A self-configuration framework for balancing services in the fog of things," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 318–332, 2024.

[91] J. Jung, J. Hong, and Y. Yi, "On self-configuring iot with dual radios: A cross-layer approach," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 4064–4077, 2021.

[92] A. Lacava, M. Polese, R. Sivaraj, R. Soundrarajan, B. S. Bhati, T. Singh, T. Zugno, F. Cuomo, and T. Melodia, "Programmable and customized intelligence for traffic steering in 5g networks using open ran architectures," *IEEE Transactions on Mobile Computing*, vol. 23, no. 4, pp. 2882–2897, 2023.

[93] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and learning in o-ran for data-driven nextg cellular networks," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 21–27, 2021.

[94] J. Thaliath, S. Niknam, S. Singh, R. Banerji, N. Saxena, H. S. Dhillon, J. H. Reed, A. K. Bashir, A. Bhat, and A. Roy, "Predictive closed-loop service automation in o-ran based network slicing," *IEEE Communications Standards Magazine*, vol. 6, no. 3, pp. 8–14, 2022.

[95] P. T. A. Quang, J. Leguay, X. Gong, and X. Huiying, "Global qos policy optimization in sd-wan," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*. IEEE, 2023, pp. 202–206.

[96] M. R. Wyatt, S. Herbein, T. Gamblin, and M. Taufer, "Ai4io: A suite of ai-based tools for io-aware scheduling," *The International Journal of High Performance Computing Applications*, vol. 36, no. 3, pp. 370–387, 2022.

[97] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega *et al.*, "Network slicing to enable scalability and flexibility in 5g mobile networks," *IEEE Communications magazine*, vol. 55, no. 5, pp. 72–79, 2017.

[98] A. Tizghadam and A. Leon-Garcia, "Autonomic traffic engineering for network robustness," *IEEE journal on selected areas in communications*, vol. 28, no. 1, pp. 39–50, 2009.

[99] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, "Ai-assisted network-slicing based next-generation wireless networks," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 45–66, 2020.

[100] M. Aboelwafa, G. Alsuhli, K. Banawan, and K. G. Seddik, "Self-optimization of cellular networks using deep reinforcement learning with hybrid action space," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 223–229.

[101] J. Deng, Q. Zheng, G. Liu, J. Bai, K. Tian, C. Sun, Y. Yan, and Y. Liu, "A digital twin approach for self-optimization of mobile networks," in *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2021, pp. 1–6.

[102] M. S. Bonfim, K. L. Dias, and S. F. Fernandes, "Integrated nfv/sdn architectures: A systematic literature review," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–39, 2019.

[103] Y. Shi, Y. E. Sagduyu, and T. Erpek, "Reinforcement learning for dynamic resource optimization in 5g radio access network slicing," in *2020 IEEE 25th international workshop on computer aided modeling and design of communication links and networks (CAMAD)*. IEEE, 2020, pp. 1–6.

[104] T. Hu, Q. Liao, Q. Liu, A. Massaro, and G. Carle, "Fast and scalable network slicing by integrating deep learning with lagrangian methods," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 6346–6351.

[105] D. Saxena and A. K. Singh, "A proactive autoscaling and energy-efficient vm allocation framework using online multi-resource neural network for cloud data center," *Neurocomputing*, vol. 426, pp. 248–264, 2021.

[106] J. Xu, W. Wan, L. Pan, W. Sun, and Y. Liu, "The fusion of deep reinforcement learning and edge computing for real-time monitoring and control optimization in iot environments," in *2024 3rd International Conference on Energy and Power Engineering, Control Engineering (EPECE)*, 2024, pp. 193–196.

[107] B. Dai, Y. Cao, Z. Wu, Z. Dai, R. Yao, and Y. Xu, "Routing optimization meets machine intelligence: A perspective for the future network," *Neurocomputing*, vol. 459, pp. 44–58, 2021.

[108] Y. Ren, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Green intelligence networking for connected and autonomous vehicles in smart cities," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1591–1603, 2022.

[109] J. D. Brutlag, "Aberrant behavior detection in time series for network service monitoring," in *14th Systems Administration Conference (LISA 2000)*, 2000.

[110] B. L. Dalmazo, "A prediction-based approach for anomaly detection in the cloud," Ph.D. dissertation, Universidade de Coimbra (Portugal), 2018.

[111] H. Olufowobi, U. Ezeobi, E. Muhati, G. Robinson, C. Young, J. Zambreno, and G. Bloom, "Anomaly detection approach using adaptive cumulative sum algorithm for controller area network," in *Proceedings of the ACM workshop on automotive cybersecurity*, 2019, pp. 25–30.

[112] G. Pang, C. Shen, and A. Van Den Hengel, "Deep anomaly detection with deviation networks," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 353–362.

[113] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, 2005, pp. 32–32.

[114] Z. Wu and J. Liu, "Network traffic monitoring and real-time risk warning based on static baseline algorithm," *Scalable Computing: Practice and Experience*, vol. 25, no. 2, pp. 928–937, 2024.

[115] S. A. Mohammed, A. R. Mohammed, D. Côté, and S. Shirmohammadi, "A machine-learning-based action recommender for network operation centers," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 2702–2713, 2021.

[116] L. Cerdà-Alabern, G. Iuhasz, and G. Gemmi, "Anomaly detection for fault detection in wireless community networks using machine learning," *Computer Communications*, vol. 202, pp. 191–203, 2023.

[117] S. K. Devi, R. Thenmozhi, and D. S. Kumar, "Self-healing iot sensor networks with isolation forest algorithm for autonomous fault detection and recovery," in *2024 International Conference on Automation and Computation (AUTOCOM)*. IEEE, 2024, pp. 451–456.

[118] V. K. Pandey, S. De, and S. Nandi, "Automated aerial assessment for seamless adaptive adhoc restoration in partially collapsed network," *Computer Communications*, vol. 219, pp. 153–172, 2024.

[119] L. Mata, M. Sousa, P. Vieira, M. Queluz, and A. Rodrigues, "A machine learning driven methodology for alarm prediction towards self-healing in wireless networks," in *2024 Wireless Telecommunications Symposium (WTS)*. IEEE, 2024, pp. 1–6.

[120] D. Uomo, A. Sgambelluri, P. Castoldi, E. De Paoli, F. Paolucci, and F. Cugini, "Failure prediction in software defined flying ad-hoc network," in *Proceedings of the Twenty-Fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, ser. MobiHoc '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 355–357. [Online]. Available: https://doi.org/10.1145/3565287.3617611

[121] X. Huang, S. Bian, Z. Shao, and H. Xu, "Predictive switch-controller association and control devolution for sdn systems," *IEEE/ACM Trans. Netw.*, vol. 28, no. 6, p. 2783–2796, Dec. 2020. [Online]. Available: https://doi.org/10.1109/TNET.2020.3021787

[122] J. a. R. Campos, R. Machado, and M. Vieira, "Leveraging time series autocorrelation through numerical differentiation for improving failure prediction," in *Proceedings of the 12th Latin-American Symposium on Dependable and Secure Computing*, ser. LADC '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 70–79. [Online]. Available: https://doi.org/10.1145/3615366.3615423

[123] W. Chai and Q. Ma, "Application of digital twin and hologram technology to achieve distribution network reliability forecast," in *2022 7th Asia Conference on Power and Electrical Engineering (ACPEE)*, 2022, pp. 783–787.

[124] L. Velasco, S. Barzegar, and M. Ruiz, "Using a snr digital twin for failure management," in *2023 23rd International Conference on Transparent Optical Networks (ICTON)*, 2023, pp. 1–4.

[125] K.-T. Foerster, Y.-A. Pignolet, S. Schmid, and G. Tredan, "Local fast failover routing with low stretch," *SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 1, p. 35–41, Apr. 2018. [Online]. Available: https://doi.org/10.1145/3211852.3211858

[126] S. G. Kulkarni, G. Liu, K. K. Ramakrishnan, M. Arumaithurai, T. Wood, and X. Fu, "Reinforce: Achieving efficient failure resiliency for network function virtualization-based services," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, p. 695–708, Apr. 2020. [Online]. Available: https://doi.org/10.1109/TNET.2020.2969961

[127] M. Reitblatt, M. Canini, A. Guha, and N. Foster, "Fattire: declarative fault tolerance for software-defined networks," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 109–114. [Online]. Available: https://doi.org/10.1145/2491185.2491187

[128] T. Li, B. Cole, P. Morton, and D. Li, "Cisco hot standby router protocol (hsrp)," Cisco Systems, Tech. Rep., 1998.

[129] D. Katz and D. Ward, "Bidirectional forwarding detection," Cisco Systems, Technical Report Document ID: FS-BFD, 2004, accessed: 2025-05-30. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/fs_bfd.html

[130] W. Reese, "Nginx: the high-performance web server and reverse proxy," *Linux Journal*, vol. 2008, no. 173, p. 2, 2008.

[131] A. Kumar, G. Somani, and M. Agarwal, "Comparing haproxy scheduling algorithms during the ddos attacks," *IEEE Networking Letters*, vol. 6, no. 2, pp. 139–142, 2024.

[132] Cisco Systems, *Cisco IOS XE High Availability Configuration Guide, Release 16: Stateful Switchover*, Cisco Systems, 2023, accessed: 2023-10-15. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ha/configuration/xe-16/ha-xe-16-book/ha-config-stateful-switchover.html

[133] Kubernetes Documentation, "StatefulSets," 2025, accessed: 7 February 2025. [Online]. Available: https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/

[134] Y. Yuan, J. Yang, R. Duan, I. Chih-Lin, and J. Huang, "Anomaly detection and root cause analysis enabled by artificial intelligence," in *2020 IEEE Globecom Workshops (GC Wkshps*. IEEE, 2020, pp. 1–6.

[135] L. Mata, M. Sousa, P. Vieira, M. P. Queluz, and A. Rodrigues, "On the use of spatial graphs for performance degradation root-cause analysis toward self-healing mobile networks," *IEEE Access*, vol. 12, pp. 20 490–20 508, 2024.

[136] Z. Guo, S. Dou, W. Jiang, and Y. Xia, "Toward improved path programmability recovery for software-defined wans under multiple controller failures," *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, p.

143–158, Jul. 2023. [Online]. Available: https://doi.org/10.1109/TNET.2023.3286456

[137] O. Ayoub, N. Di Cicco, F. Ezzeddine, F. Bruschetta, R. Rubino, M. Nardecchia, M. Milano, F. Musumeci, C. Passera, and M. Tornatore, "Explainable artificial intelligence in communication networks: A use case for failure identification in microwave networks," *Computer Networks*, vol. 219, p. 109466, 2022.

[138] K.-Y. Nie, C.-W. Chang, C.-C. Kao, and J. Pei, "An intelligent fault location approach using fuzzy logic for improving autonomous network," in *2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2021, pp. 291–296.

[139] J. Ali-Tolppa, M. Kajo, B. Gajic, I. Malanchini, B. Schultz, and Q. Liao, "Cognitive autonomy for network self-healing," *Towards Cognitive Autonomous Networks: Network Management Automation for 5G and Beyond*, pp. 345–384, 2020.

[140] H. Fang, D. Zhang, C. Tan, P. Yu, Y. Wang, and W. Li, "Large language model enhanced autonomous agents for proactive fault-tolerant edge networks," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications Workshops*. IEEE, 2024, pp. 1–2.

[141] I. B. Hafaiedh and M. B. Slimane, "A distributed formal-based model for self-healing behaviors in autonomous systems: from failure detection to self-recovery," *The Journal of Supercomputing*, vol. 78, no. 17, pp. 18 725–18 753, 2022.

[142] H. Fang, P. Yu, C. Tan, J. Zhang, D. Lin, L. Zhang, Y. Zhang, W. Li, and L. Meng, "Self-healing in knowledge-driven autonomous networks: Context, challenges, and future directions," *IEEE Network*, 2024.

[143] S. Caleb, G. Padmapriya, T. Nandhini, F. D. Shadrach, R. Latha *et al.*, "Revolutionizing fault detection in self-healing network via multi-serial cascaded and adaptive network," *Knowledge-Based Systems*, vol. 309, p. 112732, 2025.

[144] B. Mukherjee, M. F. Habib, and F. Dikbiyik, "Network adaptability from disaster disruptions and cascading failures," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 230–238, 2014.

[145] M. Alias, N. Saxena, and A. Roy, "Efficient cell outage detection in 5g hetnets using hidden markov model," *IEEE Communications Letters*, vol. 20, no. 3, pp. 562–565, 2016.

[146] T. Omar, T. Ketseoglou, and I. Naffaa, "A novel self-healing model using precoding & big-data based approach for 5g networks," *Pervasive and Mobile Computing*, vol. 73, p. 101365, 2021.

[147] R. K. Devi and M. Muthukannan, "Self-healing fault tolerance technique in cloud datacenter," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2021, pp. 731–737.

[148] Y. Zhang, X. Nie, J. Jiang, W. Wang, K. Xu, Y. Zhao, M. J. Reed, K. Chen, H. Wang, and G. Yao, "Bds+: An inter-datacenter data replication system with dynamic bandwidth separation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 918–934, 2021.

[149] S. Samarakoon, S. Bandara, N. Jayasanka, and C. Hettiarachchi, "Self-healing and self-adaptive management for iot-edge computing infrastructure," in *2023 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2023, pp. 473–478.

[150] S. A. Oladosu, C. C. Ike, P. A. Adepoju, A. I. Afolabi, A. B. Ige, and O. O. Amoo, "The future of sd-wan: A conceptual evolution from traditional wan to autonomous, selfhealing network systems," *Magna Scientia Advanced Research and Reviews*, 2021.

[151] I. S. Razo-Zapata, G. Castañón, and C. Mex-Perera, "Self-healing in transparent optical packet switching mesh networks: A reinforcement learning perspective," *Computer Networks*, vol. 60, pp. 129–146, 2014.

[152] R. Ambrosone, A. D'Amico, R. D'Ingillo, E. Virgillito, S. Straullu, F. Aquilino, and V. Curri, "Optical amplified line self-healing using gnpy as a service by the sdn control," in *2024 24th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2024, pp. 1–4.

[153] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, and D. Hong, "An enhanced ai-based network intrusion detection system using generative adversarial networks," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2330–2345, 2022.

[154] D. S. M. Narayana, S. B. Nookala, S. Chopra, and U. Shanmugam, "An adaptive threat defence mechanism through self defending network to prevent hijacking in wifi network," in *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*. IEEE, 2023, pp. 133–138.

[155] Palo Alto Networks, "What is soar vs. siem vs. xdr?" 2024, accessed: 2025-02-08. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-soar-vs-siem-vs-xdr

[156] N. TN, D. Pramod, and R. Singh, "Zero trust security model: Defining new boundaries to organizational network," in *Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing*, 2023, pp. 603–609.

[157] E. B. Fernandez and A. Brazhuk, "A critical analysis of zero trust architecture (zta)," *Computer Standards & Interfaces*, vol. 89, p. 103832, 2024.

[158] Blumira, "Understanding quarantine in cybersecurity," 2024, accessed: 2025-02-08. [Online]. Available: https://www.blumira.com/glossary/quarantine

[159] N. Basta, M. Ikram, M. A. Kaafar, and A. Walker, "Towards a zero-trust micro-segmentation network security strategy: an evaluation framework," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–7.

[160] CyberArk, "What is just-in-time access?" 2024, accessed: 2025-02-08. [Online]. Available: https://www.cyberark.com/what-is/just-in-time-access/

[161] M. A. Salitin and A. H. Zolait, "The role of user entity behavior analytics to detect network attacks in real time," in *2018 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT)*. IEEE, 2018, pp. 1–5.

[162] Y. Cui, W. Shen, J. Zhang, W. Lu, C. Liu, L. Sun, and S. Chen, "Using ebgan for anomaly intrusion detection," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–7.

[163] N. Virvilis, B. Vanautgaerden, and O. S. Serrano, "Changing the game: The art of deceiving sophisticated attackers," in *2014 6th International Conference On Cyber Conflict (CyCon 2014)*. IEEE, 2014, pp. 87–97.

[164] M. Rottmann, K. Maag, R. Chan, F. Hüger, P. Schlicht, and H. Gottschalk, "Detection of false positive and false negative samples in semantic segmentation," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 1351–1356.

[165] C. Skandylas and N. Khakpour, "Design and implementation of self-protecting systems: A formal approach," *Future Generation Computer Systems*, vol. 115, pp. 421–437, 2021.

[166] S. S. Gill, M. Golec, J. Hu, M. Xu, J. Du, H. Wu, G. K. Walia, S. S. Murugesan, B. Ali, M. Kumar *et al.*, "Edge ai: A taxonomy, systematic review and future directions," *Cluster Computing*, vol. 28, no. 1, pp. 1–53, 2025.

[167] S. Huang, C. M. Poskitt, and L. K. Shar, "Actism: Threat-informed dynamic security modelling for automotive systems," *arXiv preprint arXiv:2412.00416*, 2024.

[168] J. Farmani and A. K. Zadeh, "Ai-based self-healing solutions applied to cellular networks: An overview," *arXiv preprint arXiv:2311.02390*, 2023.

[169] R. K. Vankayalapati and C. Pandugula, "Ai-powered self-healing cloud infrastructures: A paradigm for autonomous fault recovery," *Migration Letters*, vol. 19, no. 6, pp. 1173–1187, 2022.

[170] W. Grover and B. Venables, "Performance of the selfhealing network protocol with random individual link failure times," in *ICC 91 International Conference on Communications Conference Record*. IEEE, 1991, pp. 660–666.

[171] H. Q. Ali, A. B. Darabi, and S. Coleri, "Optimization theory based deep reinforcement learning for resource allocation in ultra-reliable wireless networked control systems," *IEEE Transactions on Communications*, 2024.

[172] M. Avgeris, A. Leivadeas, and I. Lambadaris, "A reinforcement-learning self-healing approach for virtual network function placement," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2023, pp. 1–5.

[173] D. A. Joseph, A. Tavakoli, and I. Stoica, "A policy-aware switching layer for data centers," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, 2008, pp. 51–62.

[174] A. Arulappan, A. Mahanti, K. Passi, S. Thiruvenkadam, R. Naha, and G. Raja, "Dqn approach for adaptive self-healing of vnfs in cloud-native network," *IEEE Access*, 2024.

[175] ITU-T, "Framework for evaluating intelligence levels of future networks including imt-2020," 2020.

[176] D. C. Verma, "Simplifying network administration using policy-based management," *IEEE network*, vol. 16, no. 2, pp. 20–26, 2002.

[177] A. I. Rana and M. Ó. Foghlú, "Policy-based network management in home area networks: Interim test results," in *2009 3rd International Conference on New Technologies, Mobility and Security*. IEEE, 2009, pp. 1–3.

[178] T. Solomon, A. M. Zungeru, R. Selvaraj, O. Phakedi, and O. Bagwasi, "Policy-based network management in biust network," *American Journal of Engineering and Applied Sciences*, vol. 10, no. 3, pp. 661–668, Jun 2017. [Online]. Available: https://thescipub.com/abstract/ajeassp.2017.661.668

[179] H. Alquhayz, N. Alalwan, A. I. Alzahrani, A. H. Al-Bayatti, and M. S. Sharif, "Policy-based security management system for 5g heterogeneous networks," *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1–14, 2019.

[180] S. T. Arzo, R. Bassoli, F. Granelli, and F. H. Fitzek, "Multi-agent based autonomic network management architecture," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3595–3618, 2021.

[181] K. Tsagkaris, M. Logothetis, V. Foteinos, G. Poulios, M. Michaloliakos, and P. Demestichas, "Customizable autonomic network management: integrating autonomic network management and software-defined networking," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 61–68, 2015.

[182] W. Jiang, M. Strufe, and H. Schotten, "Autonomic network management for software-defined and virtualized 5g systems," in *European Wireless; 23th European Wireless Conference*. VDE, 2017, pp. 1–6.

[183] A. Stamou, N. Dimitriou, K. Kontovasilis, and S. Papavassiliou, "Autonomic handover management for heterogeneous networks in a future internet context: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3274–3297, 2019.

[184] K. Abbas, M. Afaq, T. Ahmed Khan, A. Rafiq, and W.-C. Song, "Slicing the core network and radio access network domains through intent-based networking for 5g networks," *Electronics*, vol. 9, no. 10, p. 1710, 2020.

[185] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, "Network slice lifecycle management for 5g mobile networks: An intent-based networking approach," *IEEE Access*, vol. 9, pp. 80 128–80 146, 2021.

[186] A. Collet, A. Banchs, and M. Fiore, "Lossleap: Learning to predict for intent-based networking," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 2138–2147.

[187] B. Orlandi, S. Lataste, S. Kerboeuf, M. Bouillon, X. Huang, F. Faucheux, A. Shahbazi, and P. Delvallet, "Intent-based network management with user-friendly interfaces and natural language processing," in *2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN)*. IEEE, 2024, pp. 163–170.

[188] F. Rezazadeh, H. Chergui, L. Alonso, and C. Verikoukis, "Continuous multi-objective zero-touch network slicing via twin delayed ddpg and openai gym," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[189] N. F. S. de Sousa, D. L. Perez, C. E. Rothenberg, and P. H. Gomes, "End-to-end service monitoring for zero-touch networks," *Journal of ICT Standardization*, vol. 9, no. 2, pp. 91–112, 2021.

[190] B. Angui, R. Corbel, V. Q. Rodriguez, and E. Stephan, "Towards 6g zero touch networks: The case of automated cloud-ran deployments," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 1–6.

[191] R. Kumar, P. Kumar, M. Aloqaily, and A. Aljuhani, "Deep-learning-based blockchain for secure zero touch networks," *IEEE Communications Magazine*, vol. 61, no. 2, pp. 96–102, 2022.

[192] E. J. B. D. Durham, "The common open policy service) protocol," 2000.

[193] R. Yavatkar, "A framework for policy-based admission control," 2000.

[194] M. Sloman, "Policy driven management for distributed systems," *Journal of network and Systems Management*, vol. 2, pp. 333–360, 1994.

[195] Y. Cheng, R. Farha, M. S. Kim, A. Leon-Garcia, and J. W.-K. Hong, "A generic architecture for autonomic service and network management," *Computer Communications*, vol. 29, no. 18, pp. 3691–3709, 2006.

[196] B. Jennings, S. Van Der Meer, S. Balasubramaniam, D. Botvich, M. Ó. Foghlú, W. Donnelly, and J. Strassner, "Towards autonomic management of communications networks," *IEEE Communications Magazine*, vol. 45, no. 10, pp. 112–121, 2007.

[197] H. Derbel, N. Agoulmine, and M. Salaün, "Anema: Autonomic network management architecture to support self-configuration and self-optimization in ip networks," *Computer Networks*, vol. 53, no. 3, pp. 418–430, 2009.

[198] J. Famaey, S. Latré, J. Strassner, and F. De Turck, "A hierarchical approach to autonomic network management," in *2010 IEEE/IFIP Network Operations and Management Symposium Workshops*. Ieee, 2010, pp. 225–232.

[199] W. Berrayana, H. Youssef, and G. Pujolle, "A generic cross-layer architecture for autonomic network management with network wide knowledge," in *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2012, pp. 82–87.

[200] X. Long, X. Gong, X. Que, W. Wang, B. Liu, S. Jiang, and N. Kong, "Autonomic networking: Architecture design and standardization," *IEEE Internet Computing*, vol. 21, no. 5, pp. 48–53, 2017.

[201] A. Louca, A. Mauthe, and D. Hutchinson, "Autonomic network management for next generation networks," *PG Net*, 2010.

[202] E. Zeydan and Y. Turk, "Recent advances in intent-based networking: A survey," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[203] J. Andrade-Hoz, Q. Wang, and J. M. Alcaraz-Calero, "Infrastructure-wide and intent-based networking dataset for 5g-and-beyond ai-driven autonomous networks," *Sensors*, vol. 24, no. 3, p. 783, 2024.

[204] X. Zheng, A. Leivadeas, and M. Falkner, "Intent based networking management with conflict detection and policy resolution in an enterprise network," *Computer Networks*, vol. 219, p. 109457, 2022.

[205] K. Mehmood, H. K. Mendis, K. Kralevska, and P. E. Heegaard, "Intent-based network management and orchestration for smart distribution grids," in *2021 28th International Conference on Telecommunications (ICT)*. IEEE, 2021, pp. 1–6.

[206] B. E. Ujcich, A. Bates, and W. H. Sanders, "Provenance for intent-based networking," in *2020 6th IEEE conference on network softwarization (NetSoft)*. IEEE, 2020, pp. 195–199.

[207] B. Martini, M. Gharbaoui, and P. Castoldi, "Intent-based network slicing for sdn vertical services with assurance: Context, design and preliminary experiments," *Future Generation Computer Systems*, vol. 142, pp. 101–116, 2023.

[208] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, "Ensemble learning-based network data analytics for network slice orchestration and management: An intent-based networking mechanism," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–5.

[209] A. Singh, G. S. Aujla, and R. S. Bali, "Intent-based network for data dissemination in software-defined vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5310–5318, 2020.

[210] J. Mcnamara, D. Camps-Mur, M. Goodarzi, H. Frank, L. Chinchilla-Romero, F. Cañellas, A. Fernández-Fernández, and S. Yan, "Nlp powered intent based network management for private 5g networks," *IEEE Access*, vol. 11, pp. 36 642–36 657, 2023.

[211] A. S. Jacobs, R. J. Pfitscher, R. H. Ribeiro, R. A. Ferreira, L. Z. Granville, W. Willinger, and S. G. Rao, "Hey, lumi! using natural language for {intent-based} network management," in *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, 2021, pp. 625–639.

[212] A. Mekrache, A. Ksentini, and C. Verikoukis, "Intent-based management of next-generation networks: an llm-centric approach," *IEEE Network*, 2024.

[213] M. Riftadi and F. Kuipers, "P4i/o: Intent-based networking with p4," in *2019 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2019, pp. 438–443.

[214] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, and A. F. Skarmeta, "Machine learning-based zero-touch network and service management: A survey," *Digital Communications and Networks*, vol. 8, no. 2, pp. 105–123, 2022.

[215] ETSI, "Zero-touch network and service management (zsm); means of automation," https://www.etsi.org/technologies/zero-touch-network-service-management, 2020, accessed: 2024.

[216] H. Zhang, H. Zhang, K. Long, and G. K. Karagiannidis, "Deep learning based radio resource management in noma networks: User association, subchannel and power allocation," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2406–2415, 2020.

[217] S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu, "When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5g ultradense network," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2238–2251, 2020.

[218] F. Mason, G. Nencioni, and A. Zanella, "Using distributed reinforcement learning for resource orchestration in a network slicing scenario," *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 88–102, 2023.

[219] M. S. Allahham, A. A. Abdellatif, N. Mhaisen, A. Mohamed, A. Erbad, and M. Guizani, "Multi-agent reinforcement learning for network selection and resource allocation in heterogeneous multi-rat networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1287–1300, 2022.

[220] B. M. Xavier, R. S. Guimarães, G. Comarela, and M. Martinello, "Map4: A pragmatic framework for in-network machine learning traffic classification," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4176–4188, 2022.

[221] A. Shahraki, M. Abbasi, A. Taherkordi, and A. D. Jurcut, "Active learning for network traffic classification: a technical study," *IEEE*

Transactions on Cognitive Communications and Networking, vol. 8, no. 1, pp. 422–439, 2021.

[222] C. Hardegen, B. Pfülb, S. Rieger, and A. Gepperth, "Predicting network flow characteristics using deep learning and real-world network traffic," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2662–2676, 2020.

[223] Y.-R. Chen, A. Rezapour, W.-G. Tzeng, and S.-C. Tsai, "Rl-routing: An sdn routing algorithm based on deep reinforcement learning," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 3185–3199, 2020.

[224] J. Xu, Y. Wang, B. Zhang, and J. Ma, "A graph reinforcement learning based sdn routing path selection for optimizing long-term revenue," *Future Generation Computer Systems*, vol. 150, pp. 412–423, 2024.

[225] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[226] S. Ayoubi, N. Limam, M. A. Salahuddin, N. Shahriar, R. Boutaba, F. Estrada-Solano, and O. M. Caicedo, "Machine learning for cognitive network management," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 158–165, 2018.

[227] H. Rifa-Pous, V. Garcia-Font, C. Nunez-Gomez, and J. Salas, "Security, trust and privacy challenges in ai-driven 6g networks," *arXiv preprint arXiv:2409.10337*, 2024.

[228] J. M. Adeke, G. Liu, J. Zhao, N. Wu, and H. M. Bashir, "Securing network traffic classification models against adversarial examples using derived variables," *Future Internet*, vol. 15, no. 12, p. 405, 2023.

[229] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.

[230] A. S. Jacobs, R. Beltiukov, W. Willinger, R. A. Ferreira, A. Gupta, and L. Z. Granville, "Ai/ml for network security: The emperor has no clothes," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 1537–1551.

[231] C. Fiandrino, L. Bonati, S. D'Oro, M. Polese, T. Melodia, and J. Widmer, "Explora: Ai/ml explainability for the open ran," *Proceedings of the ACM on Networking*, vol. 1, no. CoNEXT3, pp. 1–26, 2023.

[232] H. Wen, P. Sharma, V. Yegneswaran, P. Porras, A. Gehani, and Z. Lin, "6g-xsec: Explainable edge security for emerging openran architectures," in *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*, 2024, pp. 77–85.

[233] A. Nazari, Y. Zhang, M. Raghothaman, and H. Chen, "Localized explanations for automatically synthesized network configurations," in *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*, 2024, pp. 52–59.

[234] H. Manthena, S. Shajarian, J. Kimmell, M. Abdelsalam, S. Khorsandroo, and M. Gupta, "Explainable artificial intelligence (xai) for malware analysis: A survey of techniques, applications, and open challenges," *IEEE Access*, 2025.

[235] N. Ullah, J. A. Khan, I. De Falco, and G. Sannino, "Explainable artificial intelligence: importance, use domains, stages, output shapes, and challenges," *ACM Computing Surveys*, vol. 57, no. 4, pp. 1–36, 2024.

[236] Y. Xiao, G. Shi, and P. Zhang, "Towards agentic ai networking in 6g: A generative foundation model-as-agent approach," *arXiv preprint arXiv:2503.15764*, 2025.

[237] Y. Xiao, H. Zhou, X. Li, Y. Gao, G. Shi, and P. Zhang, "Sannet: A semantic-aware agentic ai networking framework for multi-agent cross-layer coordination," *arXiv preprint arXiv:2505.18946*, 2025.

[238] C. Benzaid and T. Taleb, "Ai-driven zero touch network and service management in 5g and beyond: Challenges and research directions," *Ieee Network*, vol. 34, no. 2, pp. 186–194, 2020.

[239] M. Shokrnezhad and T. Taleb, "An autonomous network orchestration framework integrating large language models with continual reinforcement learning," *arXiv preprint arXiv:2502.16198*, 2025.

[240] O. G. Lira, O. M. Caicedo, and N. L. da Fonseca, "Large language models for zero touch network configuration management," *IEEE Communications Magazine*, 2024.

[241] M. Udawant, "Network service orchestration & automation: The cloudify way," https://www.calsoftinc.com/blogs/network-service-orchestration-automation-the-cloudify-way.html, 2025, accessed: 2025-05-30.

[242] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[243] R. Xu, D. Nagothu, Y. Chen, A. Aved, E. Ardiles-Cruz, and E. Blasch, "A secure interconnected autonomous system architecture for multi-domain iot ecosystems," *IEEE Communications Magazine*, vol. 62, no. 7, pp. 52–57, 2024.

[244] W. L. Chang, D. Boyd, and O. Levin, "Nist big data interoperability framework: Volume 6, reference architecture," 2019.

[245] C. Cath, "Governing artificial intelligence: ethical, legal and technical opportunities and challenges," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2133, p. 20180080, 2018.

[246] N. Bostrom and E. Yudkowsky, "The ethics of artificial intelligence," in *The Cambridge Handbook of Artificial Intelligence*, K. Frankish and W. M. Ramsey, Eds. Cambridge: Cambridge University Press, 2014, pp. 316–334.

[247] B. C. Cheong, "Transparency and accountability in ai systems: safeguarding wellbeing in the age of algorithmic decision-making," *Frontiers in Human Dynamics*, vol. 6, p. 1421273, 2024.

[248] S. Patel, D. Han, N. Narodystka, and S. A. Jyothi, "Toward trustworthy learning-enabled systems with concept-based explanations," in *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*, 2024, pp. 60–67.

[249] S. Sterz, K. Baum, S. Biewer, H. Hermanns, A. Lauber-Rönsberg, P. Meinel, and M. Langer, "On the quest for effectiveness in human oversight: Interdisciplinary perspectives," in *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, 2024, pp. 2495–2507.

[250] H.-N. Dai, R. C.-W. Wong, H. Wang, Z. Zheng, and A. V. Vasilakos, "Big data analytics for large-scale wireless networks: Challenges and opportunities," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–36, 2019.

[251] H. Hong, Q. Wu, F. Dong, W. Song, R. Sun, T. Han, C. Zhou, and H. Yang, "Netgraph: An intelligent operated digital twin platform for data center networks," in *Proceedings of the ACM SIGCOMM 2021 workshop on network-application integration*, 2021, pp. 26–32.

[252] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[253] S. Wassermann, T. Cuvelier, P. Mulinka, and P. Casas, "Adaptive and reinforcement learning approaches for online network monitoring and analysis," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1832–1849, 2020.

[254] R. Yu, S. Xie, Y. Xu, Y. Zhang, and M. Xie, "When deep reinforcement learning meets federated learning: Intelligent multi-timescale resource management for mec in 5g ultra dense network," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2238–2251, 2020.

[255] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE communications surveys & tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[256] H. Lee, S. H. Lee, and T. Q. Quek, "Artificial intelligence meets autonomy in wireless networks: A distributed learning approach," *IEEE Network*, vol. 36, no. 6, pp. 100–107, 2022.

[257] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, ai-based, or both?" *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7331–7376, 2019.

[258] Y. Liu, J. Han, K. Xue, J. Li, Q. Sun, and J. Lu, "Decc: Achieving low latency in data center networks with deep reinforcement learning," *IEEE Transactions on Network and Service Management*, 2023.

[259] D. Wu, X. Wang, Y. Qiao, Z. Wang, J. Jiang, S. Cui, and F. Wang, "Netllm: Adapting large language models for networking," in *Proceedings of the ACM SIGCOMM 2024 Conference*, 2024, pp. 661–678.

[260] S. Shajarian, S. Khorsandroo, and M. Abdelsalam, "Poster: Intelligent network management: Rag-enhanced llms for log analysis, troubleshooting, and documentation," in *Proceedings of the 20th International Conference on emerging Networking EXperiments and Technologies*, 2024, pp. 27–28.

[261] Y. Wang, Q. Men, Y. Xiao, Y. Chen, and G. Liu, "Confmask: Enabling privacy-preserving configuration sharing via anonymization," in *Proceedings of the ACM SIGCOMM Conference*, 2024, pp. 465–483.

[262] C. Wang, M. Scazzariello, A. Farshin, S. Ferlin, D. Kostić, and M. Chiesa, "Netconfeval: Can llms facilitate network configuration?" *Proceedings of the ACM on Networking*, vol. 2, no. CoNEXT2, pp. 1–25, 2024.